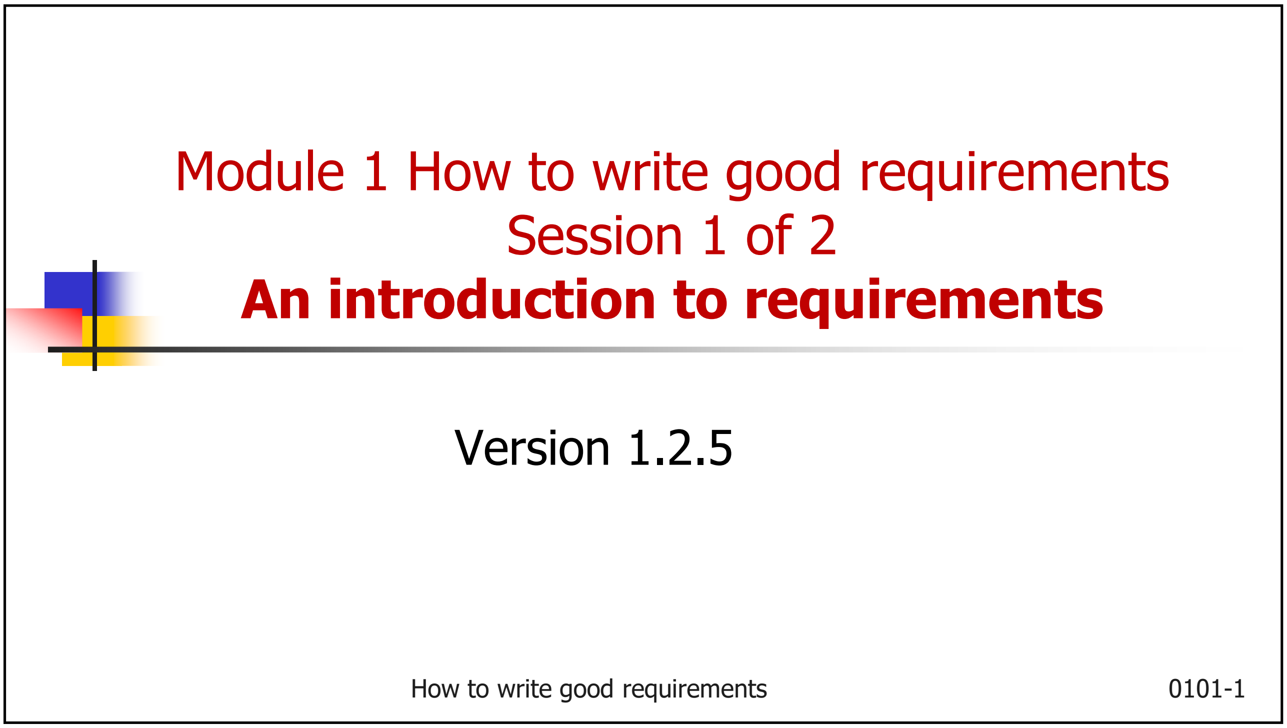
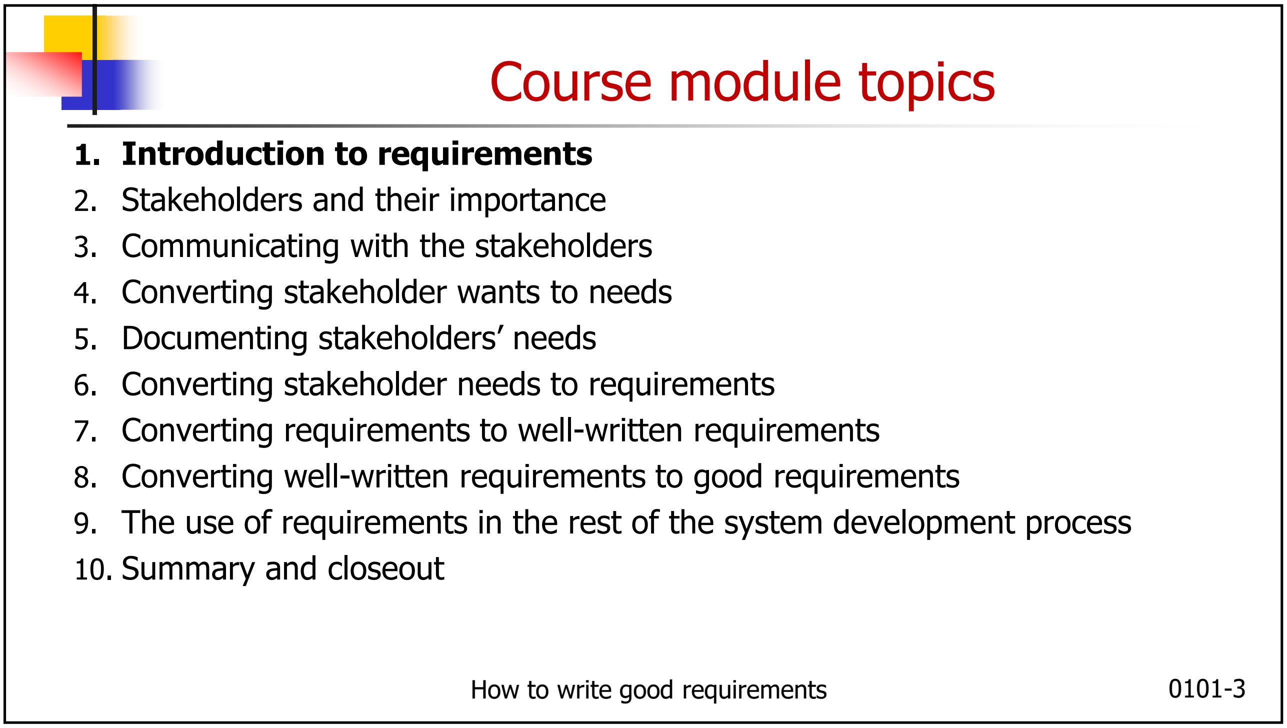


Module 1 How to write good requirements
Session 1 of 2
An introduction to requirements

