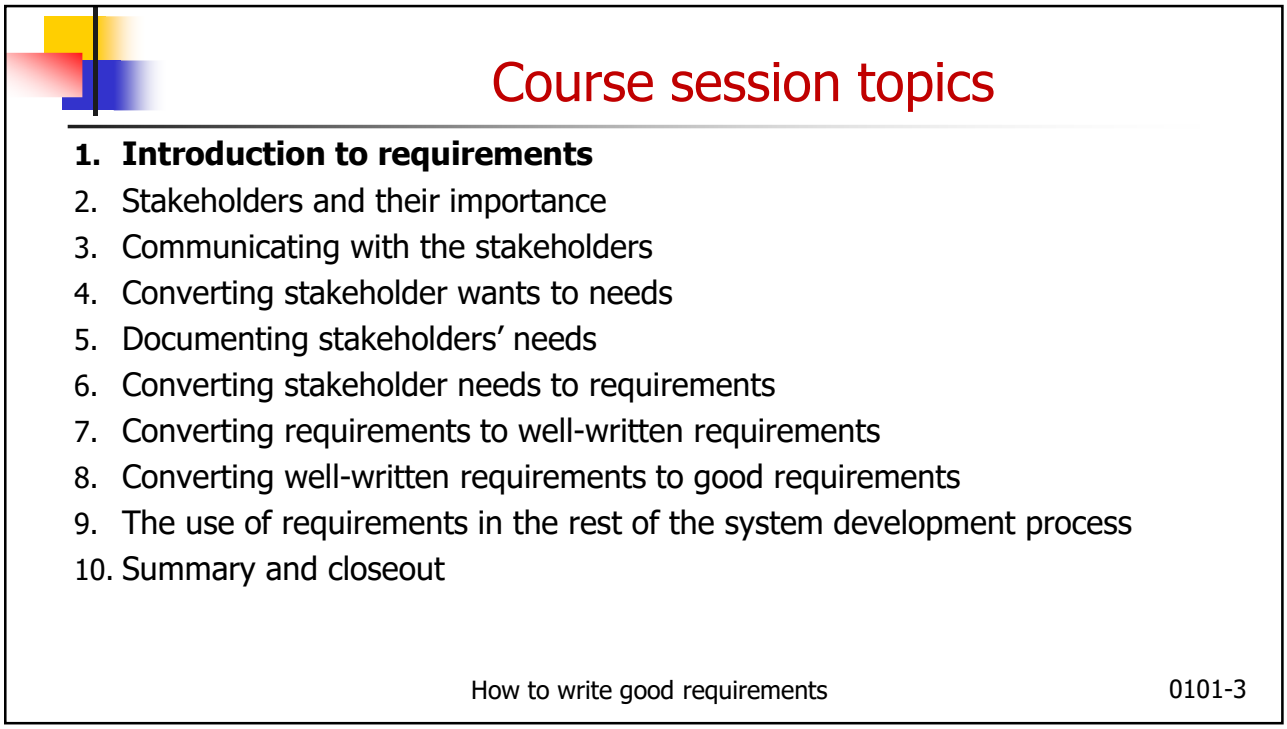
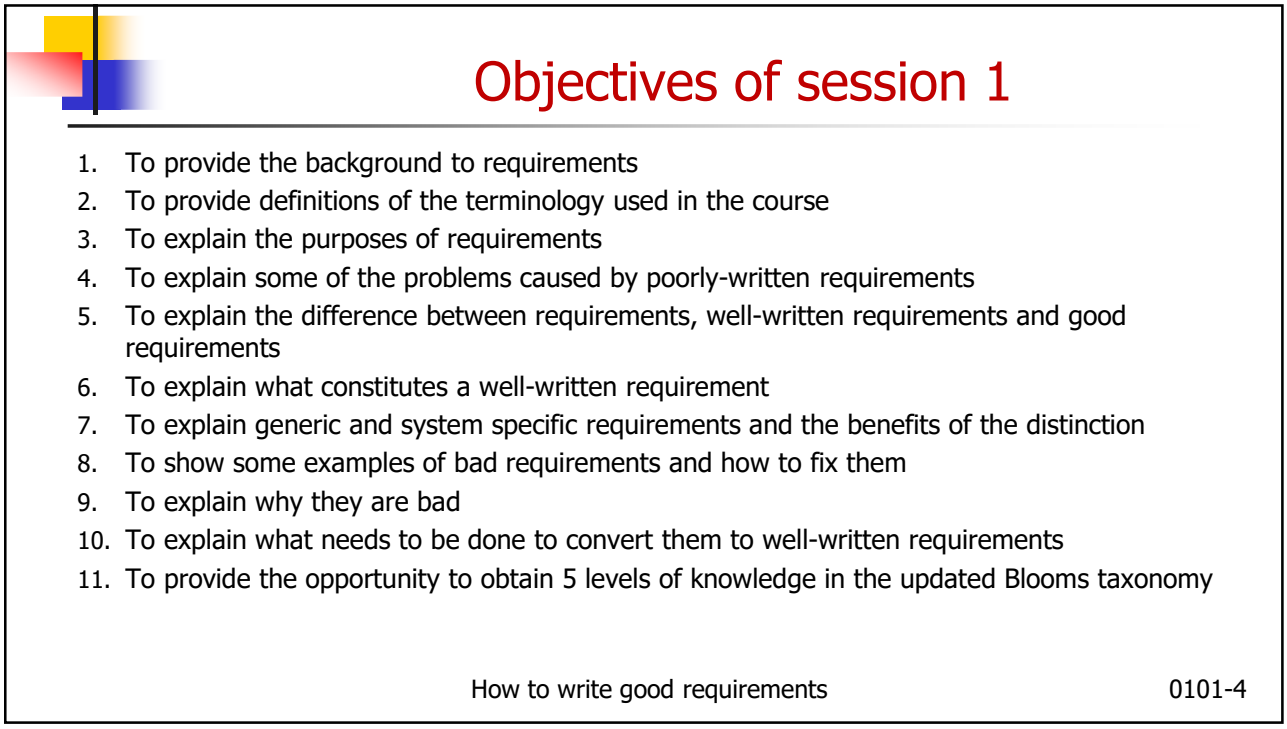
How to write good requirements
Session 1 of 10
Introduction to requirements

Version 1.0.2

Notice

- This session is being recorded
- By participating in the session, the participants agree:
  1. To be recorded
  2. Prof Kasser has the right, but not the obligation, to make the recording available to the participants
  3. Prof Kasser has the right to use the recordings in other classes (present and future)
- Prof Kasser agrees:
  1. To delay publication of the recording for at least 48 hours to allow participants to request, in writing, removal of remarks they made in the session (nature of remarks and approximate timing to be provided with the request).
  2. To remove those requested remarks before publishing the recordings


Course session topics
1. Introduction to requirements
2. Stakeholders and their importance
3. Communicating with the stakeholders
4. Converting stakeholder wants to needs
5. Documenting stakeholders' needs
6. Converting stakeholder needs to requirements
7. Converting requirements to well-written requirements
8. Converting well-written requirements to good requirements
9. The use of requirements in the rest of the system development process
10. Summary and closeout
How to write good requirements
0101-3


Objectives of session 1
1. To provide the background to requirements
2. To provide definitions of the terminology used in the course
3. To explain the purposes of requirements
4. To explain some of the problems caused by poorly-written requirements
5. To explain the difference between requirements, well-written requirements and good requirements
6. To explain what constitutes a well-written requirement
7. To explain generic and system specific requirements and the benefits of the distinction
8. To show some examples of bad requirements and how to fix them
9. To explain why they are bad
10. To explain what needs to be done to convert them to well-written requirements
11. To provide the opportunity to obtain 5 levels of knowledge in the updated Blooms taxonomy
How to write good requirements
0101-4

Knowledge components

- Lecture
  - Sets the context and provides overview
- Readings
  - 0102 Extracts from requirements document
- Exercises
  - 1-0 Tell us about yourself in less than 20 seconds
  - 1-1 Compliance to a poorly written requirement
  - 1-2 Requirements evaluation
  - 1-3 Knowledge reading 0202

Session topics

- **Exercise presentations**
- Background
- Glossaries and definitions
- Purpose of requirements
- Some of the problems caused by poorly-written requirements
- Well-written requirements
  - Attributes
  - Spelling and grammar
  - Structure of a requirement
  - Vocabulary of a requirement
- Generic and system specific requirements
- Examples of bad requirements and how to fix them
- Exercises



