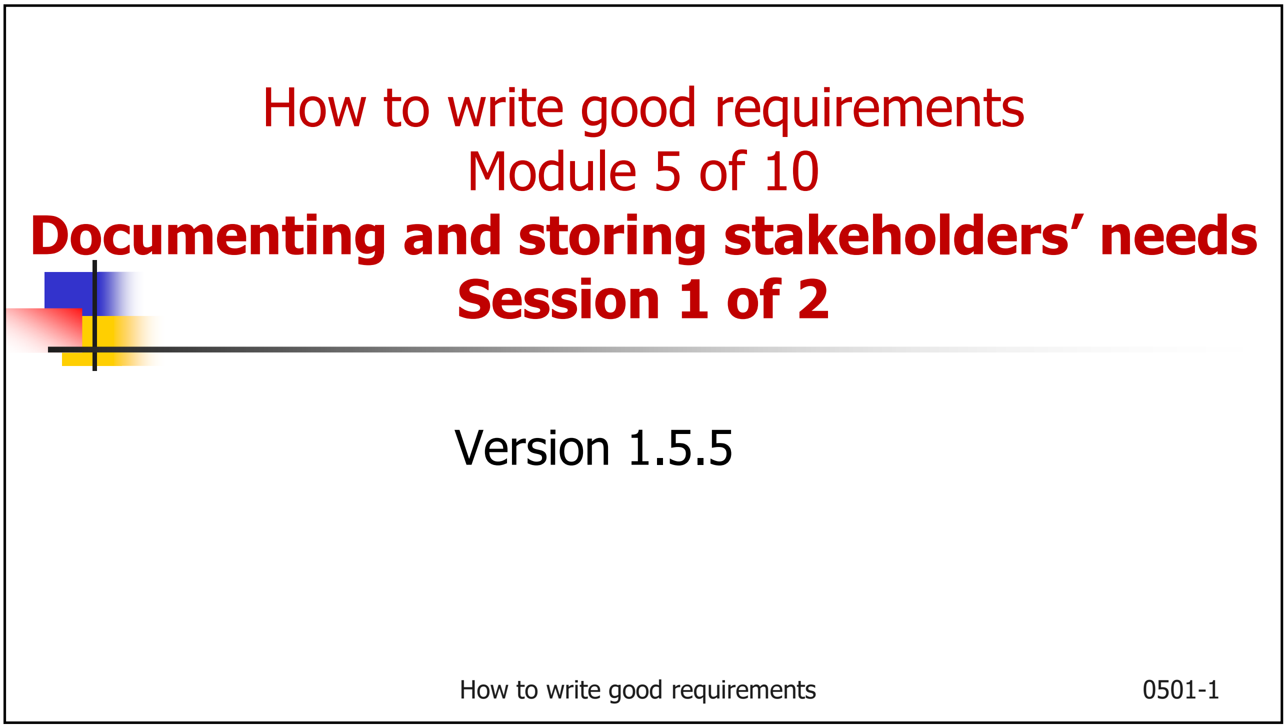
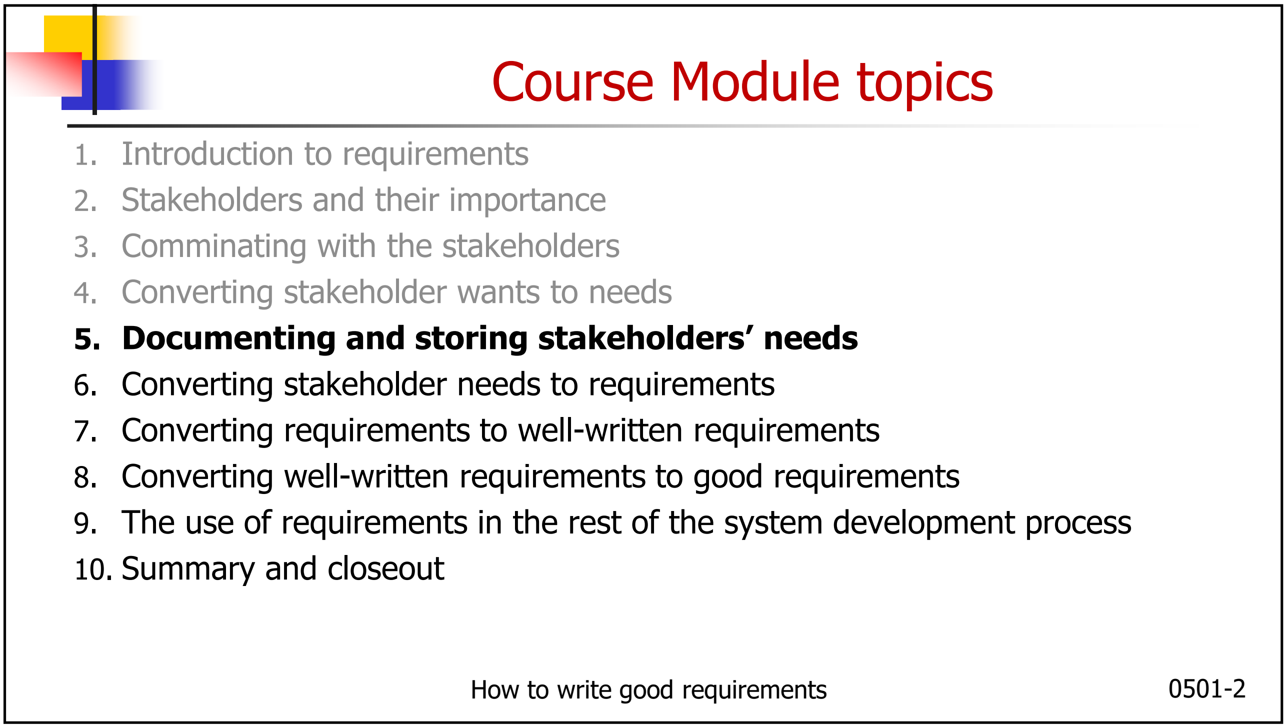