Version 1.2.5

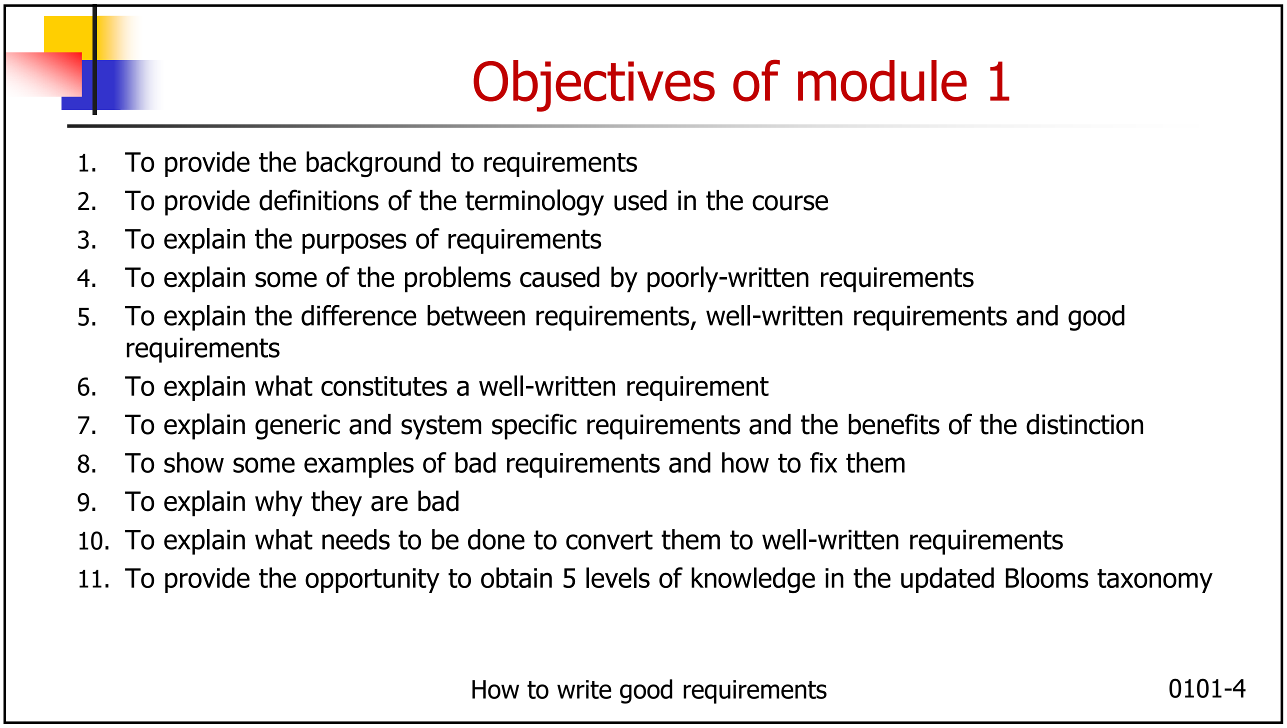
How to write good requirements 0101-1



Course module 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

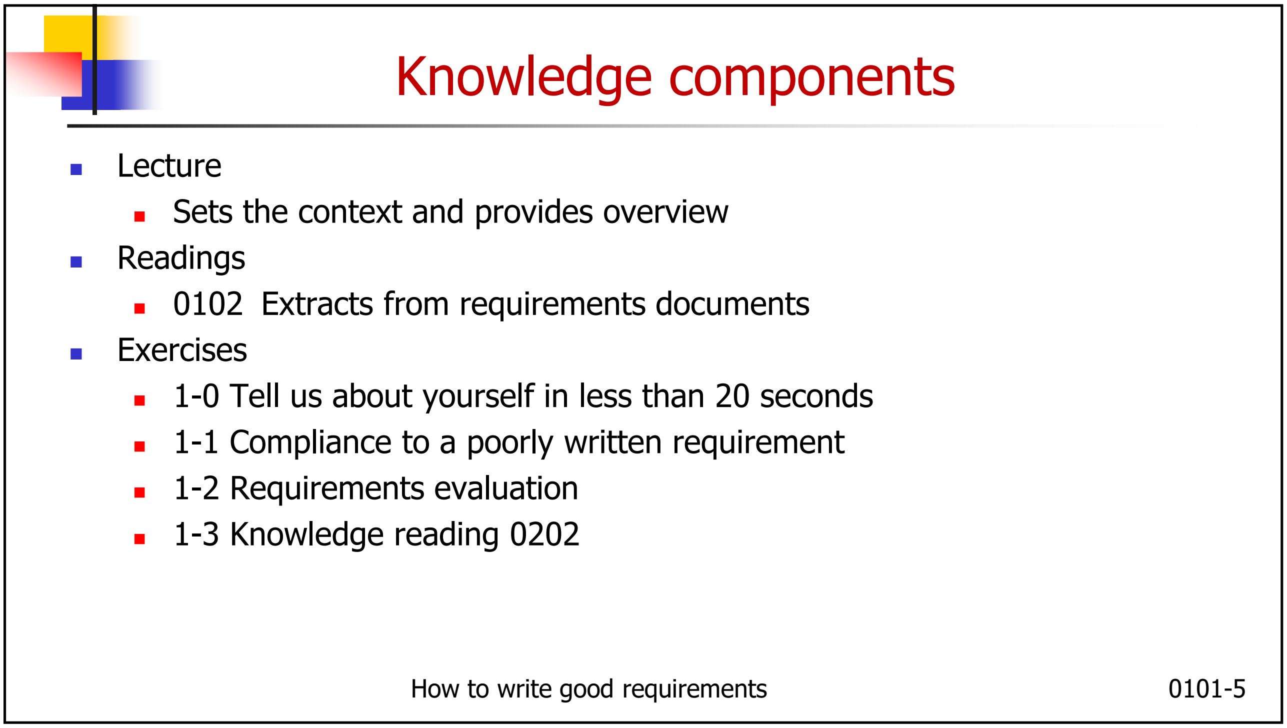
How to write good requirements 0101-3



Objectives of module 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

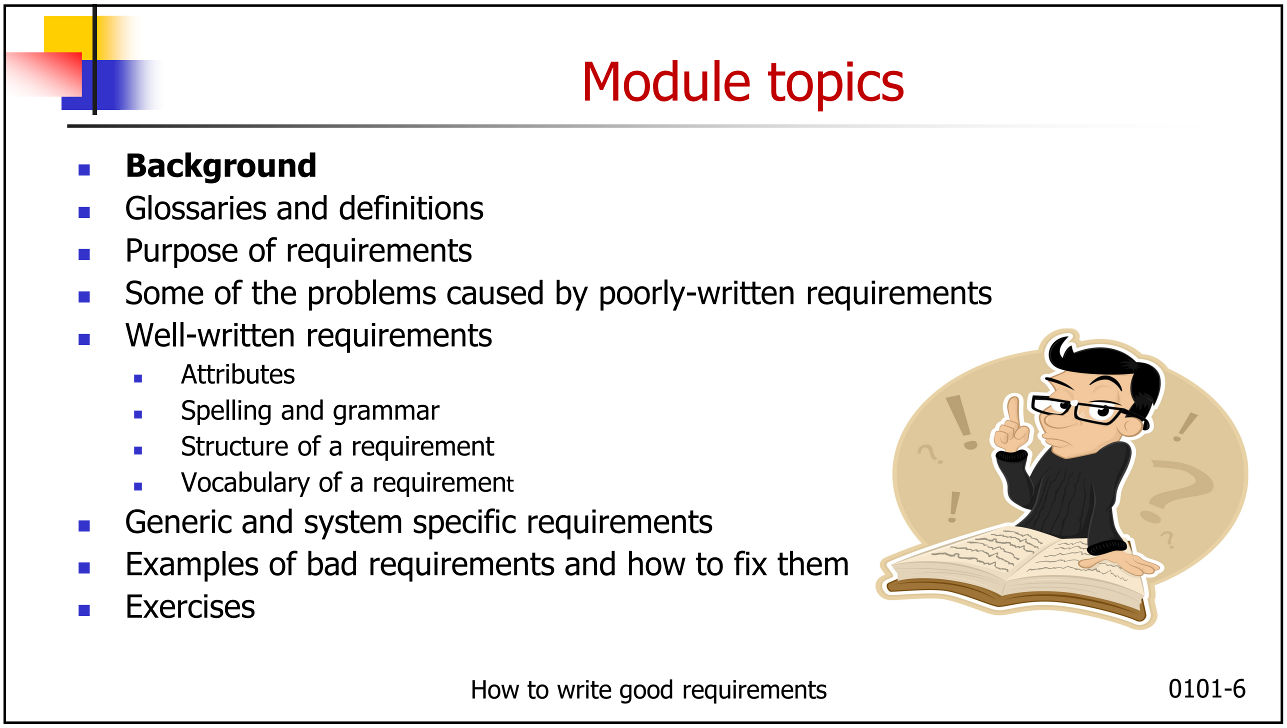
How to write good requirements 0101-4



Knowledge components

- Lecture
  - Sets the context and provides overview
- Readings
  - 0102 Extracts from requirements documents
- 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

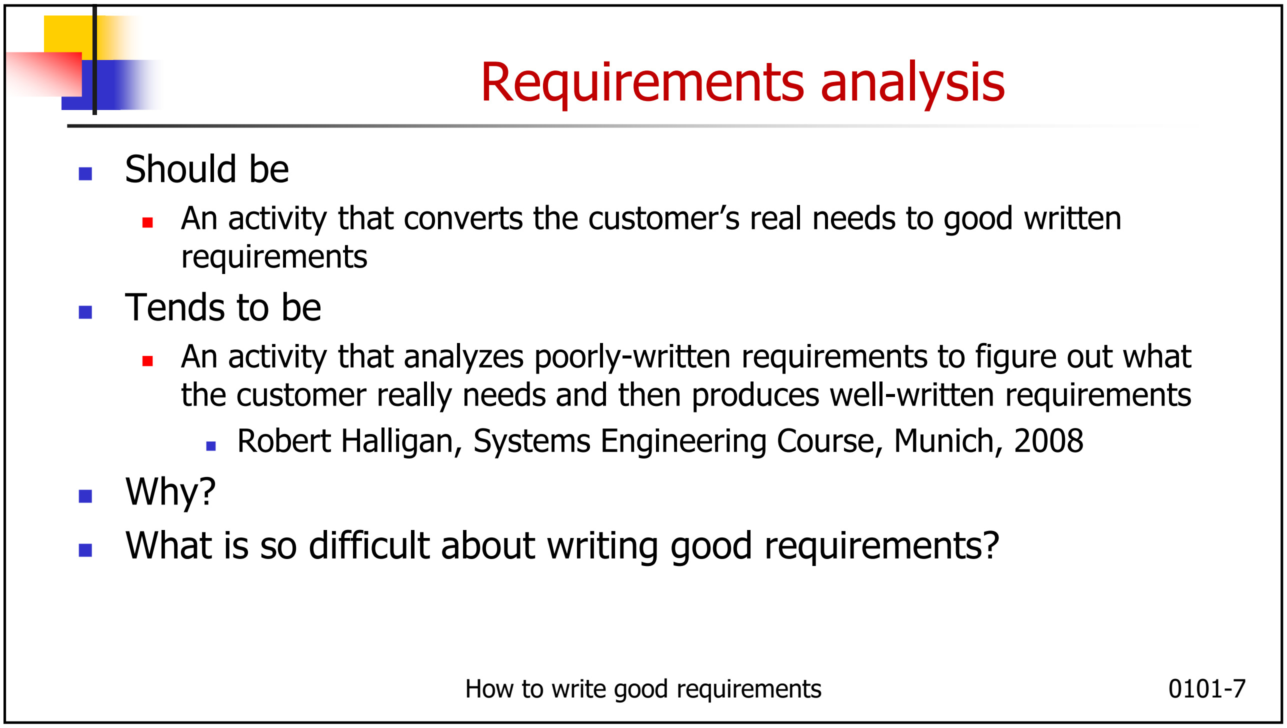
How to write good requirements 0101-5



Module topics

- **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

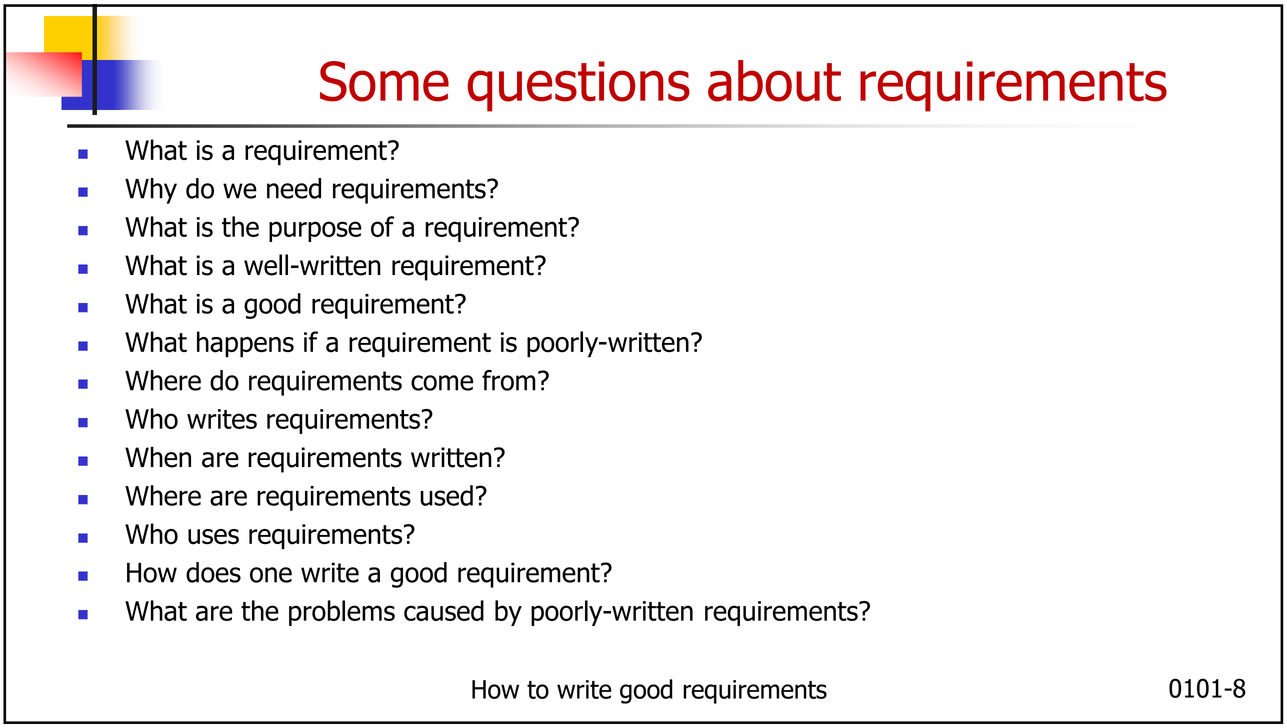
How to write good requirements 0101-6



Requirements analysis

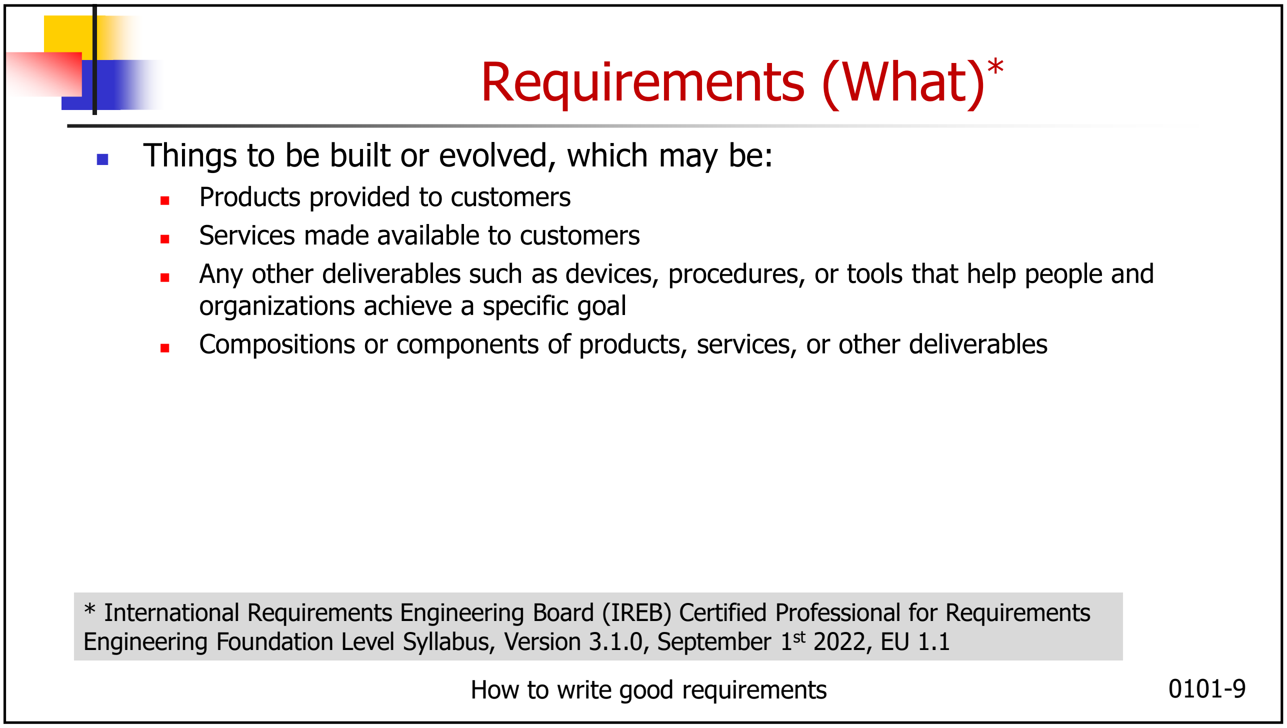
- Should be
  - An activity that converts the customer's real needs to good written requirements
- Tends to be
  - An activity that analyzes poorly-written requirements to figure out what the customer really needs and then produces well-written requirements
    - Robert Halligan, Systems Engineering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

How to write good requirements 0101-7



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?