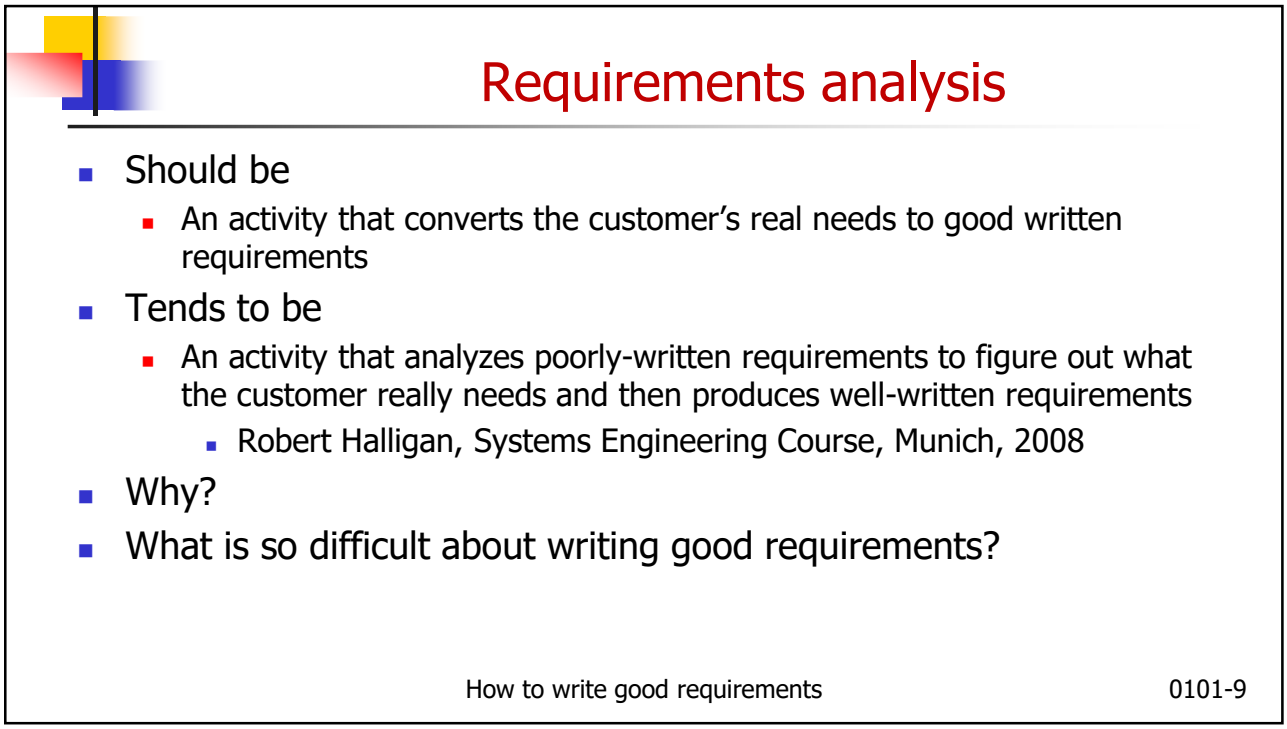
Exercise 1-0

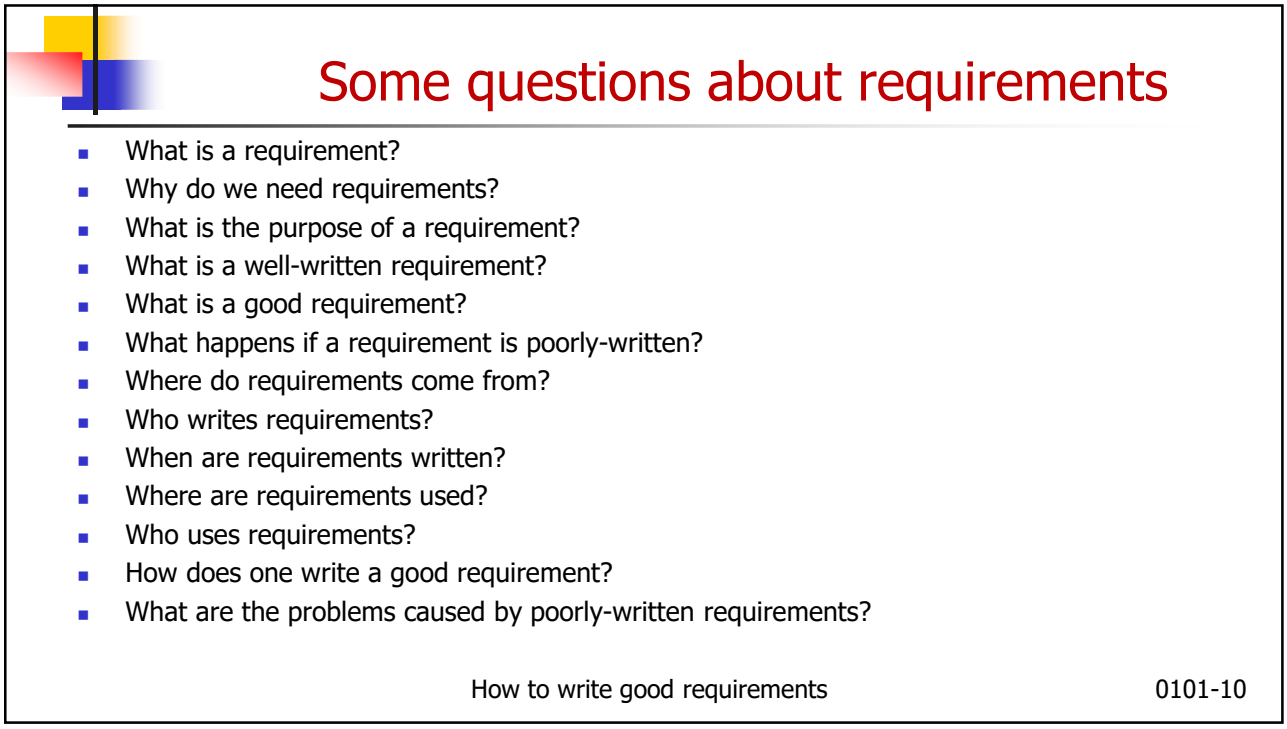
- Tell us about yourself in less than 20 seconds
  1. Who you are
  2. Where you are
  3. Why you are here
  4. What you want to learn
  5. Anything else you want to mention

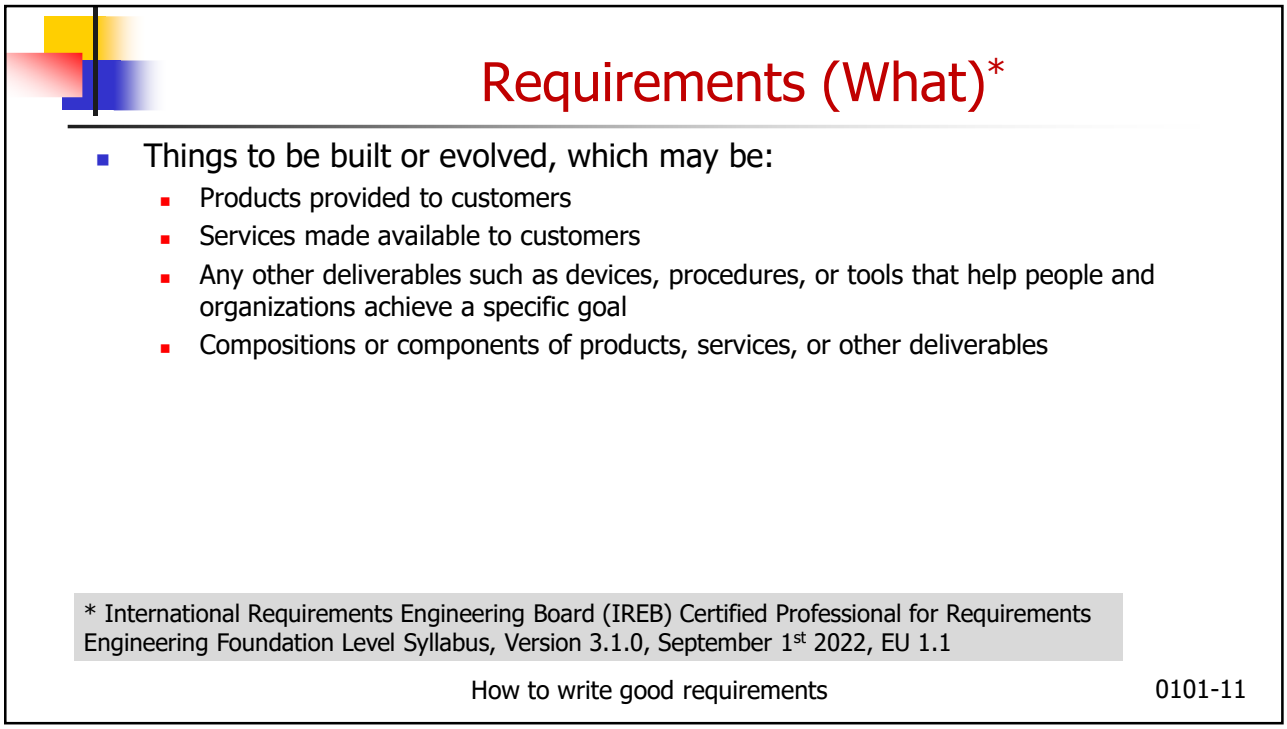
Session topics

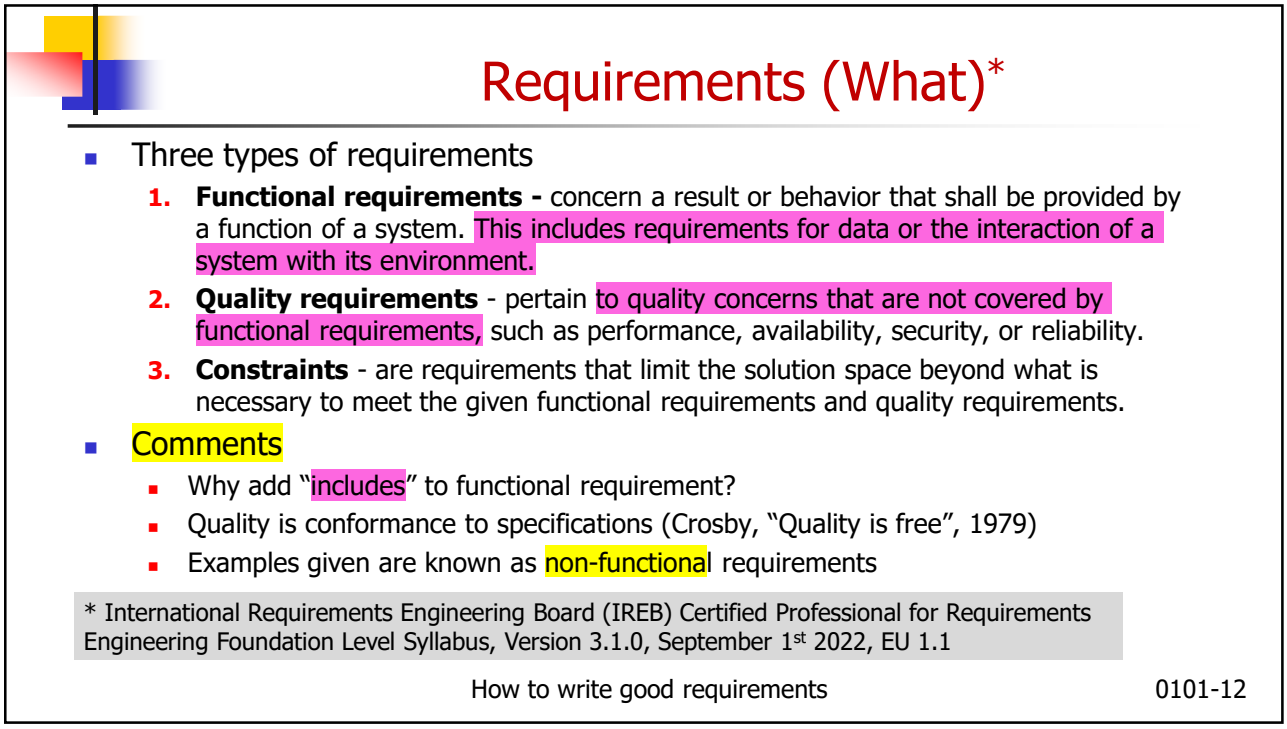
- Exercise presentations
- **Background**
- Glossaries and definitions
- Purpose of requirements
- Some of the problems caused by poorly-written requirements
- Well-written requirements
  - Attributes
  - Spelling and grammar
  - Structure of a requirement
  - Vocabulary of a requirement
- Generic and system specific requirements
- Examples of bad requirements and how to fix them
- Exercises