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

Requirements (What)*

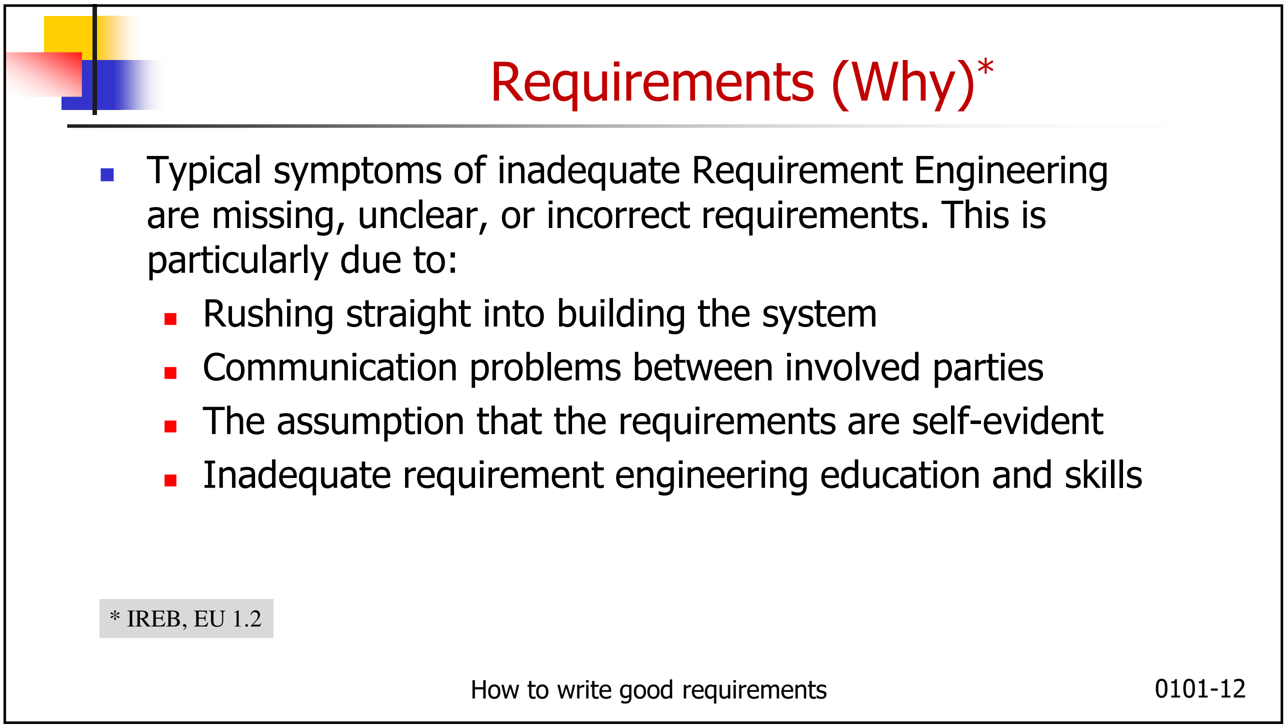
- Three types of requirements
  1. **Functional requirements -** concern a result or behavior that shall be provided by a function of a system. This includes requirements for data or the interaction of a system with its environment.
  2. **Quality requirements** - pertain to quality concerns that are 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

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

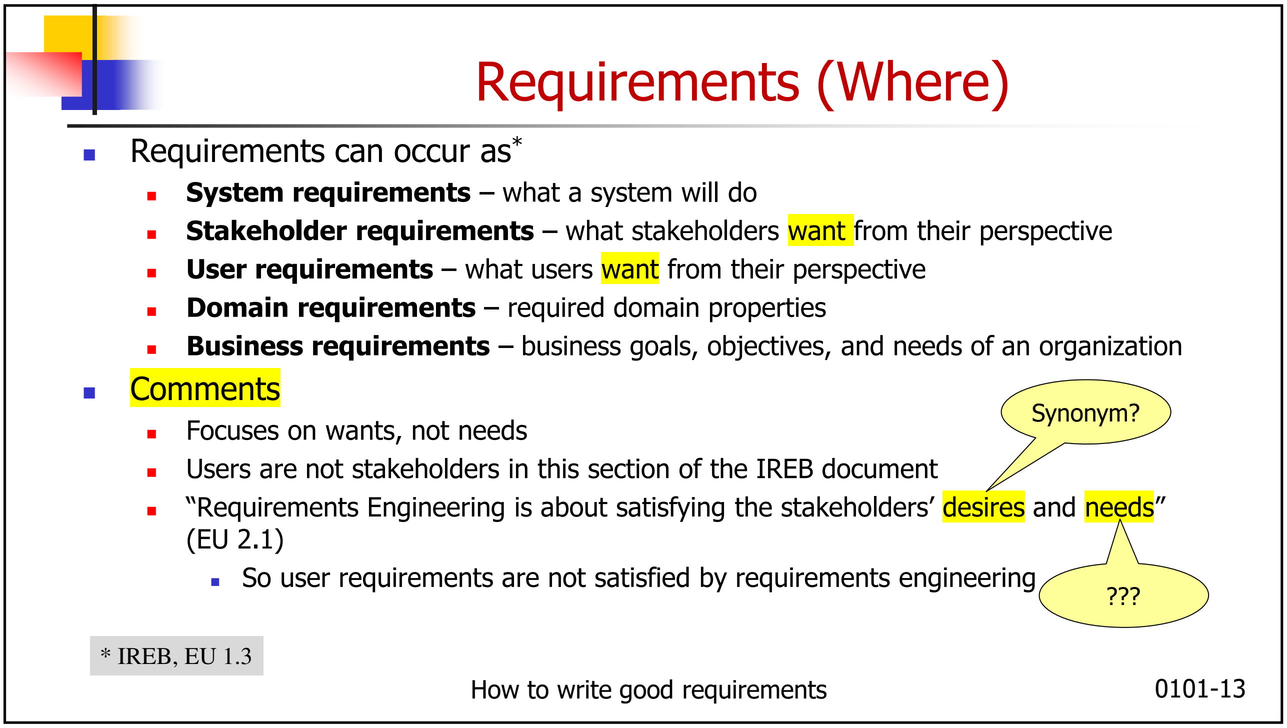


Requirements (Why)*

- 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 0101-12



Requirements (Where)

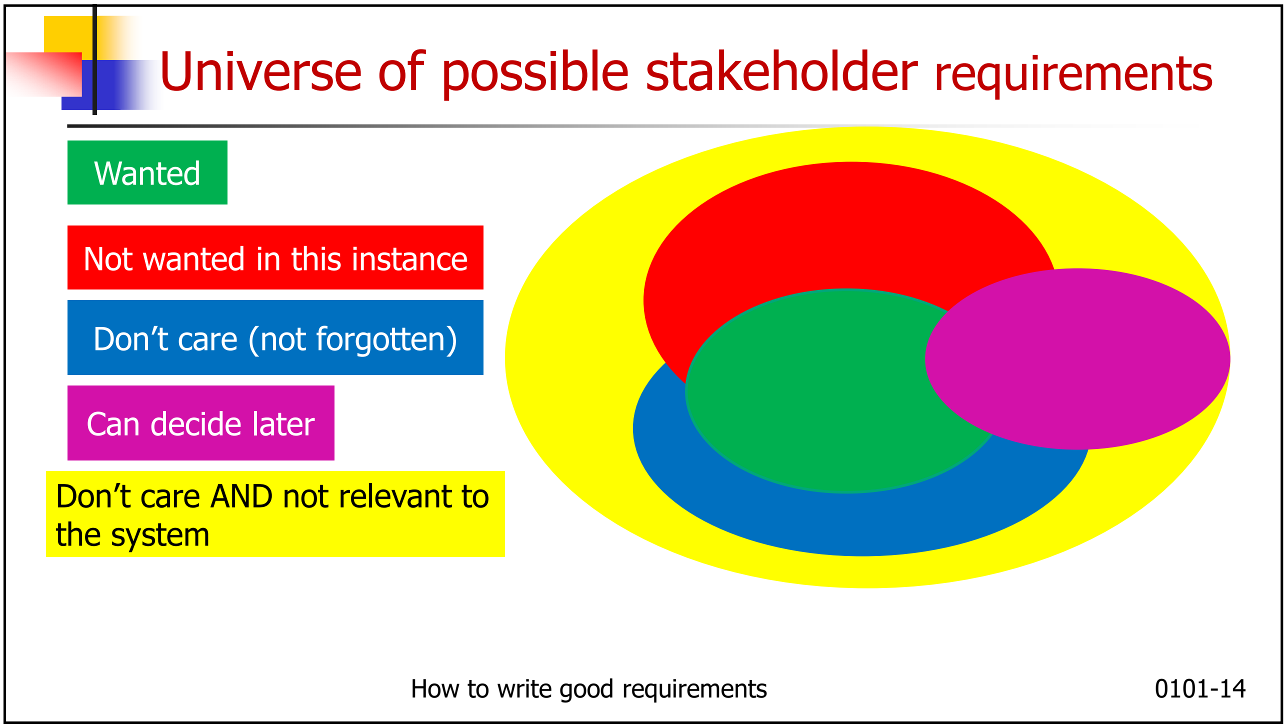
- Requirements can occur as*
  - **System requirements** – what a system will do
  - **Stakeholder requirements** – what stakeholders want from their perspective
  - **User requirements** – what users want from their perspective
  - **Domain requirements** – required domain properties
  - **Business 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