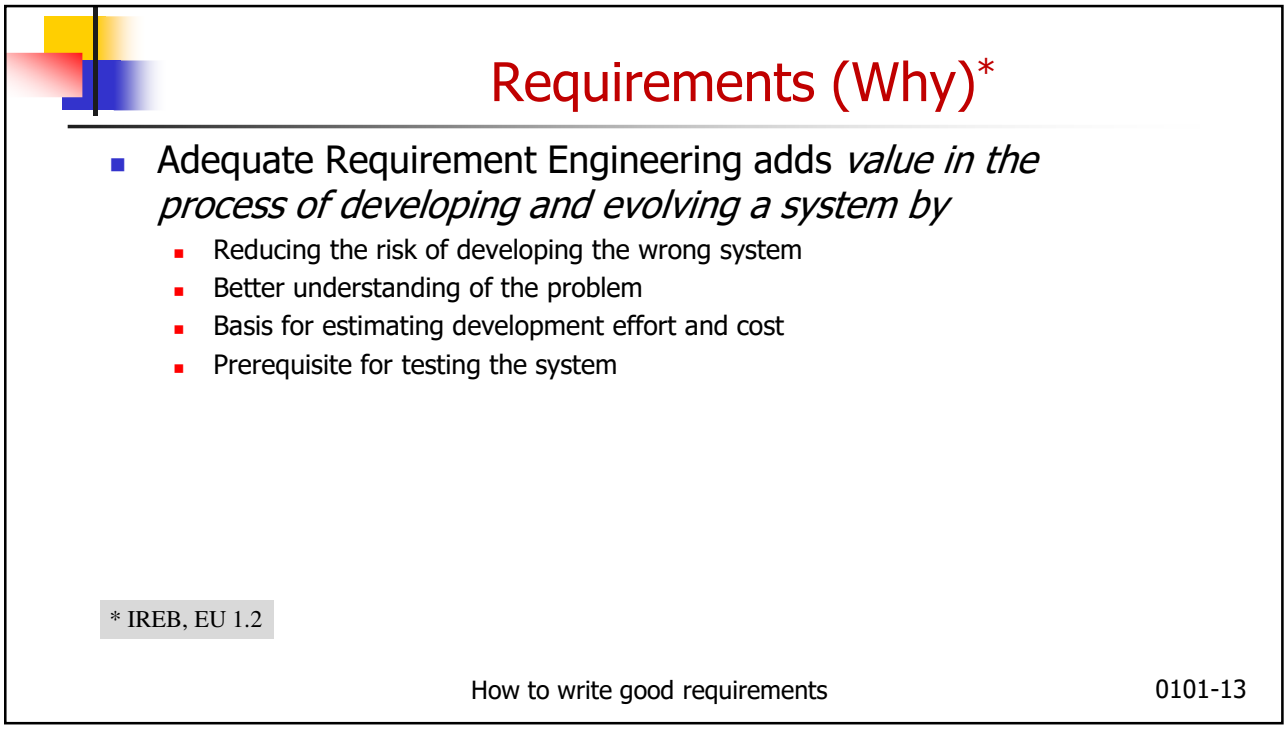


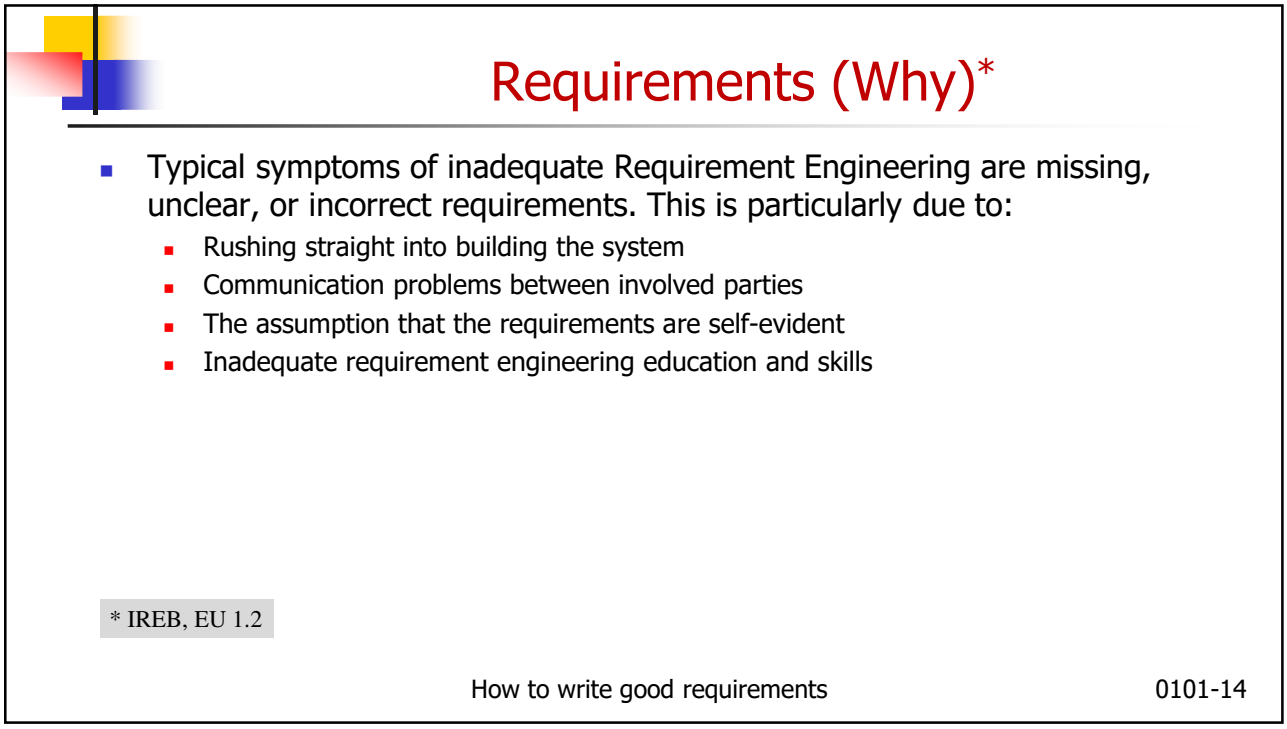

Requirements analysis
Should be
An activity that converts the customer's real needs to good written requirements
Tends to be
An activity that analyzes poorly-written requirements to figure out what the customer really needs and then produces well-written requirements
Robert Halligan, Systems Engineering Course, Munich, 2008
Why?
What is so difficult about writing good requirements?
How to write good requirements
0101-9


Some questions about requirements
What is a requirement?
Why do we need requirements?
What is the purpose of a requirement?
What is a well-written requirement?
What is a good requirement?
What happens if a requirement is poorly-written?
Where do requirements come from?
Who writes requirements?
When are requirements written?
Where are requirements used?
Who uses requirements?
How does one write a good requirement?
What are the problems caused by poorly-written requirements?
How to write good requirements
0101-10


Requirements (What)*
Things to be built or evolved, which may be:
Products provided to customers
Services made available to customers
Any other deliverables such as devices, procedures, or tools that help people and organizations achieve a specific goal
Compositions or components of products, services, or other deliverables
* International Requirements Engineering Board (IREB) Certified Professional for Requirements Engineering Foundation Level Syllabus, Version 3.1.0, September 1st 2022, EU 1.1
How to write good requirements
0101-11


Requirements (What)*
Three types of requirements
1. Functional requirements - concern a result or behavior that shall be provided by a function of a system. This includes requirements for data or the interaction of a system with its environment.
2. Quality requirements - pertain to quality concerns that are not covered by functional requirements, such as performance, availability, security, or reliability.
3. Constraints - are requirements that limit the solution space beyond what is necessary to meet the given functional requirements and quality requirements.
Comments
Why add "includes" to functional requirement?
Quality is conformance to specifications (Crosby, "Quality is free", 1979)
Examples given are known as non-functional requirements
* International Requirements Engineering Board (IREB) Certified Professional for Requirements Engineering Foundation Level Syllabus, Version 3.1.0, September 1st 2022, EU 1.1
How to write good requirements
0101-12


Requirements (Why)*
Adequate Requirement Engineering adds *value in the process of developing and evolving a system by*
Reducing the risk of developing the wrong system
Better understanding of the problem
Basis for estimating development effort and cost
Prerequisite for testing the system
* IREB, EU 1.2
How to write good requirements
0101-13


Requirements (Why)*
Typical symptoms of inadequate Requirement Engineering are missing, unclear, or incorrect requirements. This is particularly due to:
Rushing straight into building the system
Communication problems between involved parties
The assumption that the requirements are self-evident
Inadequate requirement engineering education and skills
* IREB, EU 1.2
How to write good requirements
0101-14

Requirements (Where)

- Requirements can occur as*
  - **System requirements** – what a system will do
  - **Stakeholder requirements** – what stakeholders want from their perspective
  - **User requirements** – what users want from their perspective
  - **Domain requirements** – required domain properties
  - **Business requirements** – business goals, objectives, and needs of an organization
- Comments
  - Focuses on wants, not needs
  - Users are not stakeholders in this section of the IREB document
  - "Requirements Engineering is about satisfying the stakeholders' desires and needs" (EU 2.1)
    - So user requirements are not satisfied by requirements engineering

Synonym?

???