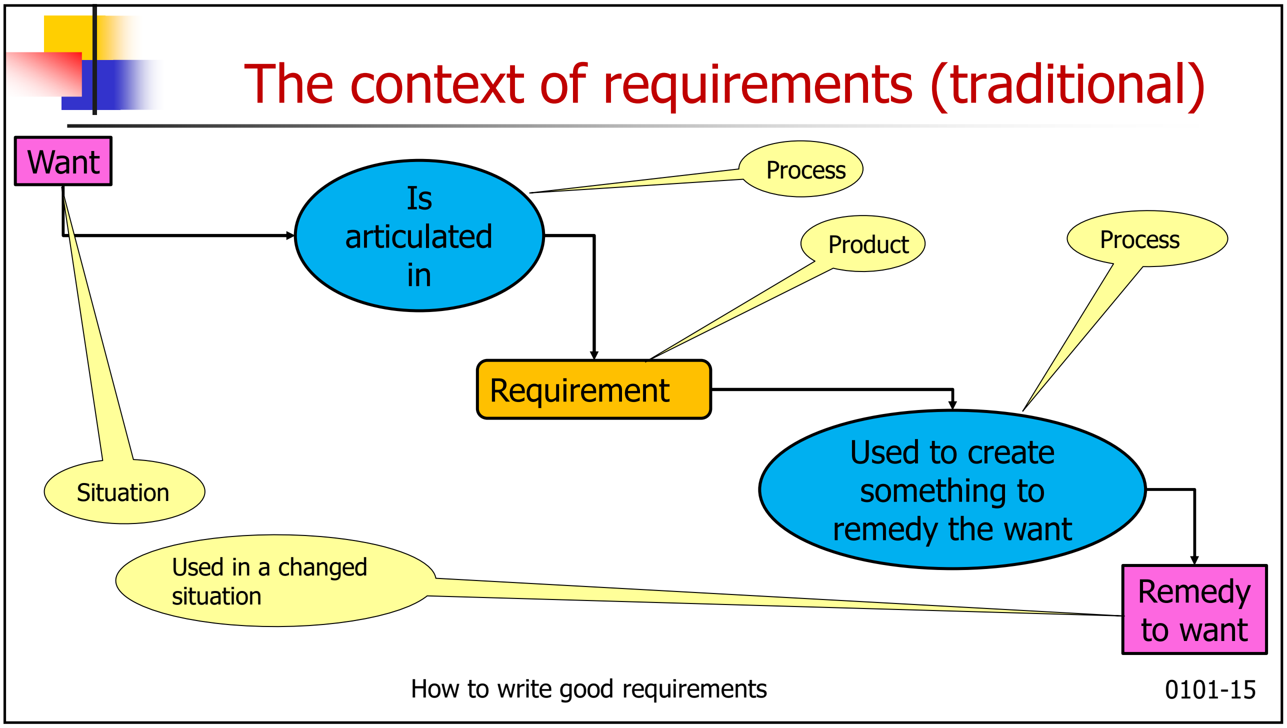
How to write good requirements 0101-13



Universe of possible stakeholder requirements

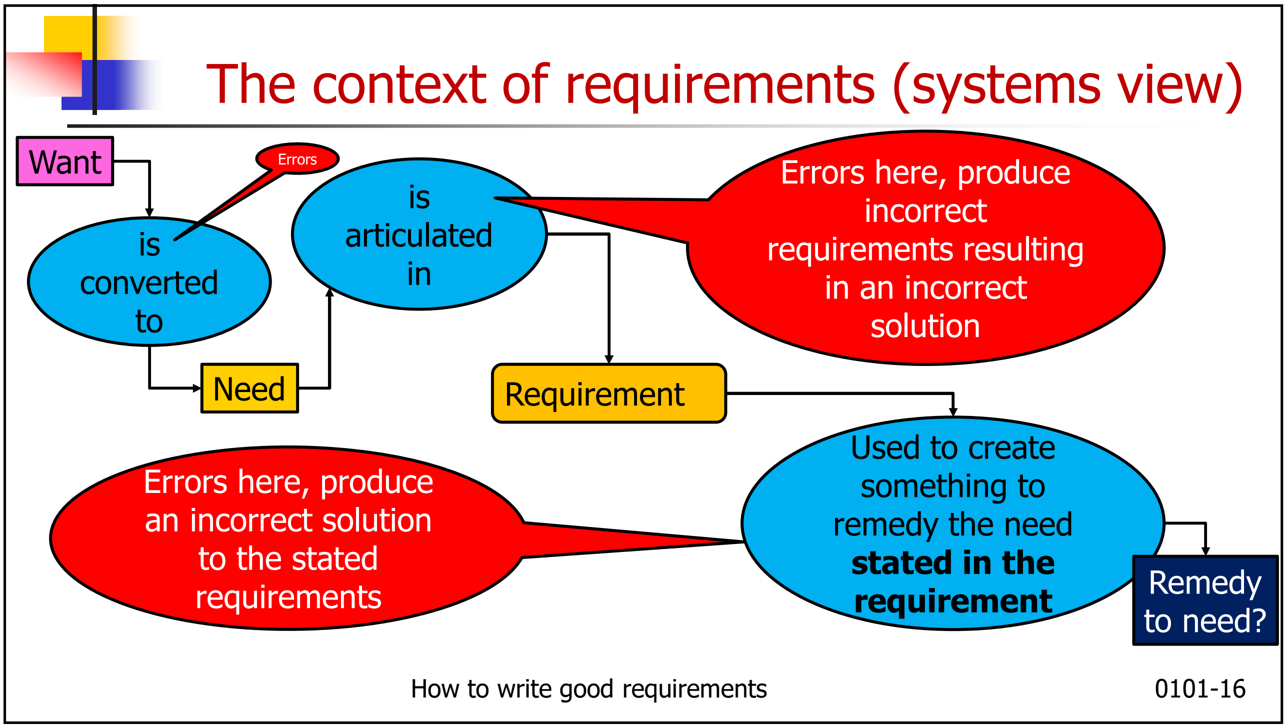
- 🟩 Wanted
- 🟥 Not wanted in this instance
- 🟦 Don’t care (not forgotten)
- 🟪 Can decide later
- 🟨 Don’t care AND not relevant to the system

How to write good requirements 0101-14



The context of requirements (traditional)

How to write good requirements 0101-15



The context of requirements (systems view)

Want

is converted to

Errors

Need

is articulated in

Errors here, produce incorrect requirements resulting in an incorrect solution

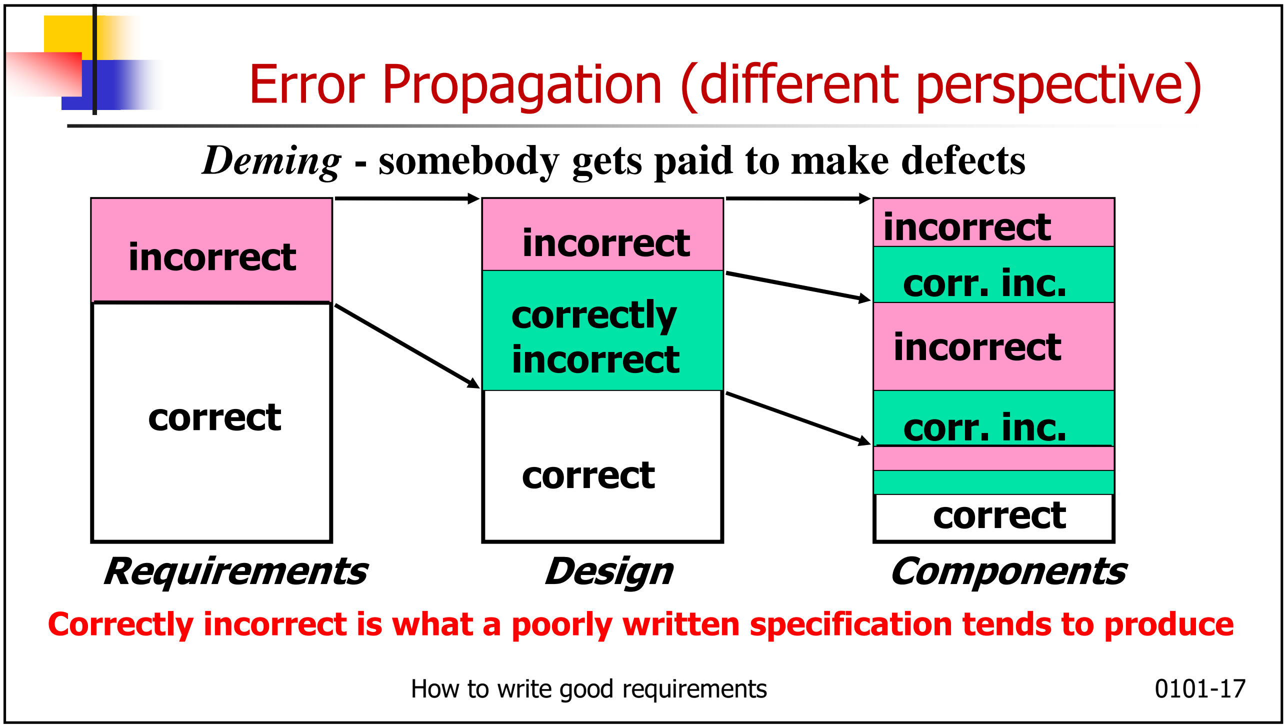
Requirement

Used to create something to remedy the need **stated in the requirement**

Remedy to need?

Errors here, produce an incorrect solution to the stated requirements

How to write good requirements 0101-16



Error Propagation (different perspective)

Deming **- somebody gets paid to make defects**

incorrect

correct

Requirements

incorrect

correctly incorrect

correct

Design

incorrect

corr. inc.

incorrect

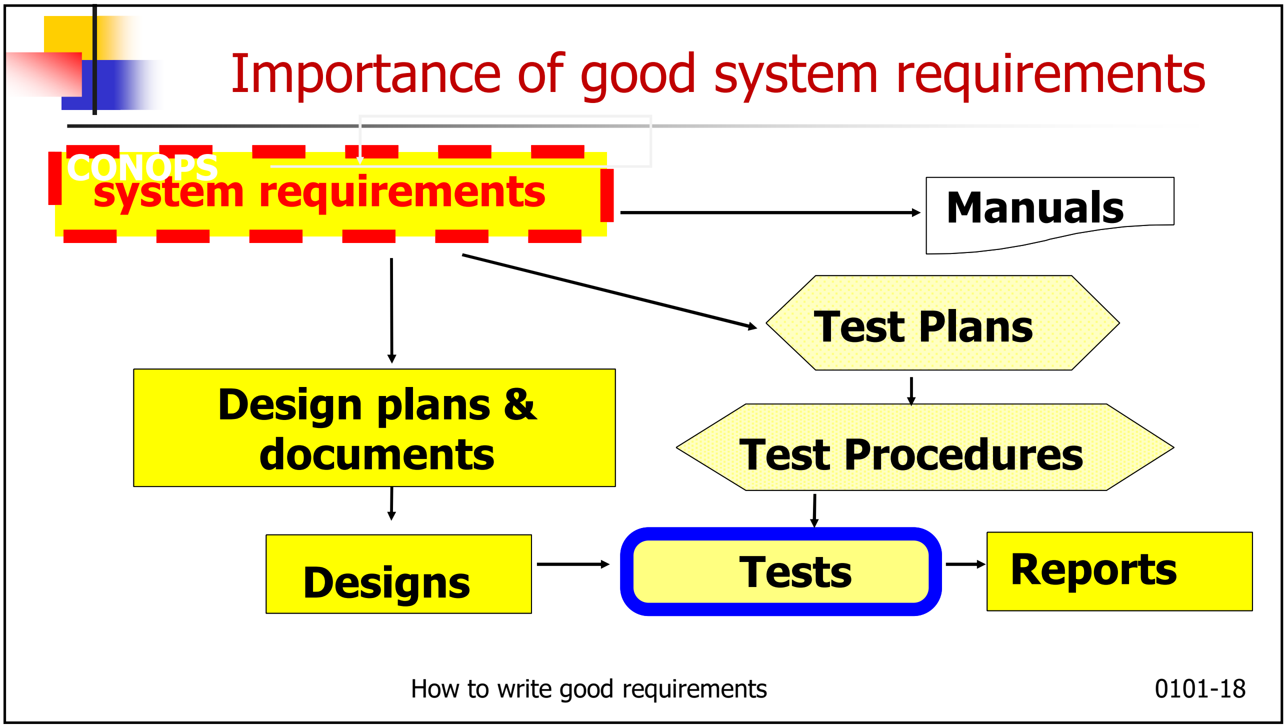
corr. inc.

correct

Components

Correctly incorrect is what a poorly written specification tends to produce

How to write good requirements 0101-17



Importance of good system requirements

CONOPS

system requirements

Manuals

Test Plans

Test Procedures

Design plans & documents

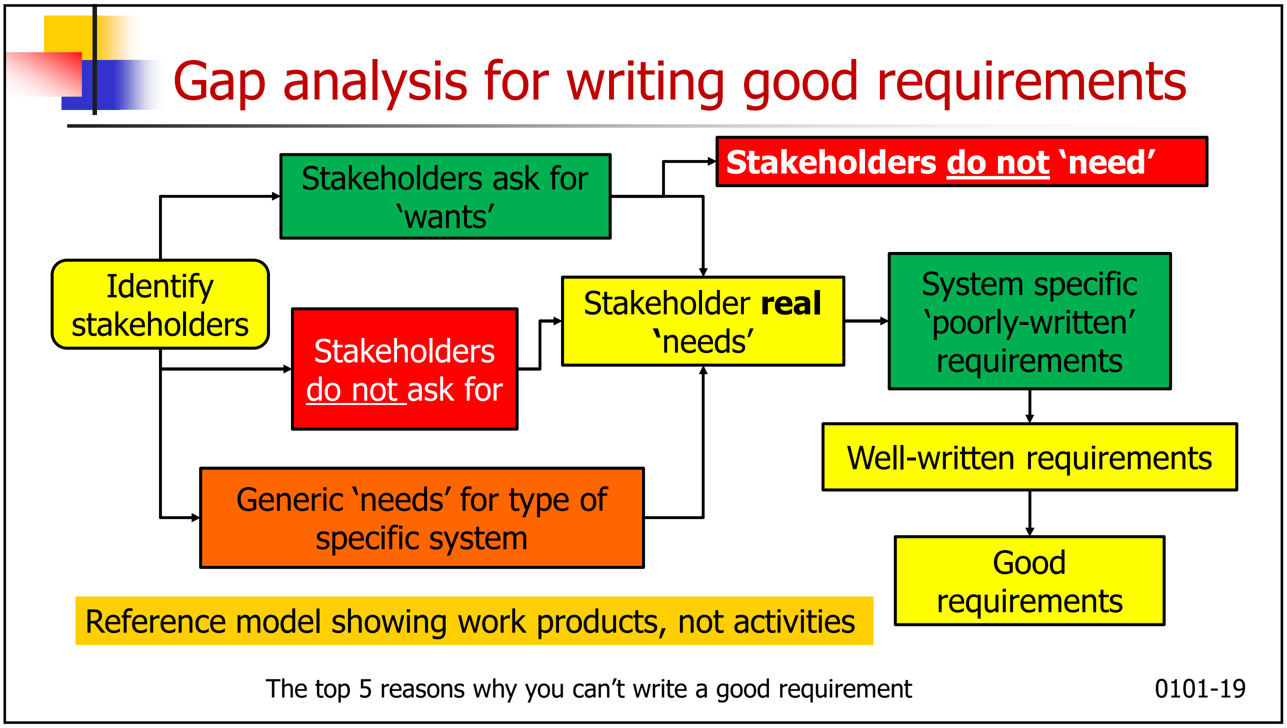
Designs

Tests

Reports

How to write good requirements

0101-18



Gap analysis for writing good requirements

Identify stakeholders

Stakeholders ask for 'wants'

Stakeholders do not 'need'

Stakeholders do not ask for

Stakeholder **real** 'needs'

Generic 'needs' for type of specific system

System specific 'poorly-written' requirements

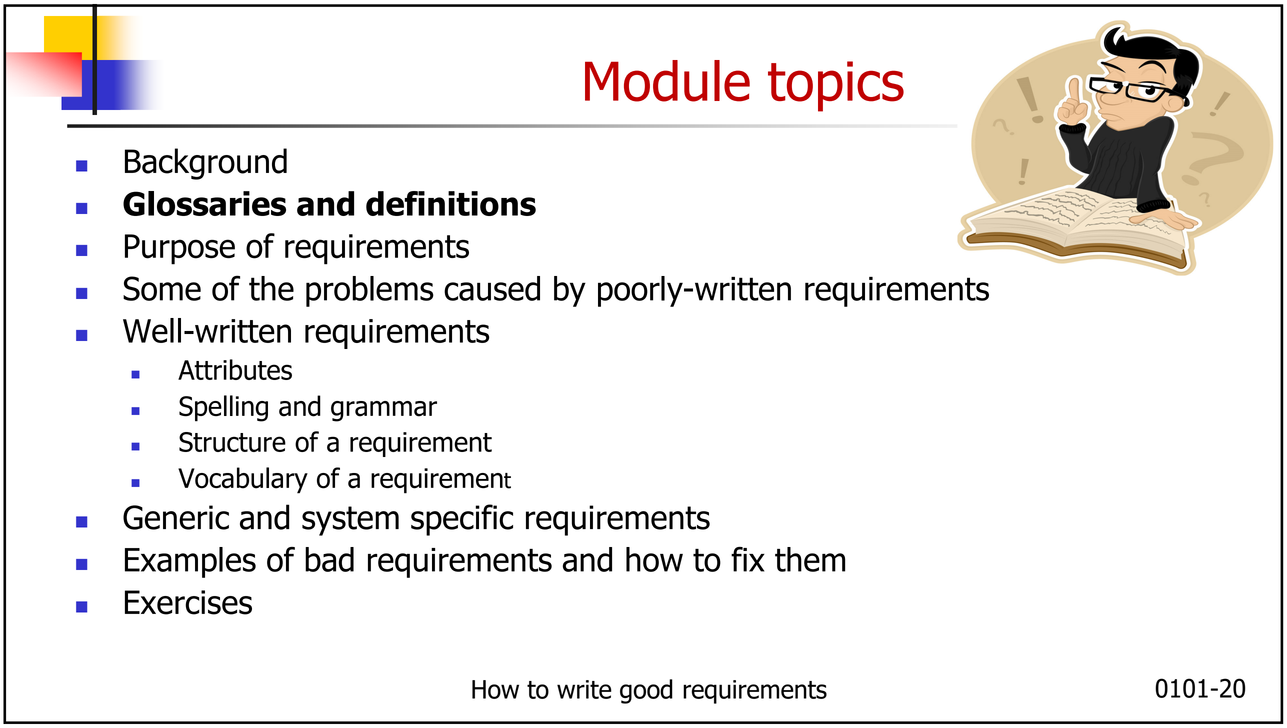
Well-written requirements

Good requirements

Reference model showing work products, not activities

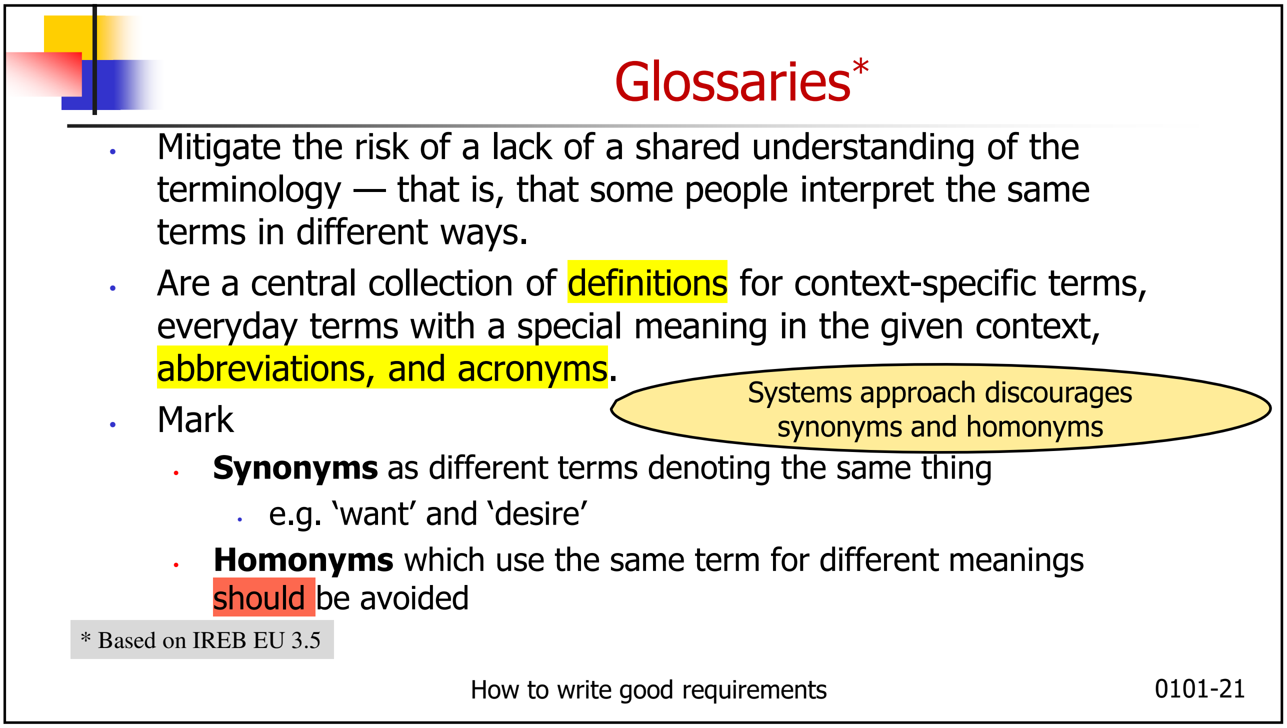
The top 5 reasons why you can't write a good requirement