* IREB, EU 1.3

How to write good requirements 0101-15

Universe of possible stakeholder requirements

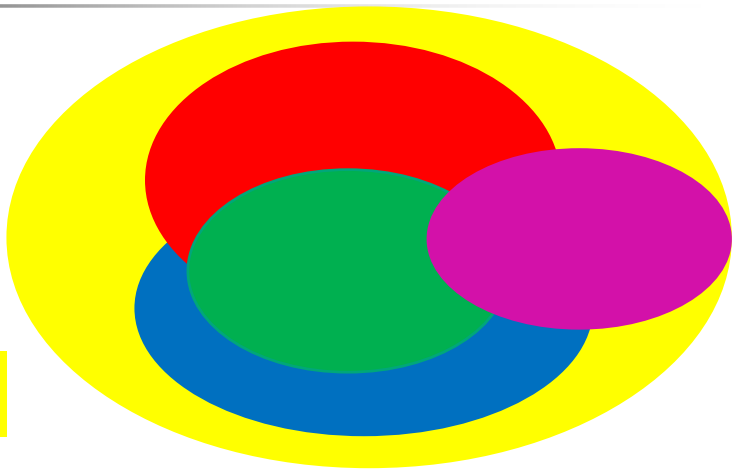
Wanted

Not wanted in this instance

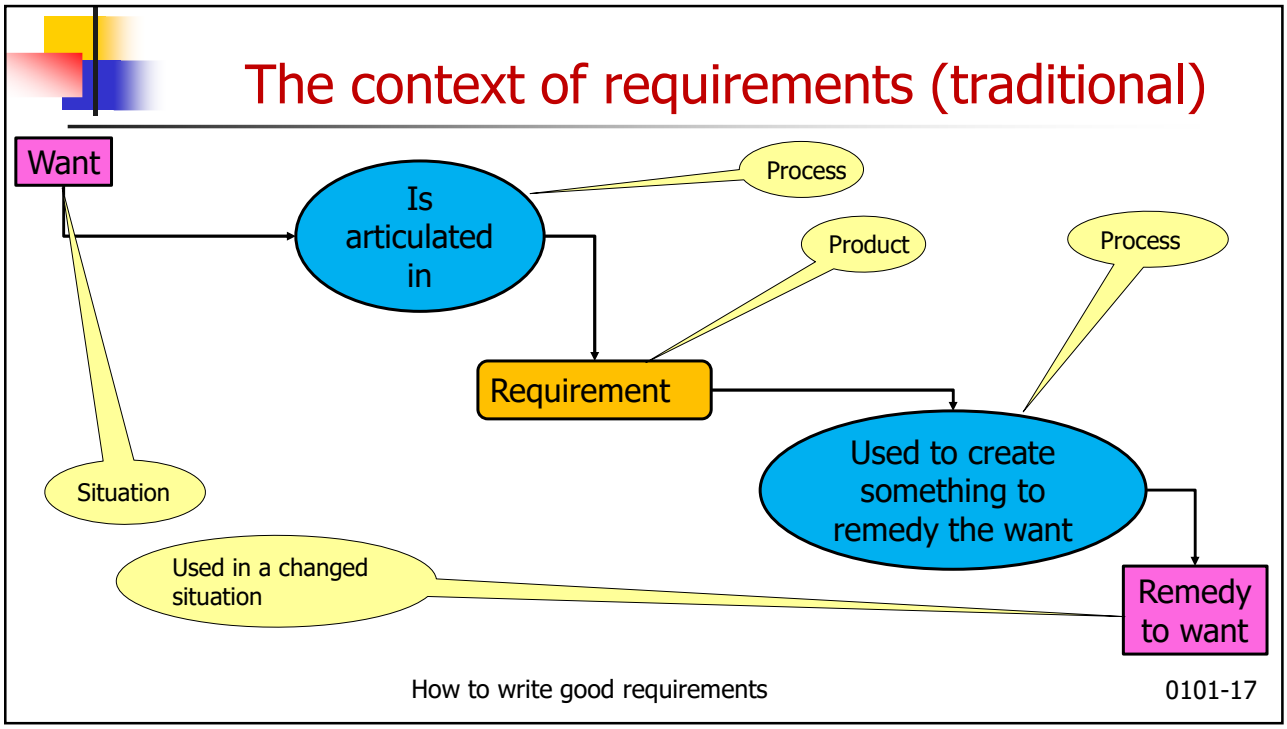
Don't care (not forgotten)

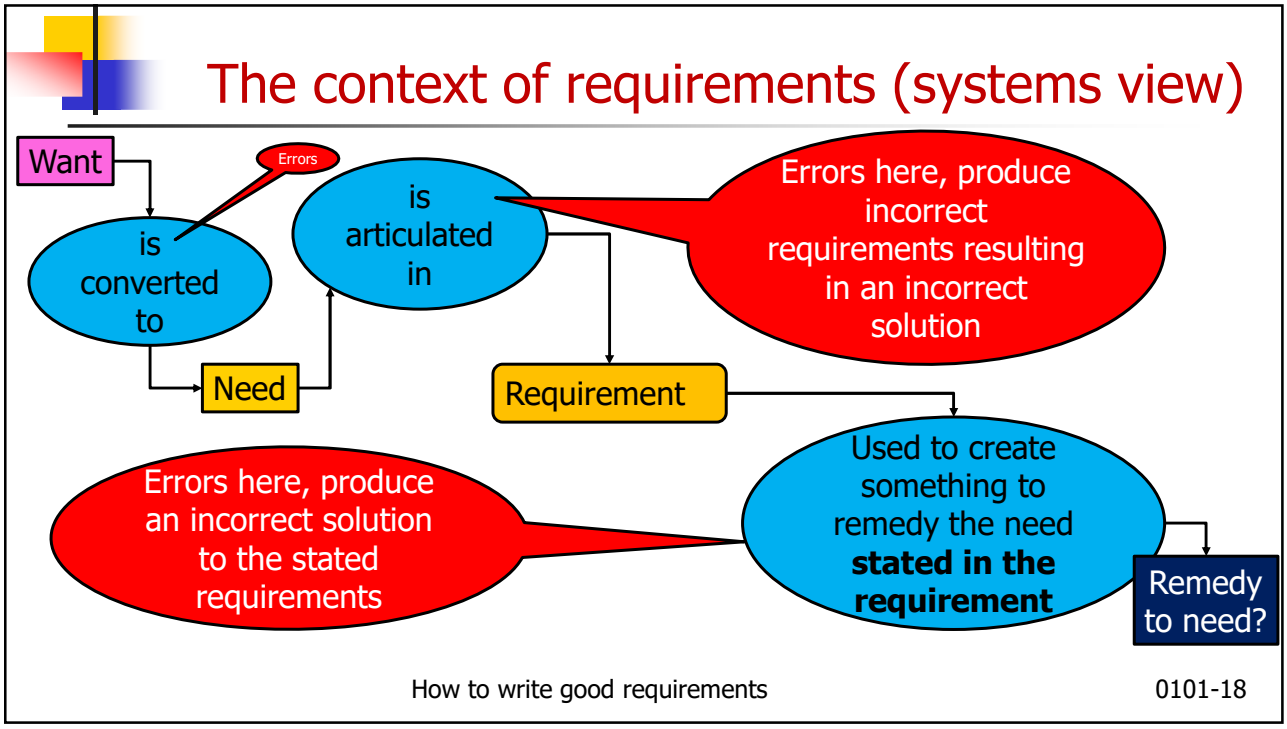
Can decide later

Don't care AND not relevant to the system



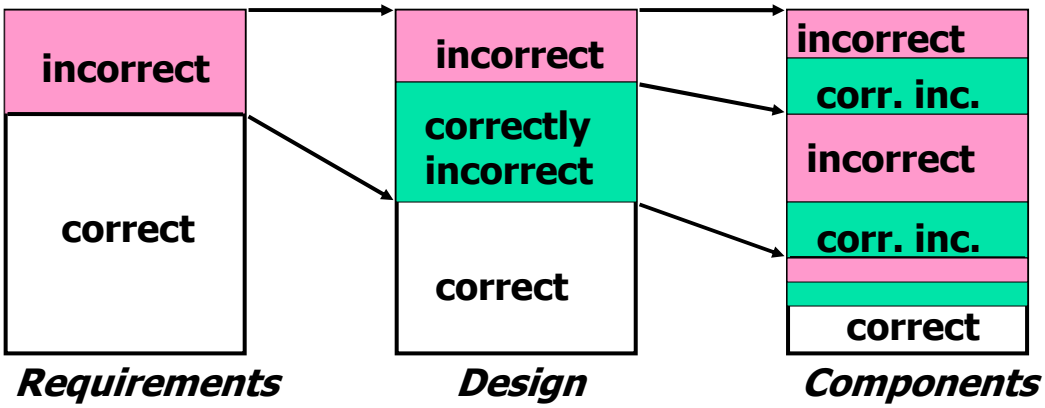
How to write good requirements 0101-16


The context of requirements (traditional)
Want
Is articulated in
Process
Product
Process
Requirement
Used to create something to remedy the want
Situation
Used in a changed situation
Remedy to want
How to write good requirements
0101-17


The context of requirements (systems view)
Want
Errors
is converted to
is articulated in
Errors here, produce incorrect requirements resulting in an incorrect solution
Need
Requirement
Errors here, produce an incorrect solution to the stated requirements
Used to create something to remedy the need **stated in the requirement**
Remedy to need?
How to write good requirements
0101-18

Error Propagation (different perspective)

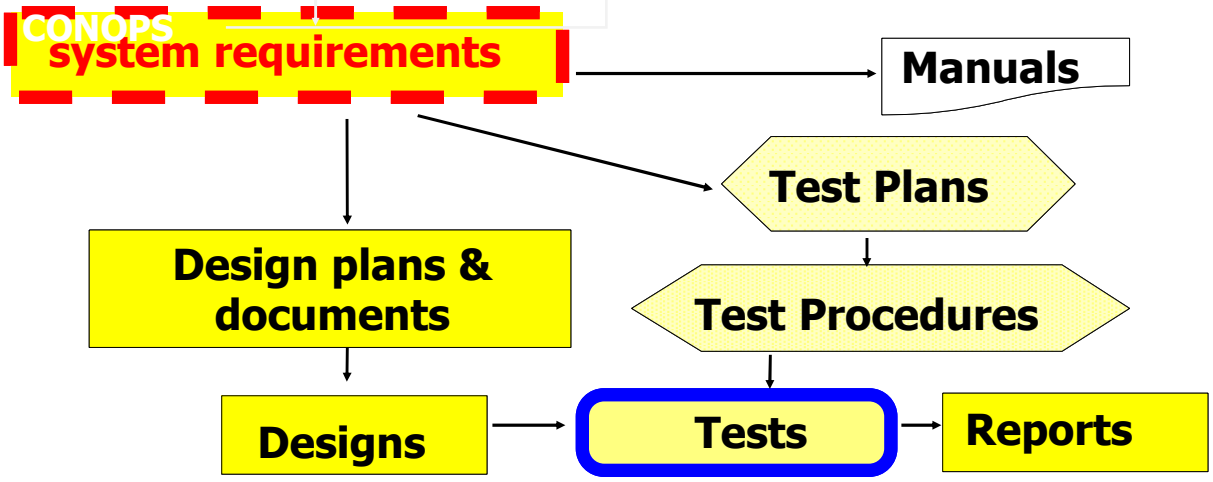
Deming - somebody gets paid to make defects



Correctly incorrect is what a poorly written specification tends to produce

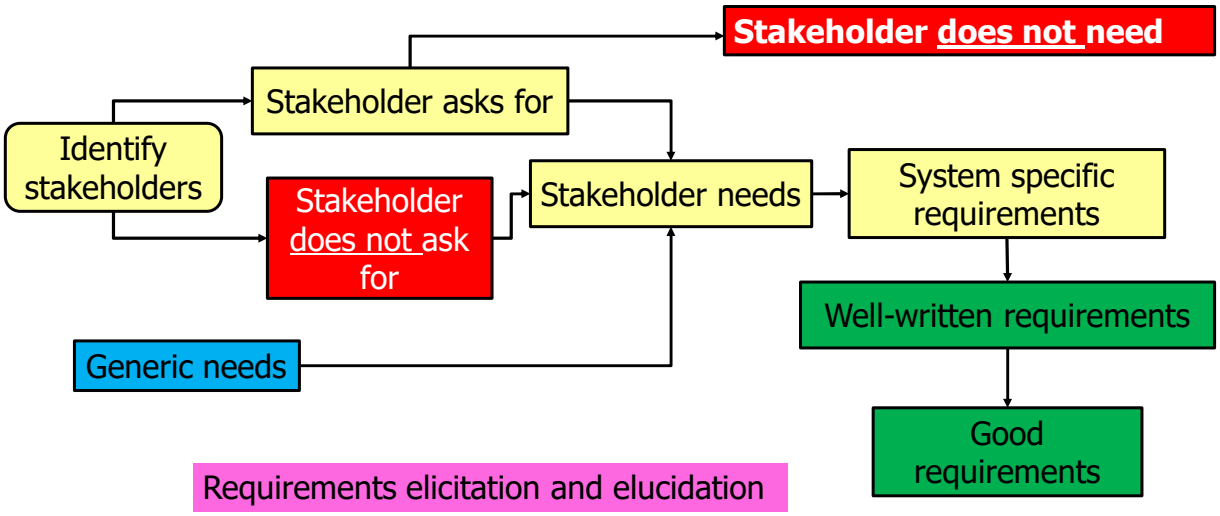
How to write good requirements 0101-19

Importance of good system requirements



How to write good requirements 0101-20

Workflow analysis for good requirements



How to write good requirements 0101-21

Session topics



- Exercise presentations
- Background
- **Glossaries and definitions**
- Purpose of requirements
- Some of the problems caused by poorly-written requirements
- Well-written requirements
  - Attributes
  - Spelling and grammar
  - Structure of a requirement
  - Vocabulary of a requirement
- Generic and system specific requirements
- Examples of bad requirements and how to fix them
- Exercises