0101-19



Module topics

- 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

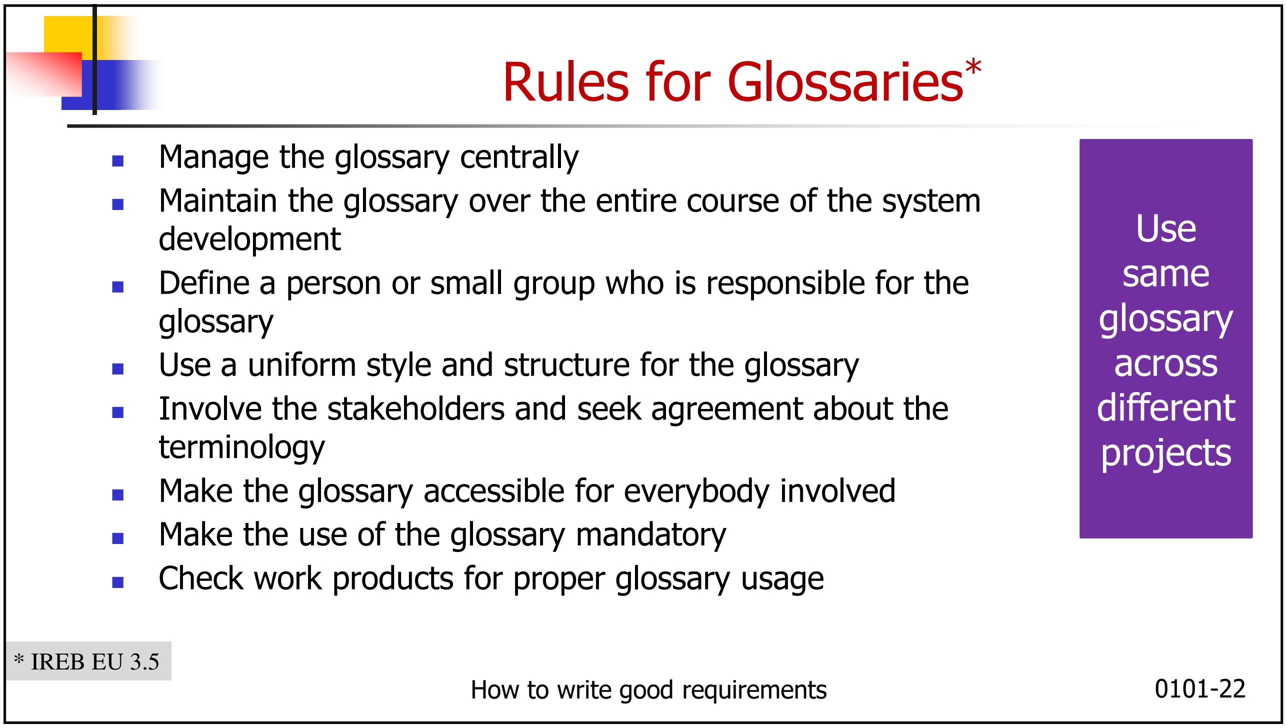


Glossaries*

- 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



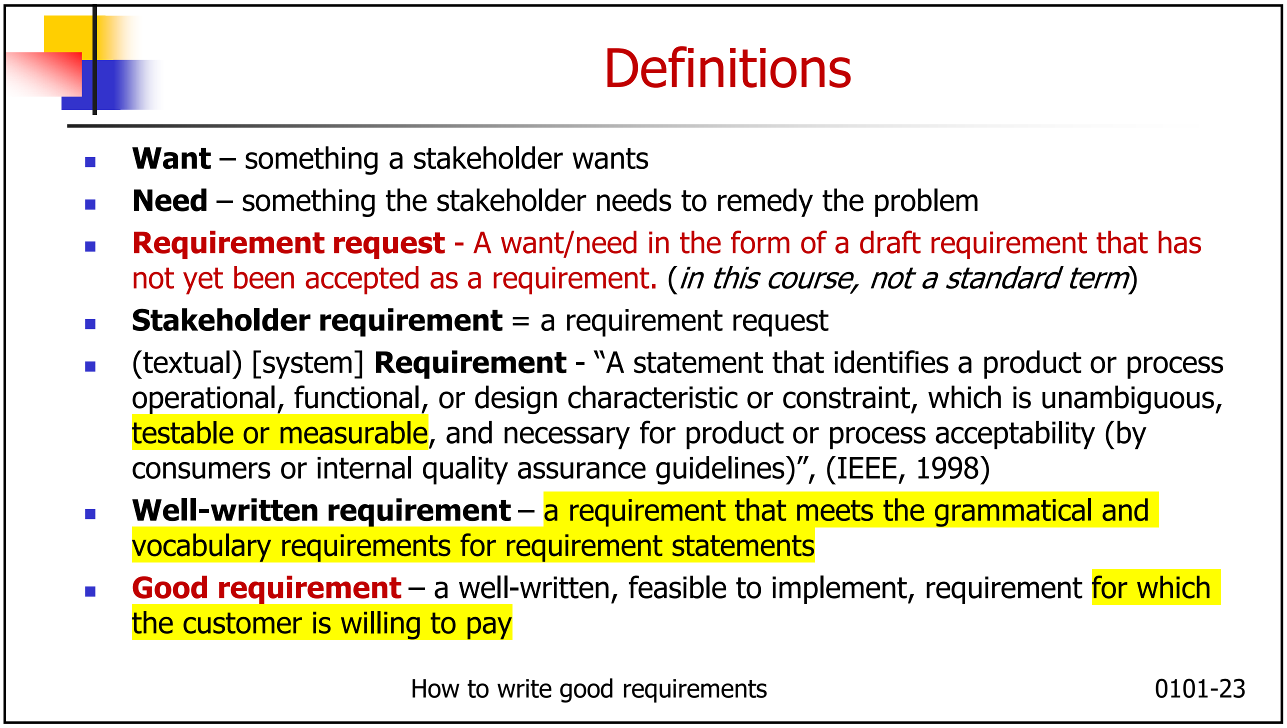
Rules for Glossaries*

- 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

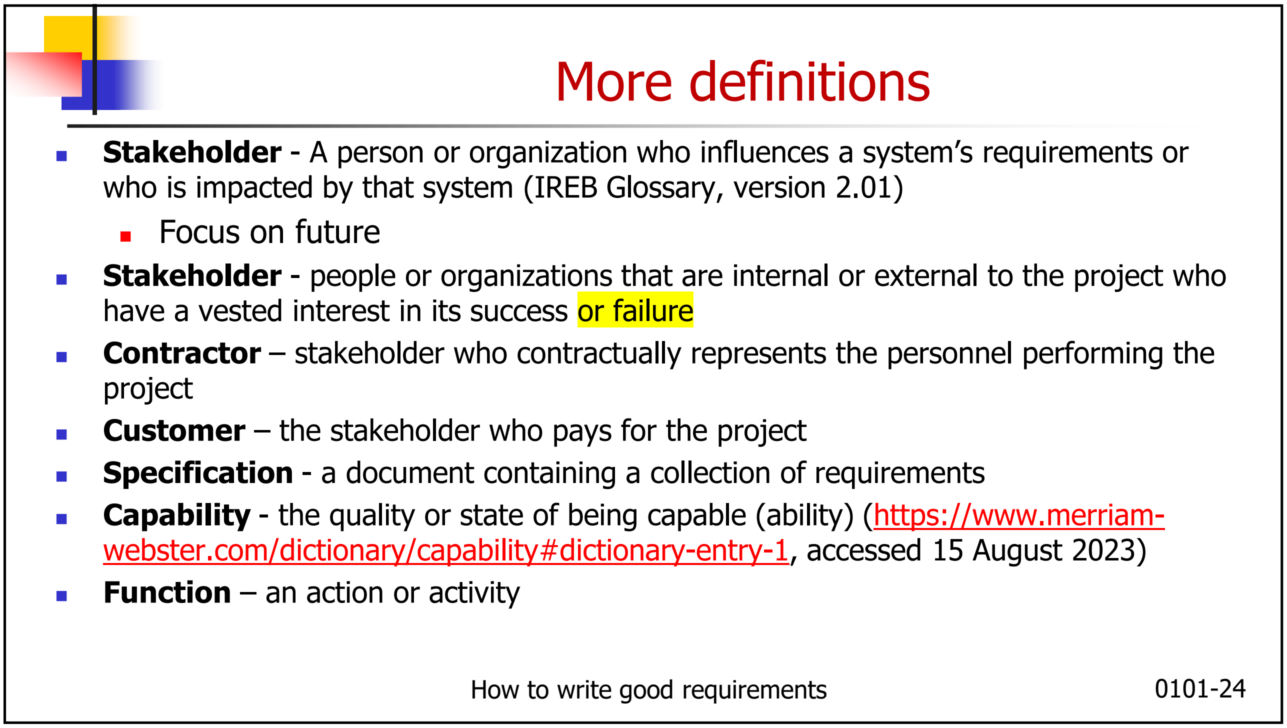
How to write good requirements 0101-22



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

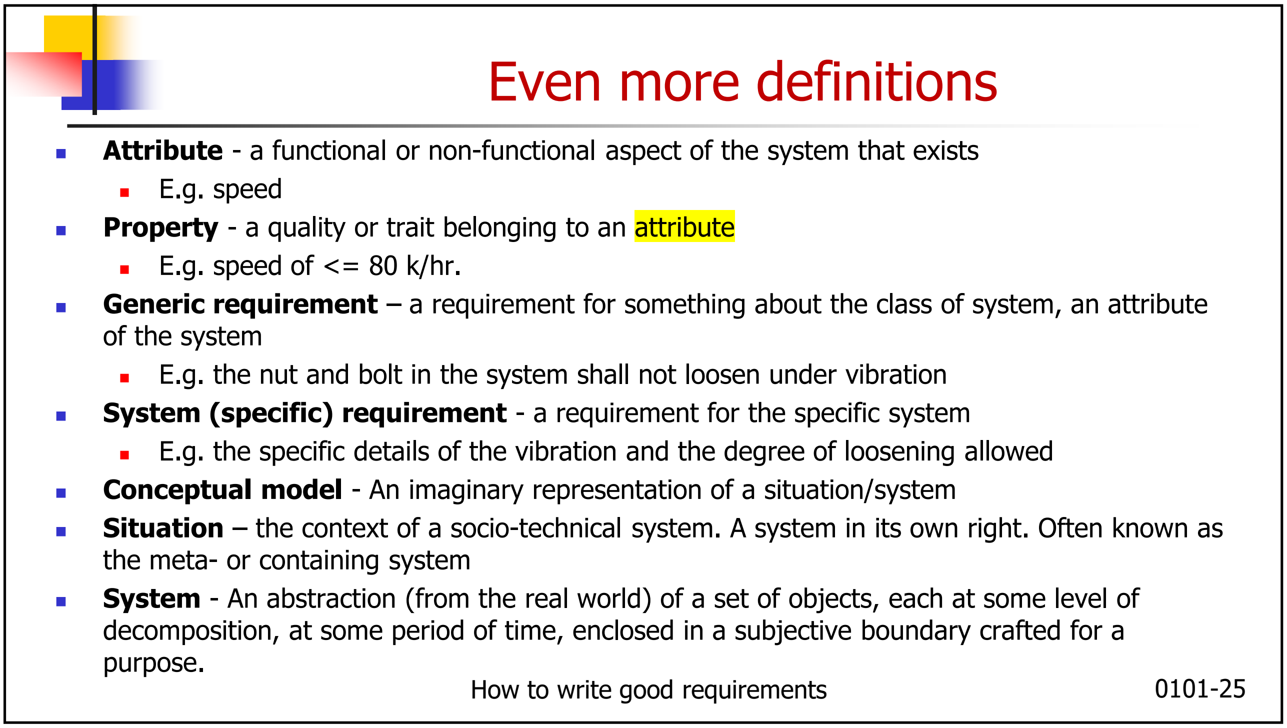
How to write good requirements 0101-23



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

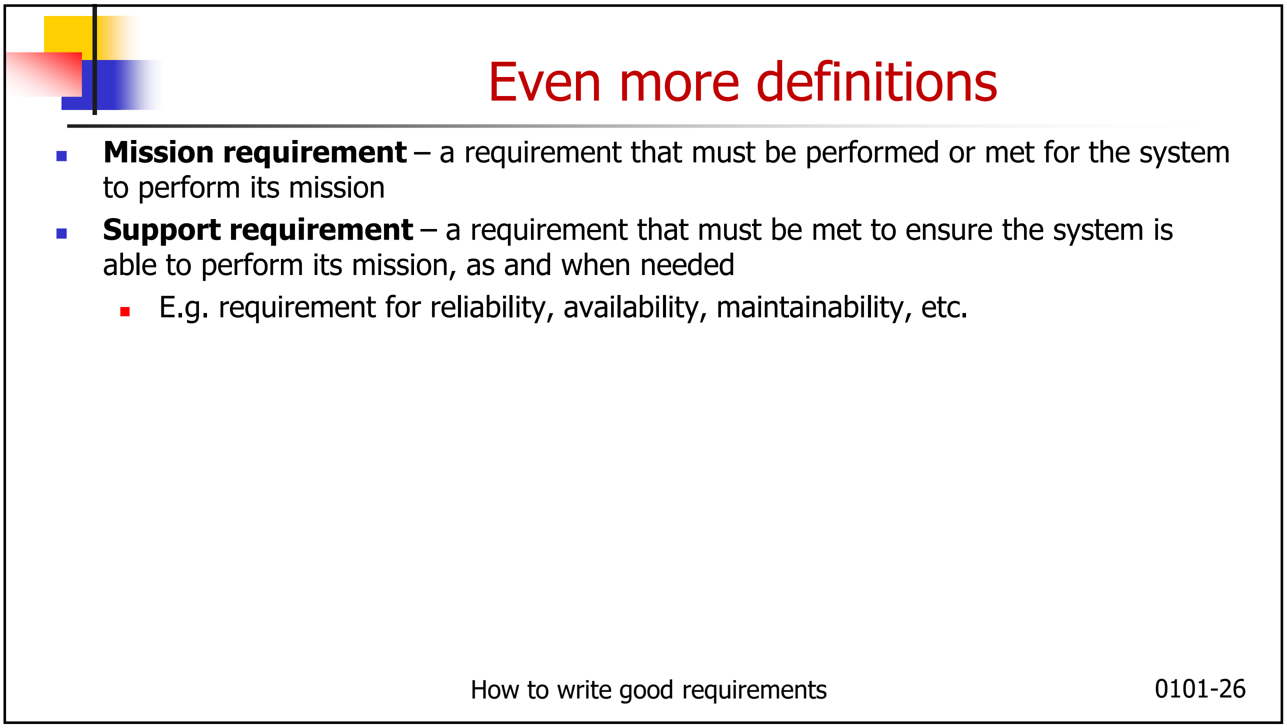
How to write good requirements 0101-24



Even more definitions

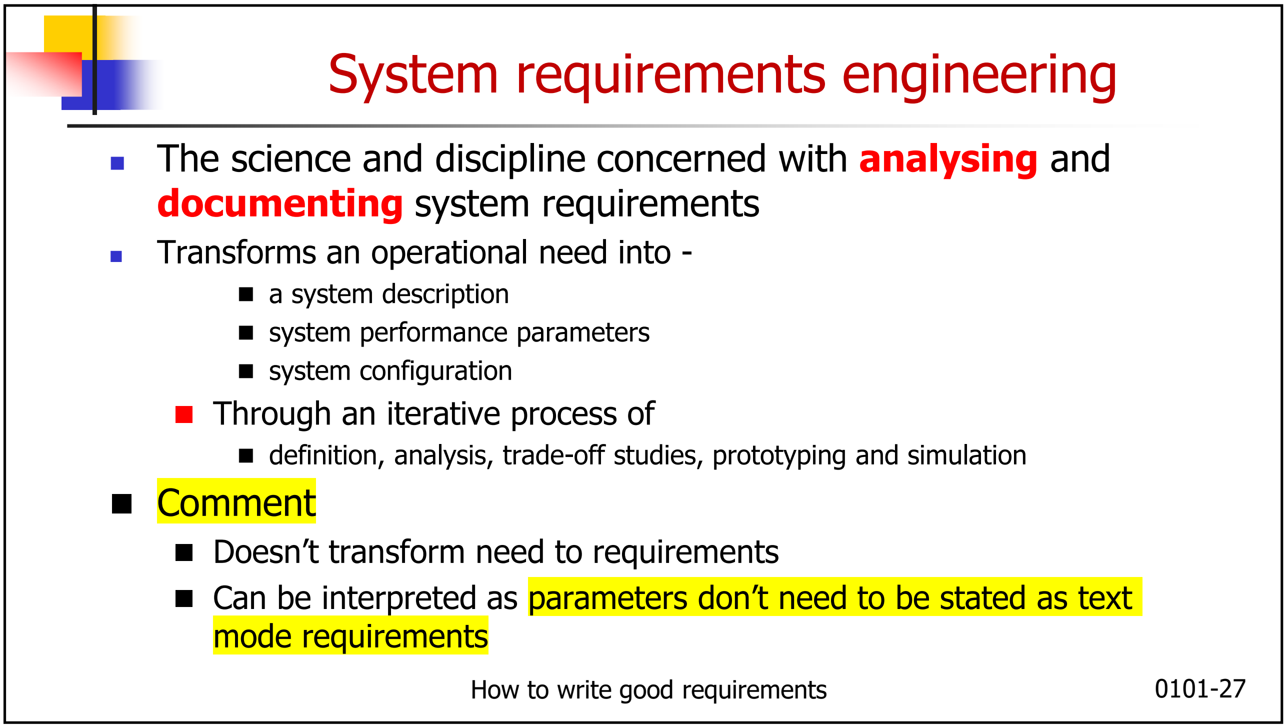
- **Attribute** - a functional or non-functional aspect of the system that exists
  - E.g. speed
- **Property** - a quality or trait belonging to an attribute
  - E.g. speed of <= 80 k/hr.
- **Generic requirement** – a requirement for something about the class of system, an attribute of the system
  - 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.