How to write good requirements 0101-22

Glossaries*

- Mitigate the risk of a lack of a shared understanding of the terminology — that is, that some people interpret the same terms in different ways.
- Are a central collection of definitions for context-specific terms, everyday terms with a special meaning in the given context, abbreviations, and acronyms.
- Mark
  - **Synonyms** as different terms denoting the same thing
    - e.g. 'want' and 'desire'
  - **Homonyms** which use the same term for different meanings should be avoided

Systems approach discourages synonyms and homonyms

* Based on IREB EU 3.5

How to write good requirements 0101-23

Rules for Glossaries*

- Manage the glossary centrally
- Maintain the glossary over the entire course of the system development
- Define a person or small group who is responsible for the glossary
- Use a uniform style and structure for the glossary
- Involve the stakeholders and seek agreement about the terminology
- Make the glossary accessible for everybody involved
- Make the use of the glossary mandatory
- Check work products for proper glossary usage

Use same glossary across different projects

* IREB EU 3.5

How to write good requirements 0101-24

Definitions

- **Want** – something a stakeholder wants
- **Need** – something the stakeholder needs to remedy the problem
- **Requirement request** - A want/need in the form of a draft requirement that has not yet been accepted as a requirement. (*in this course, not a standard term*)
- **Stakeholder requirement** = a requirement request
- (textual) [system] **Requirement** - "A statement that identifies a product or process operational, functional, or design characteristic or constraint, which is unambiguous, testable or measurable, and necessary for product or process acceptability (by consumers or internal quality assurance guidelines)", (IEEE, 1998)
- **Well-written requirement** – a requirement that meets the grammatical and vocabulary requirements for requirement statements
- **Good requirement** – a well-written, feasible to implement, requirement for which the customer is willing to pay

How to write good requirements 0101-25

More definitions

- **Stakeholder** - A person or organization who influences a system's requirements or who is impacted by that system (IREB Glossary, version 2.01)
  - Focus on future
- **Stakeholder** - people or organizations that are internal or external to the project who have a vested interest in its success or failure
- **Contractor** – stakeholder who contractually represents the personnel performing the project
- **Customer** – the stakeholder who pays for the project
- **Specification** - a document containing a collection of requirements
- **Capability** - the quality or state of being capable (ability) (https://www.merriam-webster.com/dictionary/capability#dictionary-entry-1, accessed 15 August 2023)
- **Function** – an action or activity

How to write good requirements 0101-26

Even more definitions

- **Attribute** - a functional or non-functional aspect of the system that exists
  - E.g. speed
- **Property** - a quality or trait belonging to an attribute
  - E.g. speed of <= 80 k/hr.
- **Generic requirement** – a requirement for something about the class of system, an attribute of the system
  - E.g. the nut and bolt in the system shall not loosen under vibration
- **System (specific) requirement** - a requirement for the specific system
  - E.g. the specific details of the vibration and the degree of loosening allowed
- **Conceptual model** - An imaginary representation of a situation/system
- **Situation** – the context of a socio-technical system. A system in its own right. Often known as the meta- or containing system
- **System** - An abstraction (from the real world) of a set of objects, each at some level of decomposition, at some period of time, enclosed in a subjective boundary crafted for a purpose.

Even more definitions

- **Mission requirement** – a requirement that must be performed or met for the system to perform its mission
- **Support requirement** – a requirement that must be met to ensure the system is able to perform its mission, as and when needed
  - E.g. requirement for reliability, availability, maintainability, etc.

System Requirements Engineering

- The science and discipline concerned with **analysing** and **documenting** system requirements
- Transforms an operational need into -
  - a system description
  - system performance parameters
  - system configuration
  - Through an iterative process of
    - definition, analysis, trade-off studies, prototyping and simulation
- Comment
  - Doesn't transform need to requirements
  - Can be interpreted as parameters don't need to be stated as text mode requirements

How to write good requirements 0101-29

Session topics



- Exercise presentations
- Background
- Glossaries and definitions
- **Purpose of requirements**
- Some of the problems caused by poorly-written requirements
- Well-written requirements
  - Attributes
  - Spelling and grammar
  - Structure of a requirement
  - Vocabulary of a requirement
- Generic and system specific requirements
- Examples of bad requirements and how to fix them
- Exercises

How to write good requirements 0101-30

Purposes

- **Stakeholder requirements** – to communicate a stakeholder want for X (which may or may not be needed, feasible or affordable) to a contractor
- **(system) Requirements** - to communicate (system) requirements for X to a contractor who is paid to deliver the X (on or before a specific date and time) for which the customer is willing to pay
- **Well-written requirements** - to communicate **requirements** for X in an unambiguous and verifiable manner to a contractor for which the customer is willing to pay
- **Good requirements** - to communicate **well-written requirements for X**, which are affordable and feasible to implement, AND for which the customer is willing to pay, in a complete, consistent, unambiguous and verifiable manner

How to write good requirements 0101-31

Session topics

- Exercise presentations
- Background
- Glossaries and definitions
- Purpose of requirements
- **Some of the problems caused by poorly-written requirements**
- Well-written requirements
  - Attributes
  - Spelling and grammar
  - Structure of a requirement
  - Vocabulary of a requirement
- Generic and system specific requirements
- Examples of bad requirements and how to fix them
- Exercises



How to write good requirements 0101-32

Some of the problems caused by poorly written requirements

- Requirements fail to reflect the real needs of the customer
- Requirements are inconsistent, incomplete and/or contradictory
- Increases in cost, delays in schedule
  - It is expensive to make changes to requirements after they have been agreed (contractual impediment)
- Misunderstandings and/or misinterpretations of the requirements between customers, those developing system requirements and the people developing or maintaining the system
- Lack of technical knowledge can lead to requirements that are unreasonably expensive to implement

How to write good requirements 0101-33

Problems to be overcome

- People can't always communicate their needs
- People don't always know what they need
- People don't always know the cost of meeting a stated need
- People state requirements in their language
- Analysts, customers, users, developers, and other stakeholders don't always speak the same language
- Requirements from different sources may conflict
- Organisational and political issues drive programmatic requirements
- Insufficient time is allocated for system development
- Stated requirements may not be real
  - hidden agendas
- Requirements change over time
- Priorities change over time

How to write good requirements 0101-34

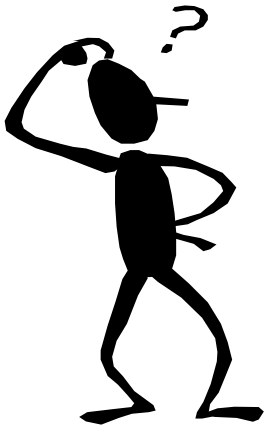
Session topics



- Exercise presentations
- Background
- Glossaries and definitions
- Purpose of requirements
- Some of the problems caused by poorly-written requirements
- **Well-written requirements**
  - **Attributes**
  - Spelling and grammar
  - Structure of a requirement
  - Vocabulary of a requirement
- Generic and system specific requirements
- Examples of bad requirements and how to fix them
- Exercises

How to write good requirements 0101-35

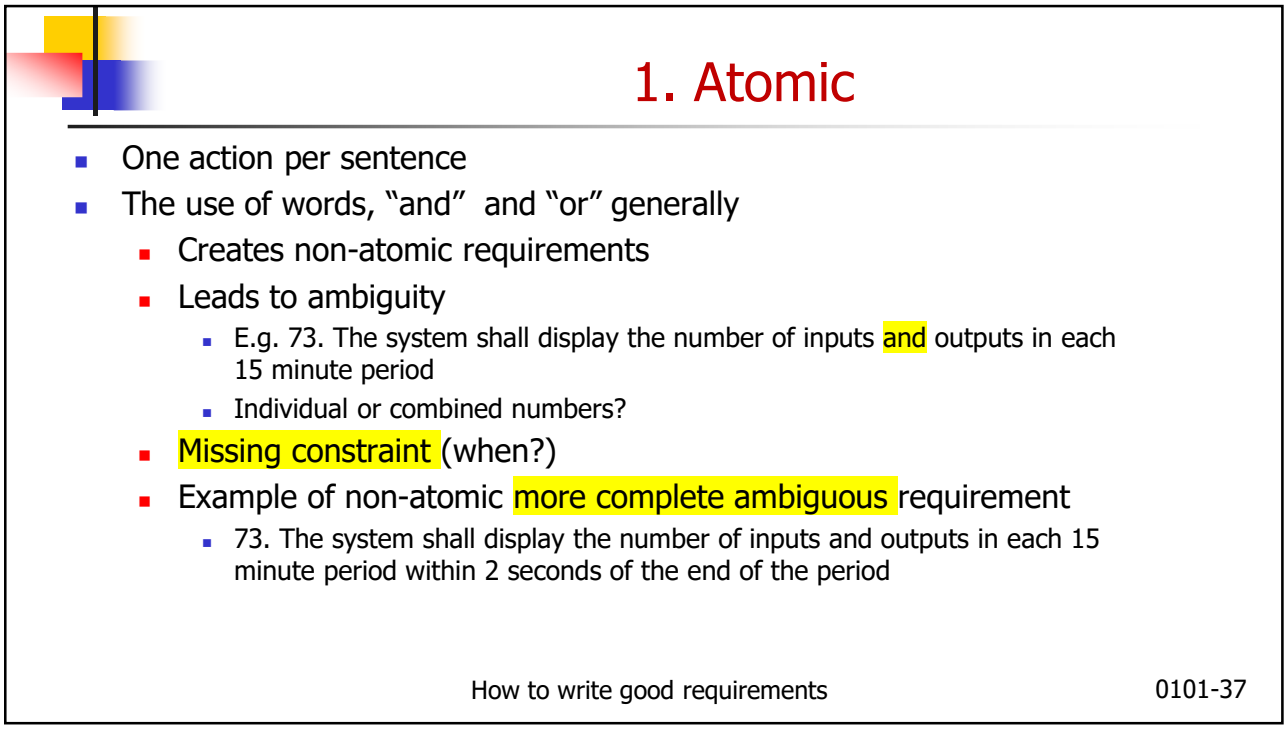
Six attributes of a well-written requirement