How to write good requirements 0101-25



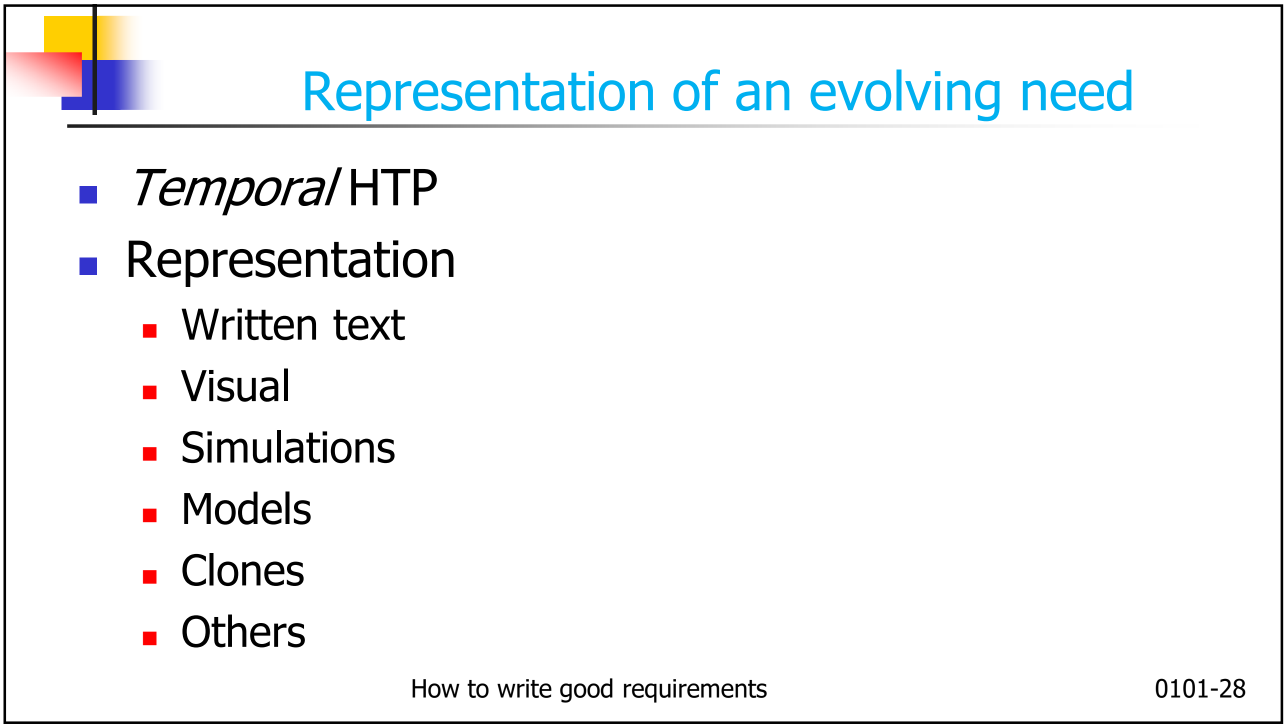
Even more definitions

- **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

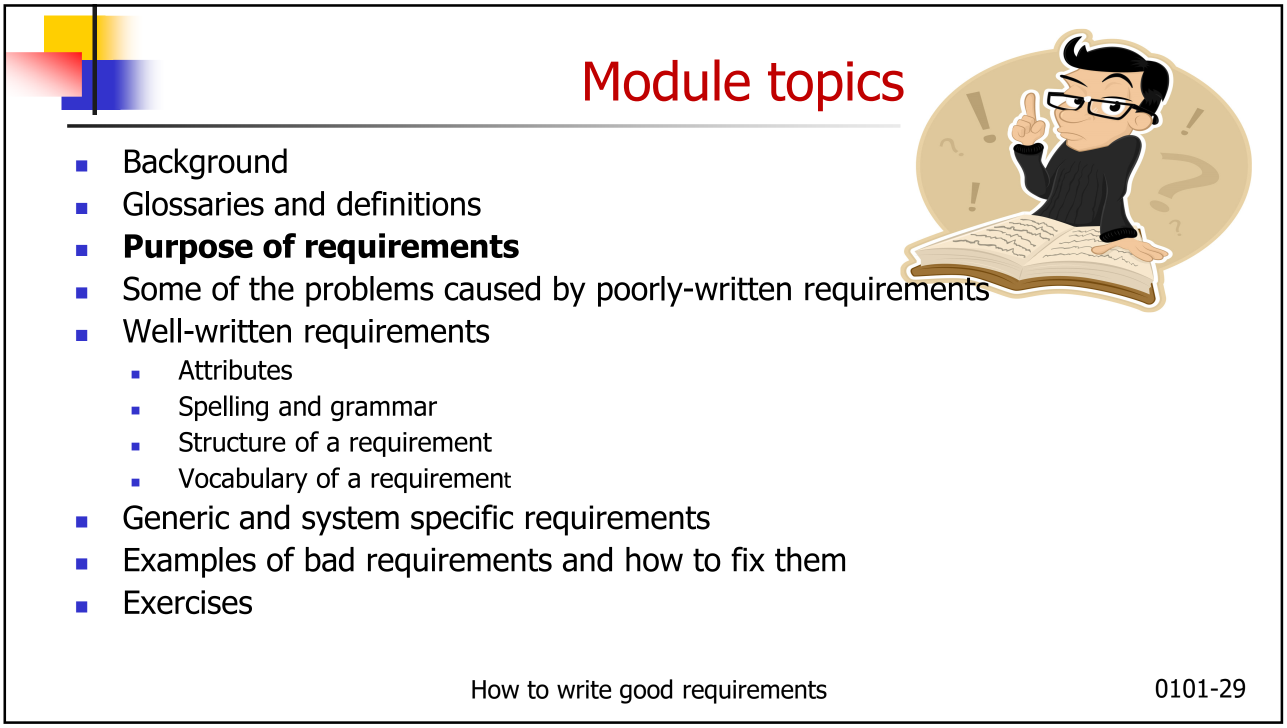
- The science and discipline concerned with **analysing** and **documenting** system requirements
- Transforms an operational need into -
    - a system description
    - system performance parameters
    - system configuration
  - Through an iterative process of
    - definition, analysis, trade-off 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



Representation of an evolving need

- *Temporal* HTP
- Representation
  - Written text
  - Visual
  - Simulations
  - Models
  - Clones
  - Others

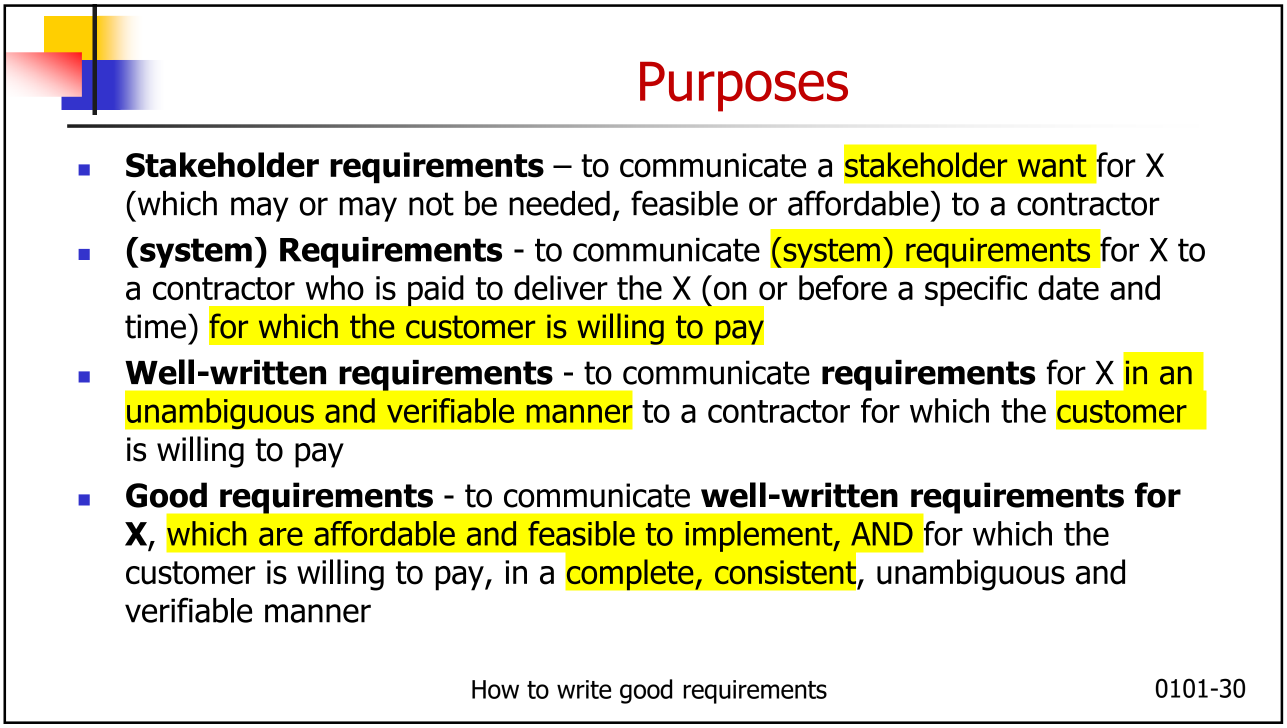
How to write good requirements 0101-28



Module topics

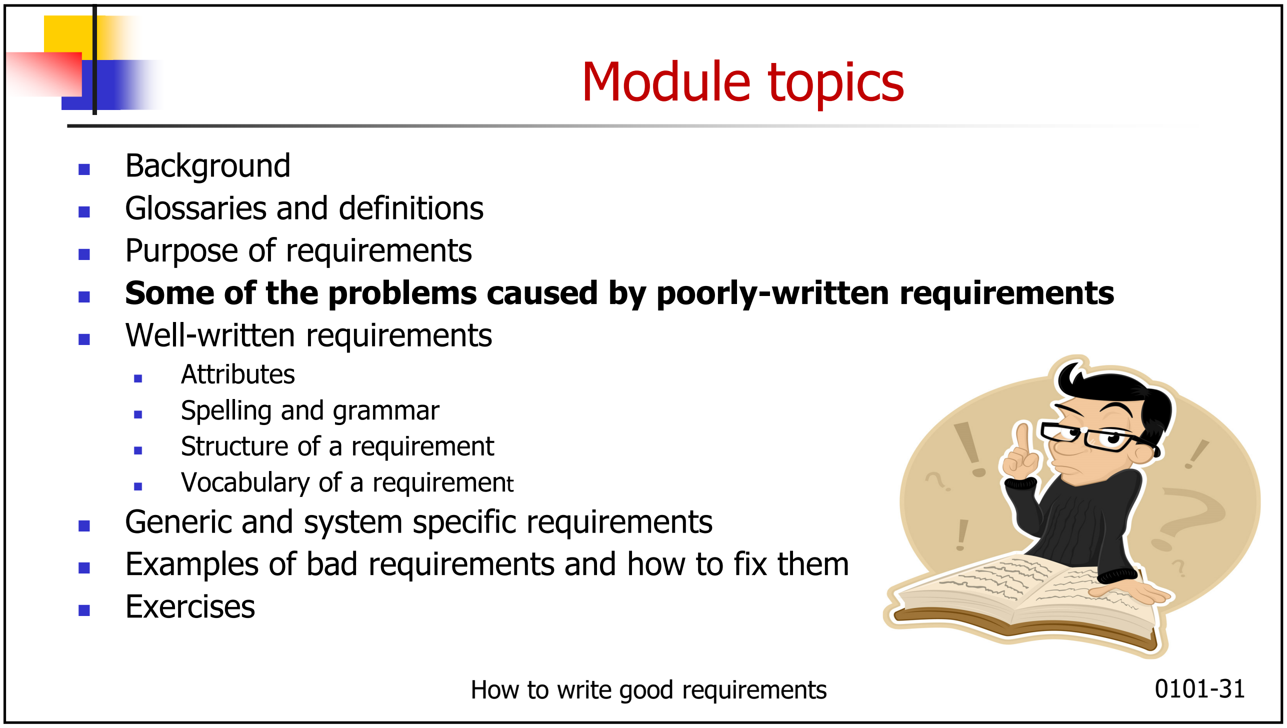
- Background
- Glossaries and definitions
- **Purpose of requirements**
- Some of the problems caused by poorly-written requirements
- Well-written requirements
  - Attributes
  - Spelling and grammar
  - Structure of a requirement
  - Vocabulary of a requirement
- Generic and system specific requirements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

How to write good requirements 0101-29



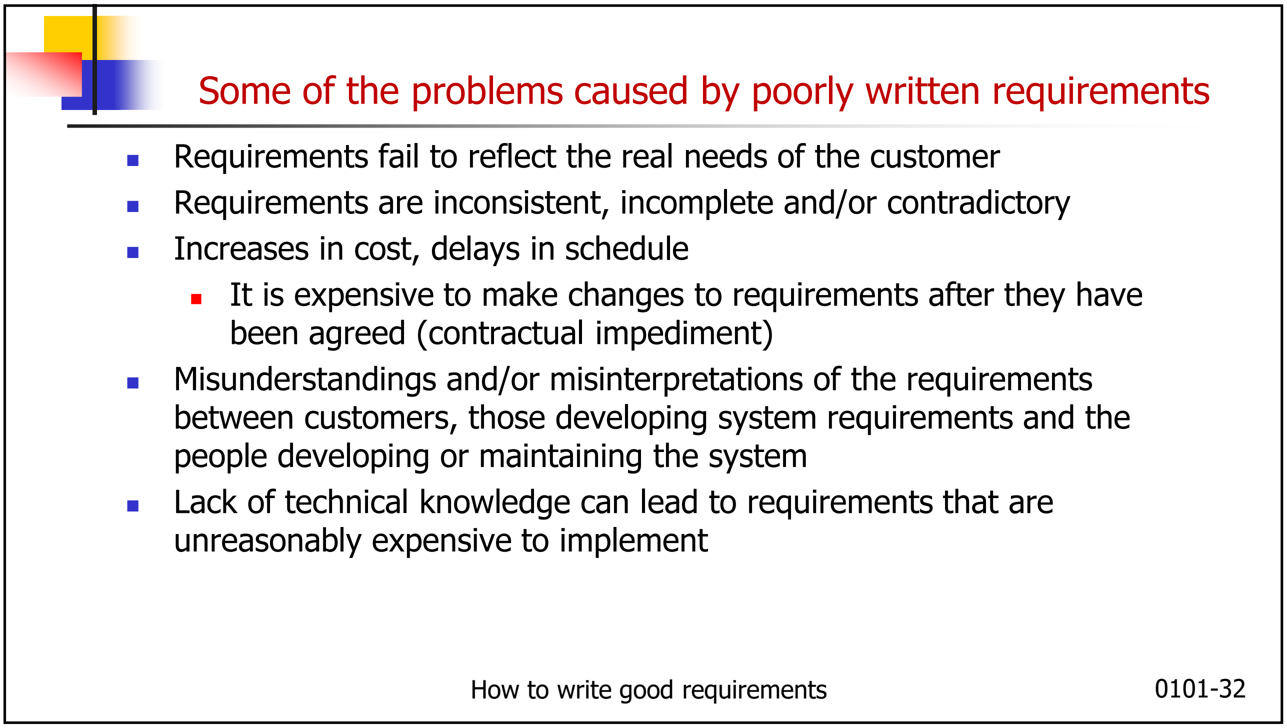
Purposes

- **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



Module topics

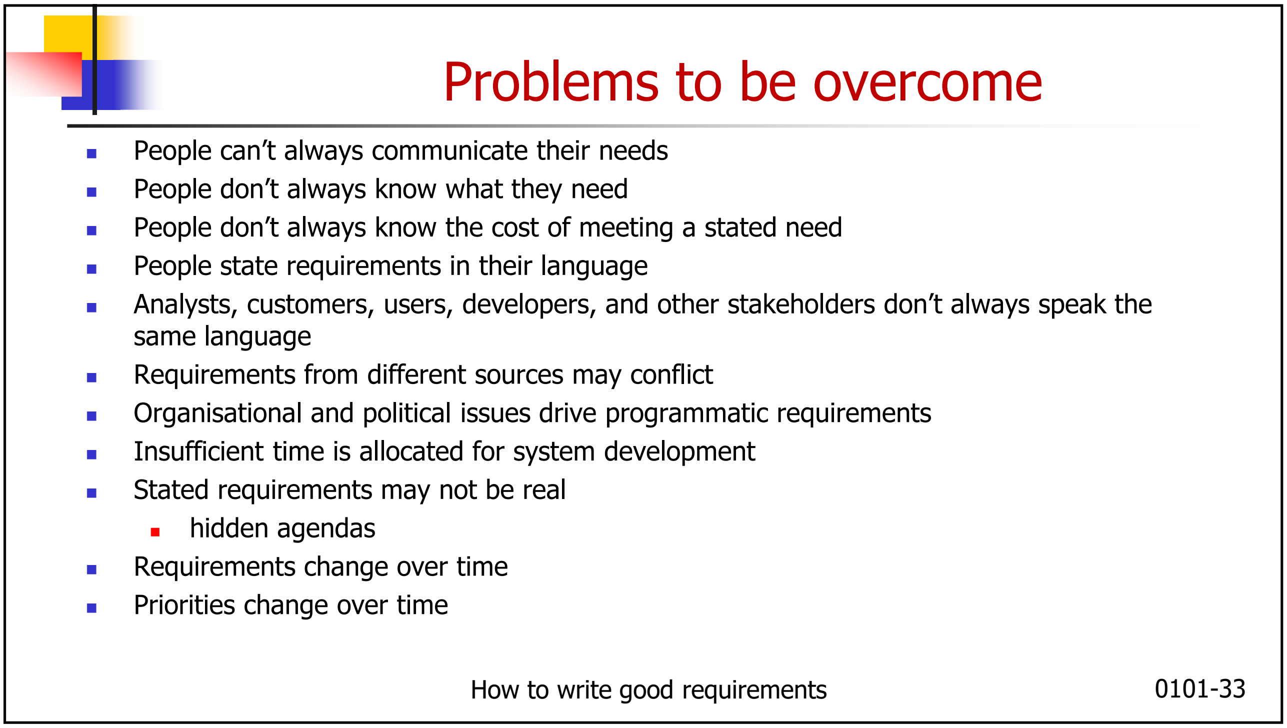
- Background
- Glossaries and definitions
- Purpose of requirements
- **Some of the problems caused by poorly-written requirements**
- Well-written requirements
  - Attributes
  - Spelling and grammar
  - Structure of a requirement
  - Vocabulary of a requirement
- Generic and system specific requirements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



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

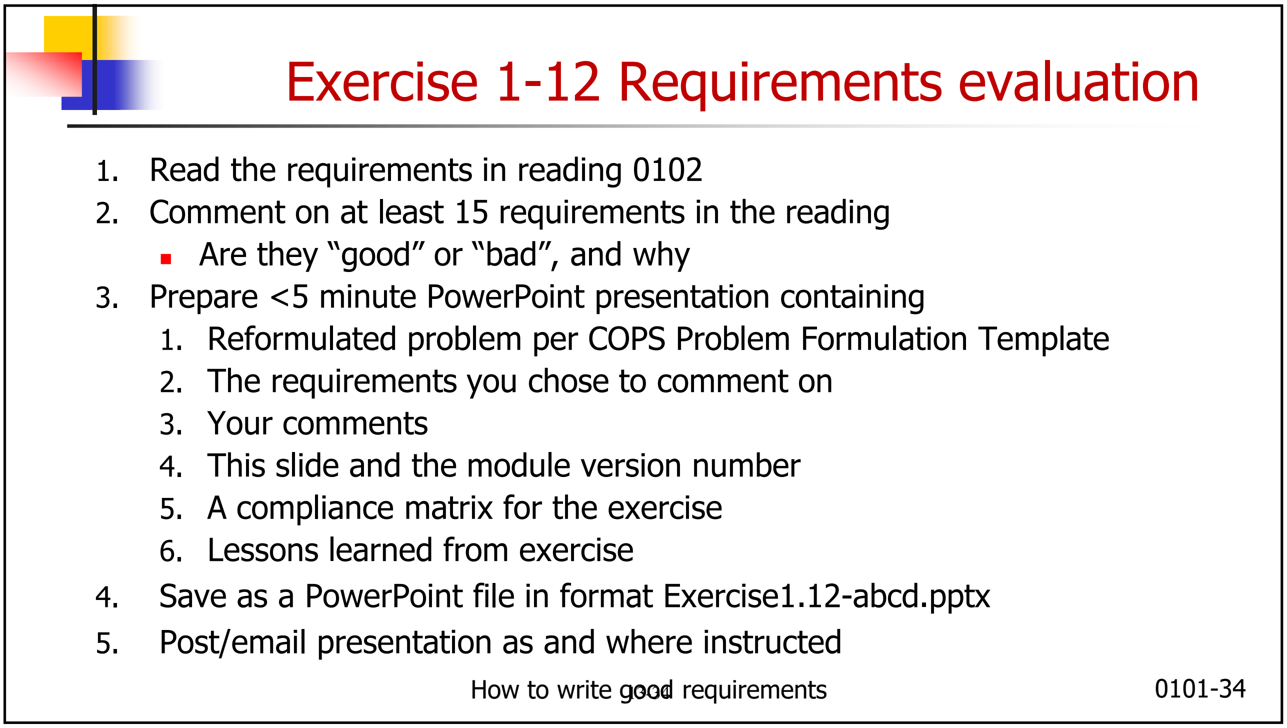
How to write good requirements 0101-32



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

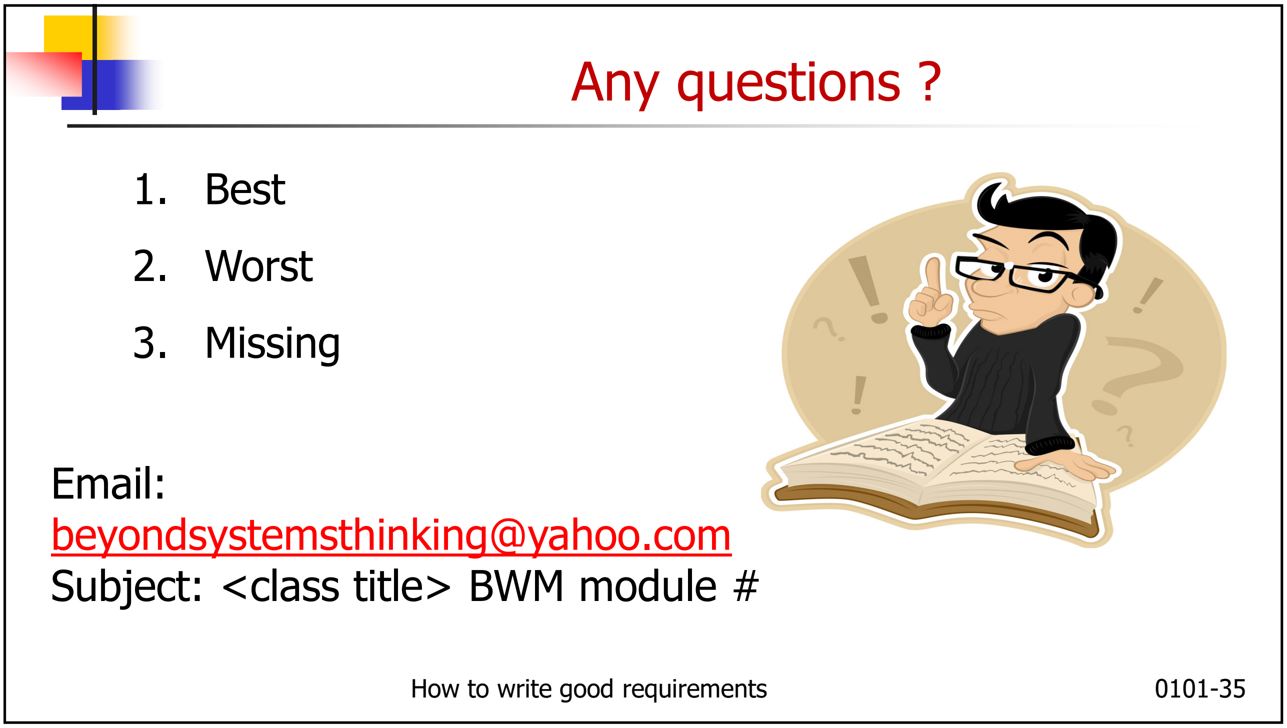
How to write good requirements 0101-33



Exercise 1-12 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. This slide and the module version number
   5. A compliance matrix for the exercise
   6. Lessons learned from exercise
4. Save as a PowerPoint file in format Exercise1.12-abcd.pptx
5. Post/email presentation as and where instructed

How to write good requirements 0101-34



Any questions ?

1. Best
2. Worst
3. Missing

Email:
beyondsystemsthinking@yahoo.com
Subject: <class title> BWM module #

How to write good requirements 0101-35