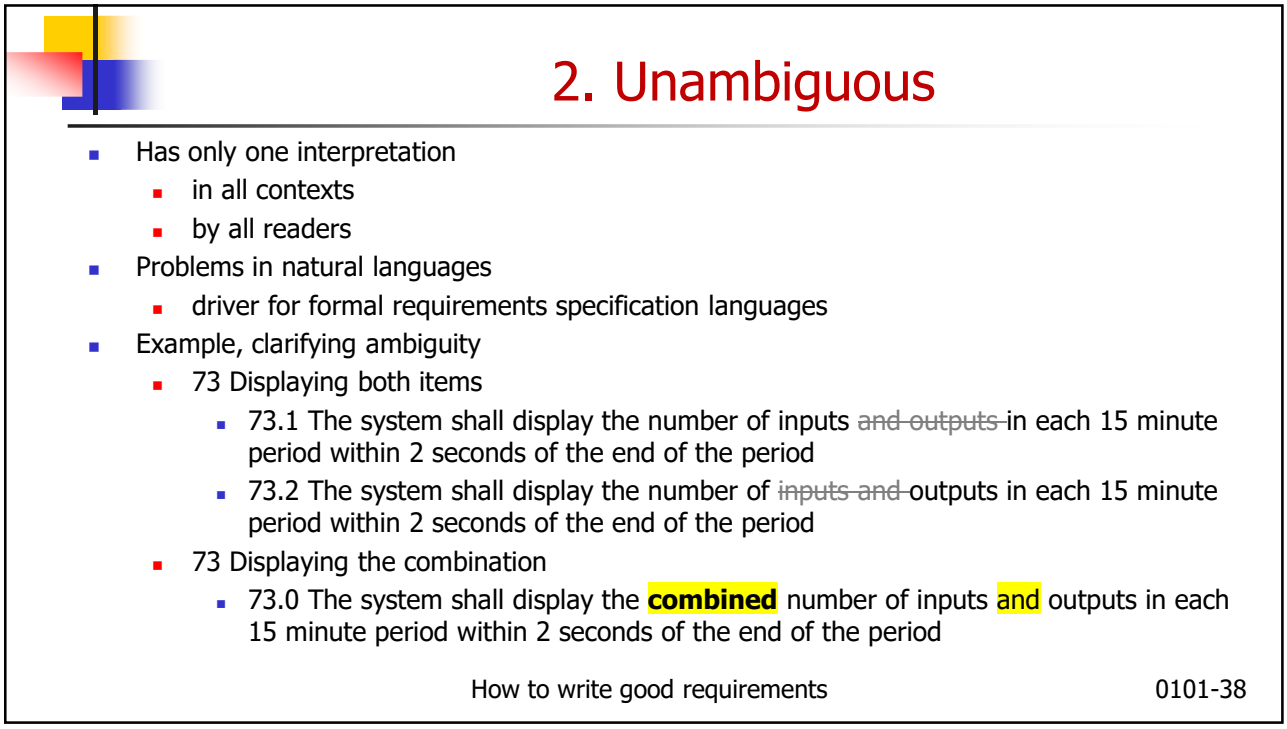


1. Atomic
2. Unambiguous
3. Verifiable
4. Consistent
5. Understandable*
6. Adequate*
7. ~~Linked or traced~~
   1. is an attribute of a 'requirement' but does not show up in the statement
   2. Is an attribute of a good requirement

* IREB, EU 3.8

How to write good requirements 0101-36


1. Atomic
One action per sentence
The use of words, "and" and "or" generally
Creates non-atomic requirements
Leads to ambiguity
E.g. 73. The system shall display the number of inputs and outputs in each 15 minute period
Individual or combined numbers?
Missing constraint (when?)
Example of non-atomic more complete ambiguous requirement
73. The system shall display the number of inputs and outputs in each 15 minute period within 2 seconds of the end of the period
How to write good requirements
0101-37


2. Unambiguous
Has only one interpretation
in all contexts
by all readers
Problems in natural languages
driver for formal requirements specification languages
Example, clarifying ambiguity
73 Displaying both items
73.1 The system shall display the number of inputs and outputs in each 15 minute period within 2 seconds of the end of the period
73.2 The system shall display the number of inputs and outputs in each 15 minute period within 2 seconds of the end of the period
73 Displaying the combination
73.0 The system shall display the combined number of inputs and outputs in each 15 minute period within 2 seconds of the end of the period
How to write good requirements
0101-38

Can you see the ambiguity?

- Help victims?
- Offering the service?



How to write good requirements 0101-39

3. Verifiable

- Every requirement shall be verifiable by a cost effective method
  - Well-written ~~test~~ verification procedure
  - Stated in specific terms
- Verification shall be by
  - Test or demonstration
  - Analysis
  - Simulation
  - Other, as agreed by customer and contractor
- Defined acceptance criteria



How to write good requirements 0101-40

4. Consistent

- No two or more requirements shall describe the same something using different terminology
  - No synonyms
- Requirements shall not contradict each other
  - Logical
  - Temporal
  - Conflicting instructions

How to write good requirements 0101-41

5 and 6. Understandable and adequate

202.1 DADS shall statistically monitor the integrity of data stored in the archive and safe-store in order to detect degrading media.

- Understandable and adequate at the time it was written
- But 18 months later after staff turnover?
  - No data dictionary or glossary

- Define the following terms
  - Statistically monitor
  - Integrity
  - Degrading media

How to write good requirements 0101-42

Session topics



- Exercise presentations
- Background
- Glossaries and definitions
- Purpose of requirements
- Some of the problems caused by poorly-written requirements
- **Well-written requirements**
  - Attributes
  - **Spelling and grammar**
  - **Structure of a requirement**
  - Vocabulary of a requirement
- Generic and system specific requirements
- Examples of bad requirements and how to fix them
- Exercises

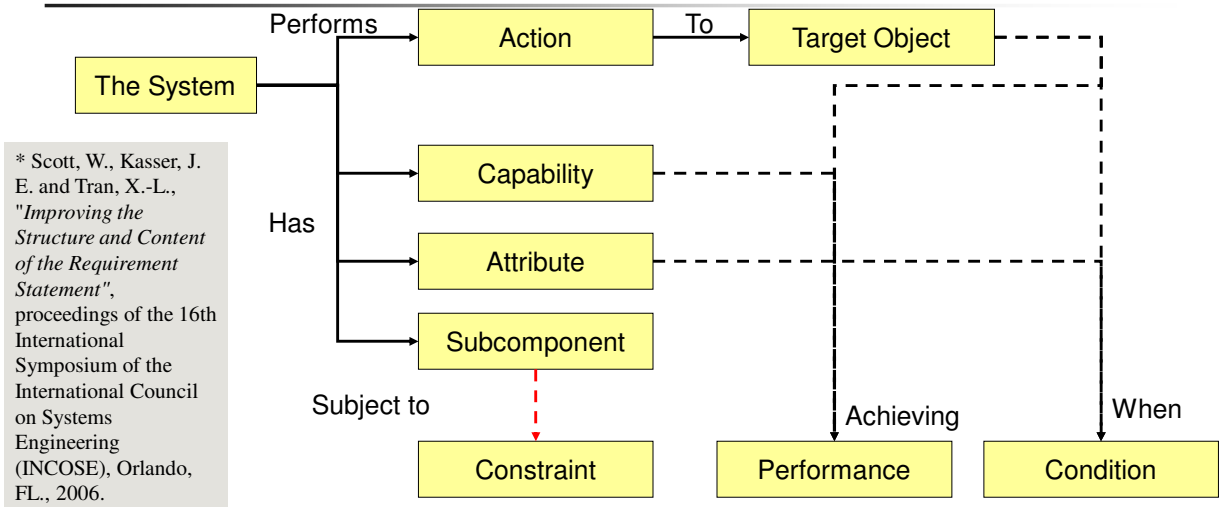
How to write good requirements 0101-43

1. Spelling and grammar

- Spelling
  - What
    - The correct arrangement of letters to form the words you mean to use to communicate concepts
  - Why
    - Accurate spelling is essential for effective communication, as incorrect spelling can lead to misunderstandings and confusion
- Grammar
  - What
    - The structure and rules of how words are combined to form sentences and convey meaning
  - Why
    - Proper grammar ensures that sentences convey the intended message

How to write good requirements 0101-44

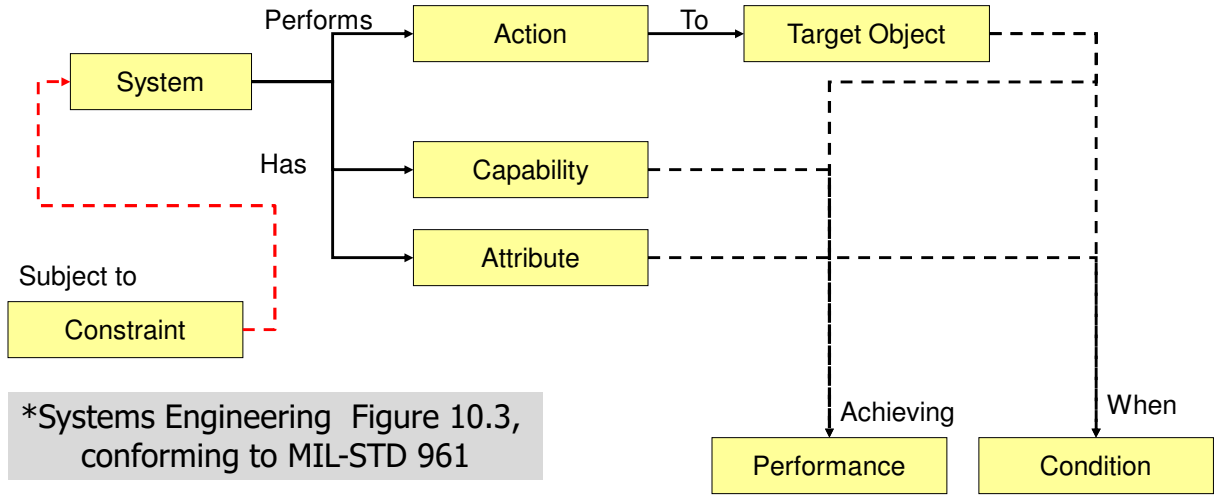
2. The structure of a requirement*



* Scott, W., Kasser, J. E. and Tran, X.-L., "*Improving the Structure and Content of the Requirement Statement*", proceedings of the 16th International Symposium of the International Council on Systems Engineering (INCOSE), Orlando, FL., 2006.

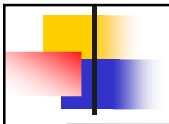
How to write good requirements 0101-45

The structure of a well-written requirement*



*Systems Engineering Figure 10.3, conforming to MIL-STD 961

How to write good requirements 0101-46



Requirement statement

A single imperative instruction that complies with the "*requirements for writing requirements*", containing the following elements:

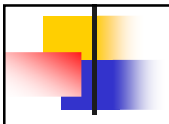
Traditional order

- ID number
- Subject
- Verb
- Target object
- Qualifiers (if any)
- Conditions (if any)

Well-written approach

- Conditions (if any)
- ID number
- Subject
- Verb
- Target object
- Qualifiers (if any)

How to write good requirements 0101-47



Traditional requirements

Bus operations

1.2.1 The buses shall operate on their specific routes from 0530 to 2300 daily except on Friday and Saturday

1.2.2 The buses shall operate on their specific routes from 0530 to 1 hour before sunset on Fridays

1.2.3 The buses shall operate on their specific routes from 1 hour after sunset to 2330 on Saturdays

How to write good requirements 0101-48

Non-functional requirements

- The state of being requirements
  - "ilities"
- Availability can be written as:
  - The system shall be operational for 24 hours each day
    - What shall be **done**
  - The Availability of the system shall be 24 hours daily
    - What shall **be**
- Acceptance criteria in this situation are generally a combination of reliability and mean time to repair
- Safety
  - Combination of "built into the system" and procedures
  - Risk procedure will be overridden
- Require safety to be "built into the system" rather than rely on procedures

Ensure system can meet the functional requirements as and when required

How to write good requirements 0101-49

Session topics

- Exercise presentations
- Background
- Glossaries and definitions
- Purpose of requirements
- Some of the problems caused by poorly-written requirements
- **Well-written requirements**
  - Attributes
  - Spelling and grammar
  - Structure of a requirement
  - **Vocabulary of a requirement**
- Generic and system specific requirements
- Examples of bad requirements and how to fix them
- Exercises



How to write good requirements 0101-50

Commonly used words and phrases - Alternative course of action*

- "Unless otherwise specified"
  - Clarify the expression by providing a reference (e.g. in Section 6).
  - Used at the start of the sentence
  - Example
    - Unless otherwise specified in Section 6, the buses shall operate on their specific routes from 0530 to 2300 daily
- "Except"
  - Implies unless otherwise specified

* MIL-STD-961B (31 May 1985) Section 4.9.5

Commonly used words and phrases - Verbs

- Requirement for use of verbs is originally from the era of paper documents and extracted from MIL-STD-961B (31 May 1985) Section 4.9.6, (accessed on 31 July 2023 from http://everyspec.com/)
- "Shall", requirement is intended to express a provision that is binding
- "Will" - used to state
  - a declaration of purpose (goal) on the part of the specifier
  - simple future tense
- "Should" and "may" - non-mandatory provisions
  - Inclusion of non-mandatory requirements is confusing and leads to errors
  - Including them as requirements (shall) but with low priorities makes them and their need unambiguous
- "Must" shall not be used
  - But may be used in a requirement request (signifying it has not yet been accepted as a requirement) if defined in the glossary and used consistently

Commonly used words and phrases - Limits

- Upper limit
  - "The diameter shall be less than 2 cm" (<2.00)
  - "The diameter shall not be greater than 2 cm " (<=2.00)
- Lower limit
  - "The diameter shall be greater than 0.5 cm" (>0.5)
  - "The diameter shall not be less than 0.5 cm" (>=0.5)
- Do not use error prone words
  - "The diameter shall not be less than 0.5 cm " (but what if you want >=0.5)
    "The diameter shall be greater **or** equal to 0.5 cm"
- Beware of specifying exact numbers. Always add a 'tolerance' or reference
  - 2 ± xx cm
  - 0.5 ± xx cm
  - Tolerance shall be as tight as needed ~~and no more~~

Commonly used words and phrases – vague and ambiguous*

- Except when the term is unambiguous,
  the terms "and/or", "etc.", "e.g." and "i.e.' shall not be used in specifications*
- Example
  - The diameter shall be greater **or** equal to 0.5 cm" (>=0.5)

* MIL-STD-961B Section 4.9.7

Commonly used words and phrases – Citations*

- Referenced documents shall be cited as:*
  - "conforming to"
  - "as specified in"
  - "in accordance with"
- Be consistent
  - Use same wording in each requirement
- Cite applicable section, not entire document
  - You may not want other parts of document to level requirements
- Add version/date of referenced document to citation
  - Documents are revised and changes may affect whatever you are citing without you knowing

* MIL-STD-961B (31 May 1985) Section 4.9.5

How to write good requirements 0101-55

Session topics



- Exercise presentations
- Background
- Glossaries and definitions
- Purpose of requirements
- Some of the problems caused by poorly-written requirements
- Well-written requirements
  - Attributes
  - Spelling and grammar
  - Structure of a requirement
  - Vocabulary of a requirement
- **Generic and system specific requirements**
- Examples of bad requirements and how to fix them
- Exercises

How to write good requirements 0101-56

The difference between "system generic" and "system specific" requirements

- System generic requirements
  - Come from the system generic attributes
  - Are specific system independent
  - Maximize the probability of creating a complete set of requirements
  - Provide an inherited template to quantify system generic attributes
- System specific" requirements are the instance of the system generic requirements (properties) for the system specific attributes

The benefits of using "system generic" and "system specific" requirements

- Requirements state the attributes and properties of something needed
- Example
  - System generic requirement for operating temperate of system is -10 to +50 degrees centigrade
  - System specific requirement for operating temperate of system in application is +5 to +50 degrees centigrade
- Prompts stakeholders not to forget operating temperature
- Instead of thinking about the needed temperature, stakeholder can think if generic value is needed or needs to be changed in specific application

Examples of generic and mission specific requirements

- Generic
  - Product requirements
    - Mission requirements
    - Support requirements
      - Safety requirements
      - Class of system
      - Foisted on project by laws and regulations
  - Process requirements
    - Foisted on project by laws and regulation, e.g. Standards, e.g. ISO9001.
    - Workplace constraints, e.g. in-house tools, etc.
- Mission specific
  - Product requirements
    - Specific mission and support requirements
    - Manufacturing requirements
    - Delivery requirements
  - Process requirements
    - Implementation requirements

How to write good requirements 0101-59

Some sources of generic functions and requirements

- Similar systems
- Government mandates
- Domain knowledge
- Standards
- Relevant documents
- Market studies
- Legacy system dictates
- Bodies of knowledge for solution system in operation
  - E.g., Business Analysis Body of Knowledge (BABOK)
- Existing and legacy systems*
- Process documents*
- Legal or regulatory documents*
- Company specific regulations*
- (marketing) information about potential future users*
- Similar situations in different domains*
- Guideline documents

* IREB, 3.10., 4.1

How to write good requirements 0101-60

Session topics

- Exercise presentations
- Background
- Glossaries and definitions
- Purpose of requirements
- Some of the problems caused by poorly-written requirements
- Well-written requirements
  - Attributes
  - Spelling and grammar
  - Structure of a requirement
  - Vocabulary of a requirement
- Generic and system specific requirements
- **Examples of bad requirements and how to fix them**
- Exercises



How to write good requirements 0101-61

Example 1 [Communications system]

1234.5 An objective requirement is that the downlink control shall be flexible enough to allocate extra downlink time slots for high-priority participants if more downlink data is required from a particular participant.

1234.5 ~~An objective requirement is that~~ the downlink control shall be flexible enough to allocate extra downlink time slots for high-priority participants if more downlink data is required from a particular participant.

How to write good requirements 0101-62

Example 2 [ST-DADDS]

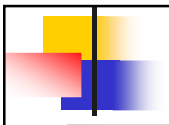
"204.1 DADS shall automatically maintain *statistics* concerning the number of times and the most recent time that each data set has been accessed. These same statistics shall be maintained for each *piece of media* in the DADS archive" (ST-DADS 1992)

- The requirement contains:
  - ***Undefined terms:*** statistics, piece of media.
  - ***Multiple requirements:*** and, two sentences.

Example 2 [ST-DADDS]

"204.1 DADS shall automatically maintain *statistics* concerning the number of times and the most recent time that each data set has been accessed. These same statistics shall be maintained for each *piece of media* in the DADS archive" (ST-DADS 1992)

- The requirement contains:
  - ***Undefined terms:*** statistics, piece of media.
  - ***Multiple requirements:*** and, two sentences.



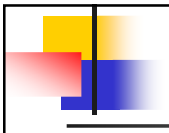
Creating four atomic requirements

204.1a DADS shall automatically maintain statistics concerning the number of times ~~and the most recent time~~ that each data set has been accessed. ~~These same statistics shall be maintained for each piece of media in the DADS archive~~.

204.1b DADS shall automatically maintain statistics concerning the ~~number of times and~~ the most recent time that each data set has been accessed. ~~These same statistics shall be maintained for each piece of media in the DADS archive.~~

204.1c DADS shall automatically maintain statistics concerning the number of times ~~and the most recent time~~ that ~~each data set has been accessed. These same statistics shall be maintained for~~ each piece of media in the DADS archive has been accessed.

204.1d DADS shall automatically maintain statistics concerning the ~~number of times and~~ the most recent time that ~~each data set has been accessed. These same statistics shall be maintained for~~ each piece of media in the DADS archive has been accessed.



Four atomic requirements

204 Access to data sets

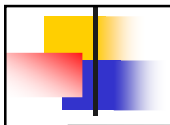
204.1a DADS shall automatically maintain statistics concerning the number of times that each data set has been accessed.

204.1b DADS shall automatically maintain statistics concerning the most recent time that each data set has been accessed.

204 Access to each piece of media

204.1c DADS shall automatically maintain statistics concerning the number of times that each piece of media in the DADS archive has been accessed.

204.1d DADS shall automatically maintain statistics concerning the most recent time that each piece of media in the DADS archive has been accessed.



Four atomic requirements

204 Access to data sets

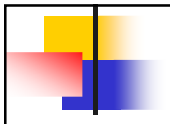
204.1a DADS shall automatically maintain statistics concerning the number of times that each data set has been accessed.

204.1b DADS shall automatically maintain statistics concerning the most recent time that each data set has been accessed.

Statistics

204.1c DADS shall automatically maintain statistics concerning the number of times that each piece of media in the DADS archive has been accessed.

204.1d DADS shall automatically maintain statistics concerning the most recent time that each piece of media in the DADS archive has been accessed.



Example 4 [ST-DADDS]

509.1 DADS shall monitor and provide reports (to the operator) on all requests for DADS products and services. This capability shall include recording the name and organization of the requester, the product or service requested, the date and time of the request, the service priority, the current disposition of the request, and the date and time of service completion.

When you write or approve (sign off) on a requirement, you are demonstrating your level of competency in requirements engineering

Better poorly-written requirement

509.1 DADS shall monitor and provide reports to the operator about the state of transactions in the system. This capability shall contain the following:

a. the name and organisation of the requester
b. the product or service requested
c. the date and time of the request
d. the service priority
e. the current disposition of the request
f. the date and time of service completion.

Better poorly-written requirement

509.1 DADS shall monitor *and* provide reports to the operator about the state of transactions in the system. This capability shall contain the following:

a. the name and organisation of the requester
b. the product or service requested
c. the date and time of the request
d. the service priority
e. the current disposition of the request
f. the date and time of service completion.

Session topics



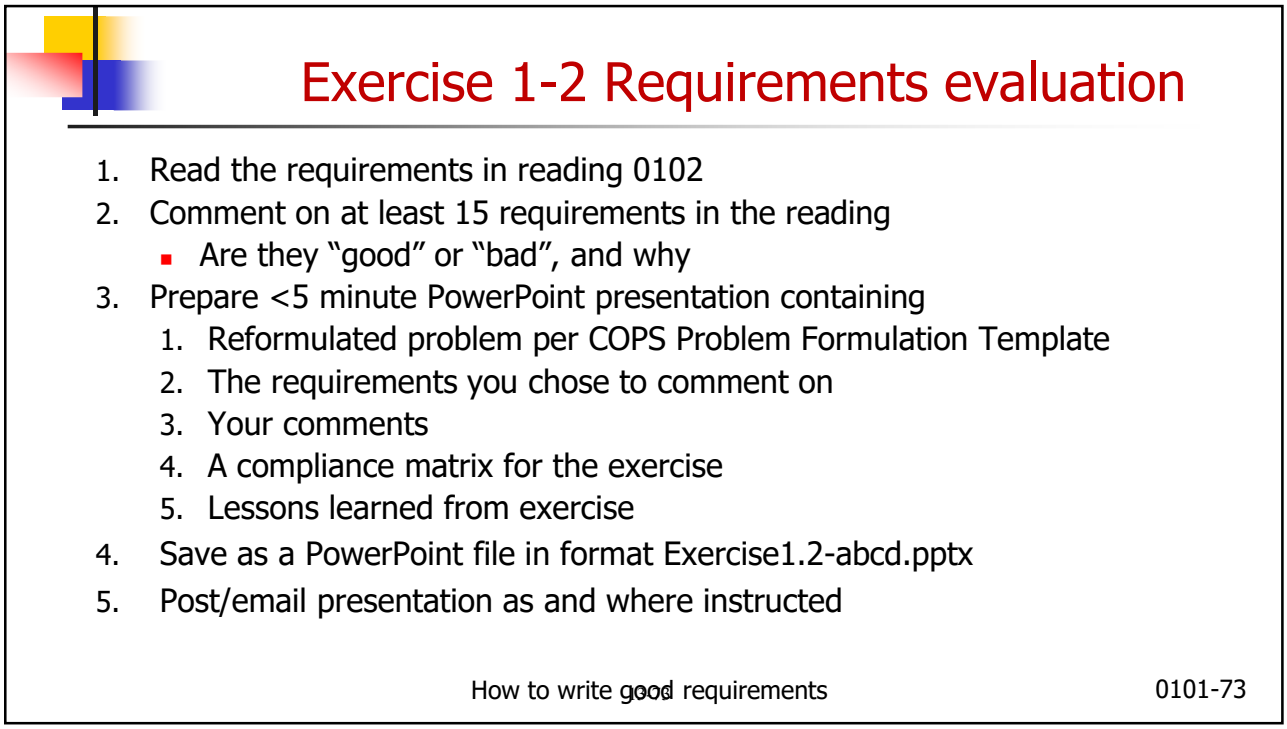
- Exercise presentations
- Background
- Glossaries and definitions
- Purpose of requirements
- Some of the problems caused by poorly-written requirements
- Well-written requirements
  - Attributes
  - Spelling and grammar
  - Structure of a requirement
  - Vocabulary of a requirement
- Generic and system specific requirements
- Examples of bad requirements and how to fix them
- **Exercises**

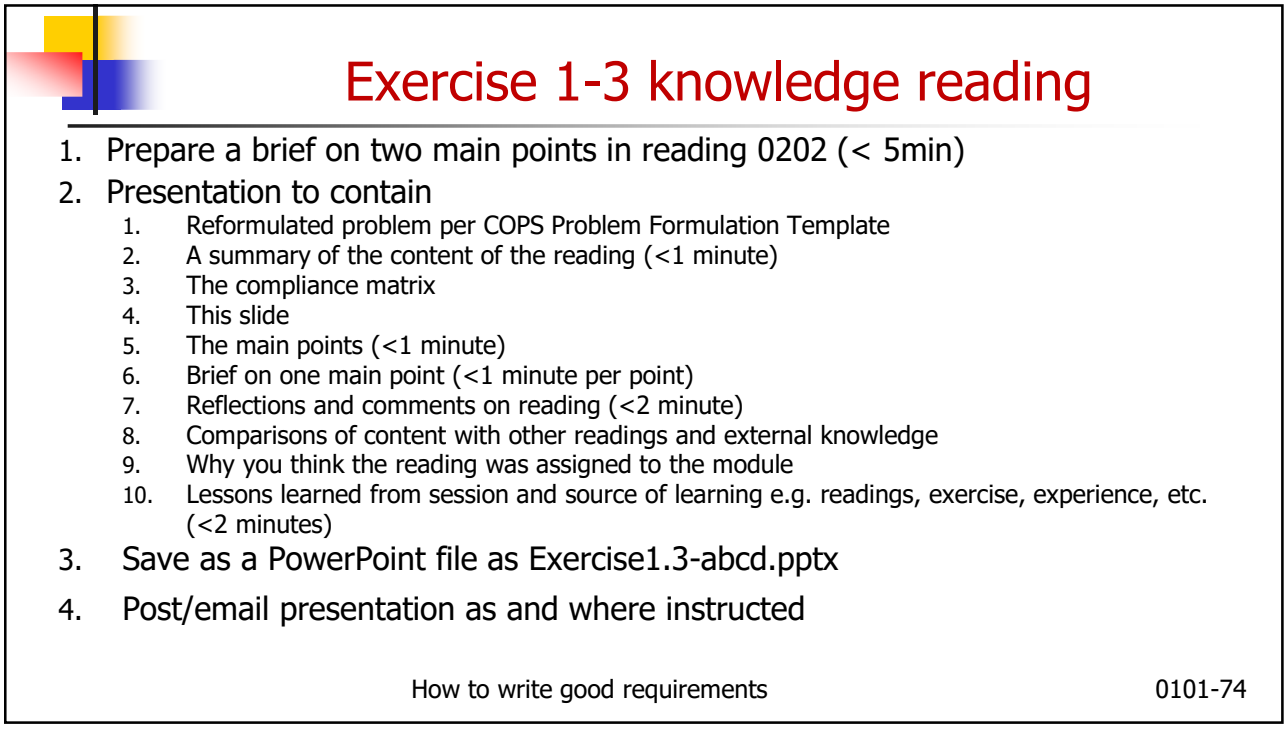
How to write good requirements 0101-71

Exercise 1-1 Compliance to a poorly written requirement

1. Imagine that you have just written the test procedure for verifying the system compliance to the poorly written requirement 509.1
2. Visualize the test procedure performed
3. Prepare a <5 minute presentation containing
   1. Reformulated problem per COPS Problem Formulation Template
   2. What you did to test each part of the requirement
   3. How you ensured the whole poorly-written requirement was tested
   4. A compliance matrix for the exercise
   5. Lessons learned from exercise
4. Save as a PowerPoint file in format Exercise1.1-abcd.pptx
   1. Note (abcd are your initials, where d is to be used when two participants have the same initials)
5. Post/email presentation as and where instructed

How to write good requirements 0101-72


Exercise 1-2 Requirements evaluation
1. Read the requirements in reading 0102
2. Comment on at least 15 requirements in the reading
Are they "good" or "bad", and why
3. Prepare <5 minute PowerPoint presentation containing
1. Reformulated problem per COPS Problem Formulation Template
2. The requirements you chose to comment on
3. Your comments
4. A compliance matrix for the exercise
5. Lessons learned from exercise
4. Save as a PowerPoint file in format Exercise1.2-abcd.pptx
5. Post/email presentation as and where instructed
How to write good requirements
0101-73


Exercise 1-3 knowledge reading
1. Prepare a brief on two main points in reading 0202 (< 5min)
2. Presentation to contain
1. Reformulated problem per COPS Problem Formulation Template
2. A summary of the content of the reading (<1 minute)
3. The compliance matrix
4. This slide
5. The main points (<1 minute)
6. Brief on one main point (<1 minute per point)
7. Reflections and comments on reading (<2 minute)
8. Comparisons of content with other readings and external knowledge
9. Why you think the reading was assigned to the module
10. Lessons learned from session and source of learning e.g. readings, exercise, experience, etc. (<2 minutes)
3. Save as a PowerPoint file as Exercise1.3-abcd.pptx
4. Post/email presentation as and where instructed
How to write good requirements
0101-74

Compliance matrix for meeting the objectives

#	Objective	Met
1	To provide the background to requirements	9-21
2	To provide definitions of the terminology used in the course	23-29
3	To explain the purposes of requirements	31
4	To explain some of the problems caused by poorly-written requirements	33-34
5	To explain the difference between requirements, well-written requirements and good requirements	25
6	To explain what constitutes a well-written requirement	36-55
7	To explain generic and system specific requirements and the benefits of the distinction	57-60
8	To show some examples of bad requirements and how to fix them	62-70
9	To explain why they are bad	62-70
10	To explain what needs to be done to convert them to well-written requirements	62-70
11	To provide the opportunity to obtain 5 levels of knowledge updated Blooms taxonomy	72-74

Objectives of Session 2

1. To share your BWMQ and exercises from previous session
2. To explain the importance of stakeholders
3. To explain where and how to locate potential stakeholders for the project
4. To explain the extended process to realize the solution system as a generic model for locating stakeholders
5. To explain the difference between
   - Customers and other stakeholders
   - Information and contractual communications between stakeholders and how to manage them
   - Stakeholder wants and needs
   - Direct and indirect stakeholders
   - Generic and specific stakeholders
6. To explain the degree of influence of each stakeholder on the requirements
7. To provide the opportunity to obtain 5 levels of knowledge in updated Blooms taxonomy

Any questions ?

1. Best
2. Worst
3. Missing
4. Questions



Email:
beyondsystemsthinking@yahoo.com
Subject: <class title> BWMQ Session #

How to write good requirements 0101-77