How to write good requirements
Module 5 of 10
Documenting and storing stakeholders’ needs
Session 1 of 2

Version 1.5.5

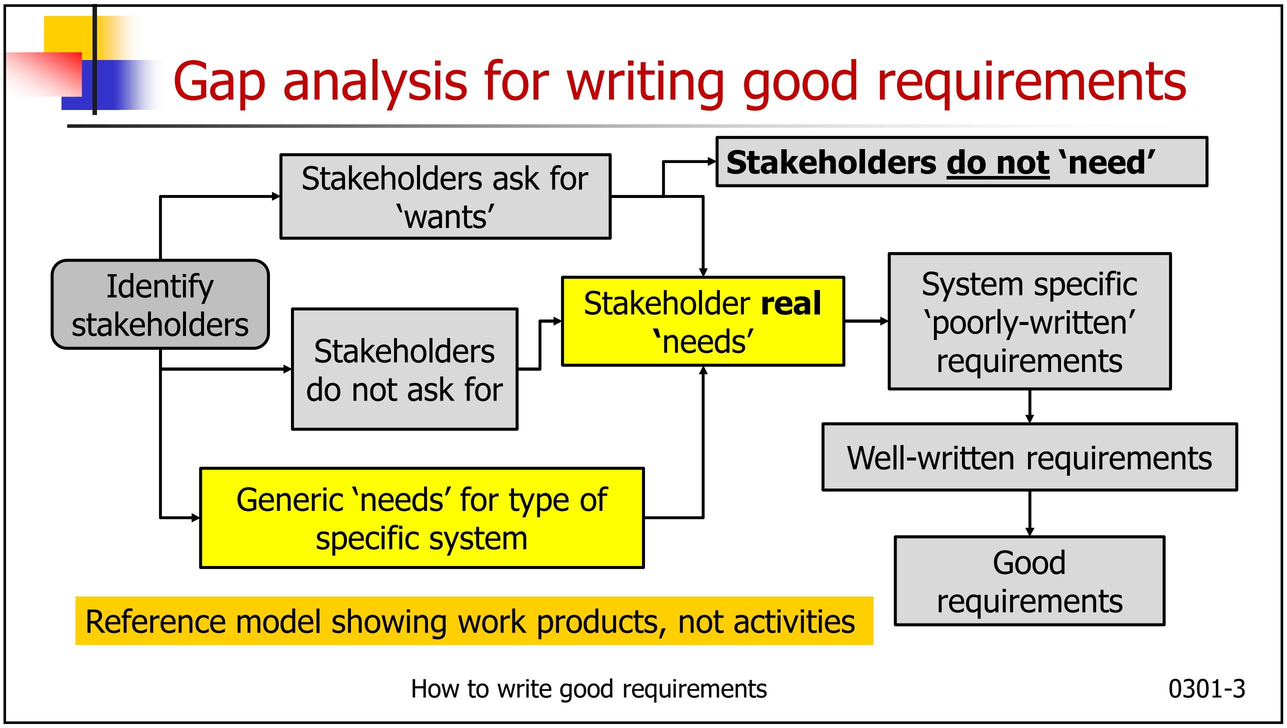
How to write good requirements 0501-1



Course Module topics

1. Introduction to requirements
2. Stakeholders and their importance
3. Comminating with the stakeholders
4. Converting stakeholder wants to needs
5. **Documenting and storing 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 0501-2

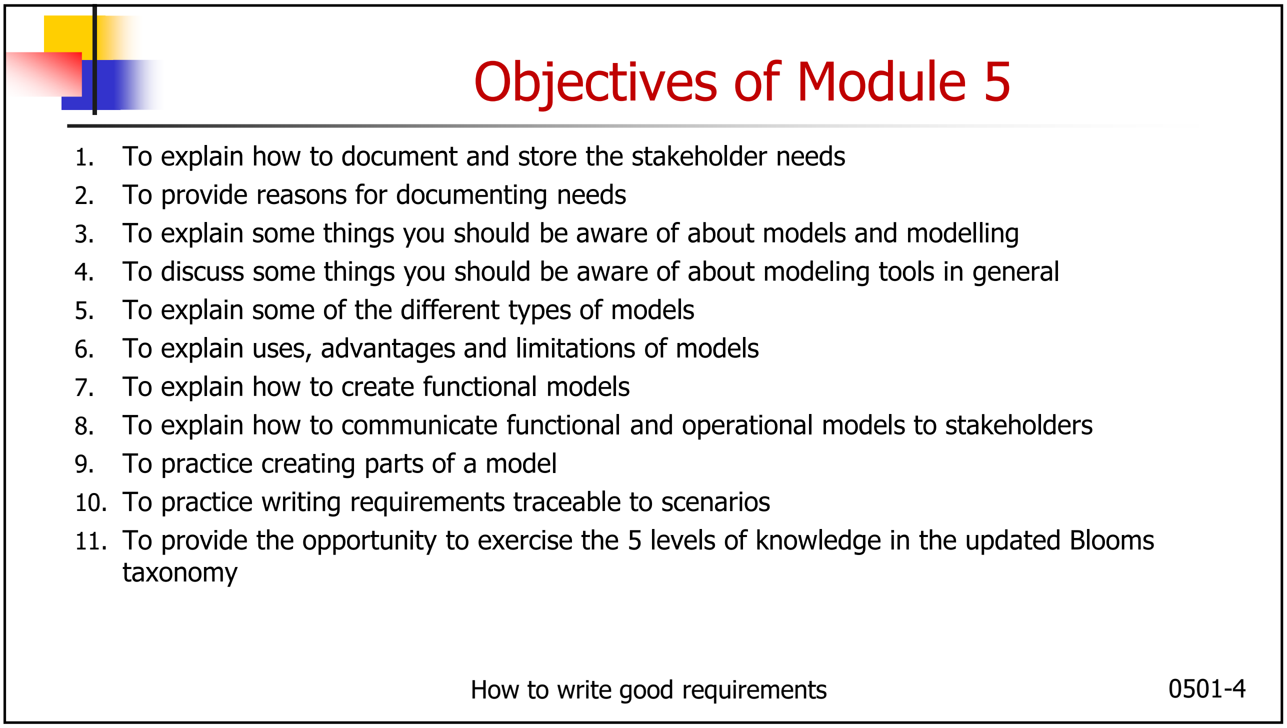


Gap analysis for writing good requirements

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

How to write good requirements 0301-3



Objectives of Module 5

1. To explain how to document and store the stakeholder needs
2. To provide reasons for documenting needs
3. To explain some things you should be aware of about models and modelling
4. To discuss some things you should be aware of about modeling tools in general
5. To explain some of the different types of models
6. To explain uses, advantages and limitations of models
7. To explain how to create functional models
8. To explain how to communicate functional and operational models to stakeholders
9. To practice creating parts of a model
10. To practice writing requirements traceable to scenarios
11. To provide the opportunity to exercise the 5 levels of knowledge in the updated Blooms taxonomy

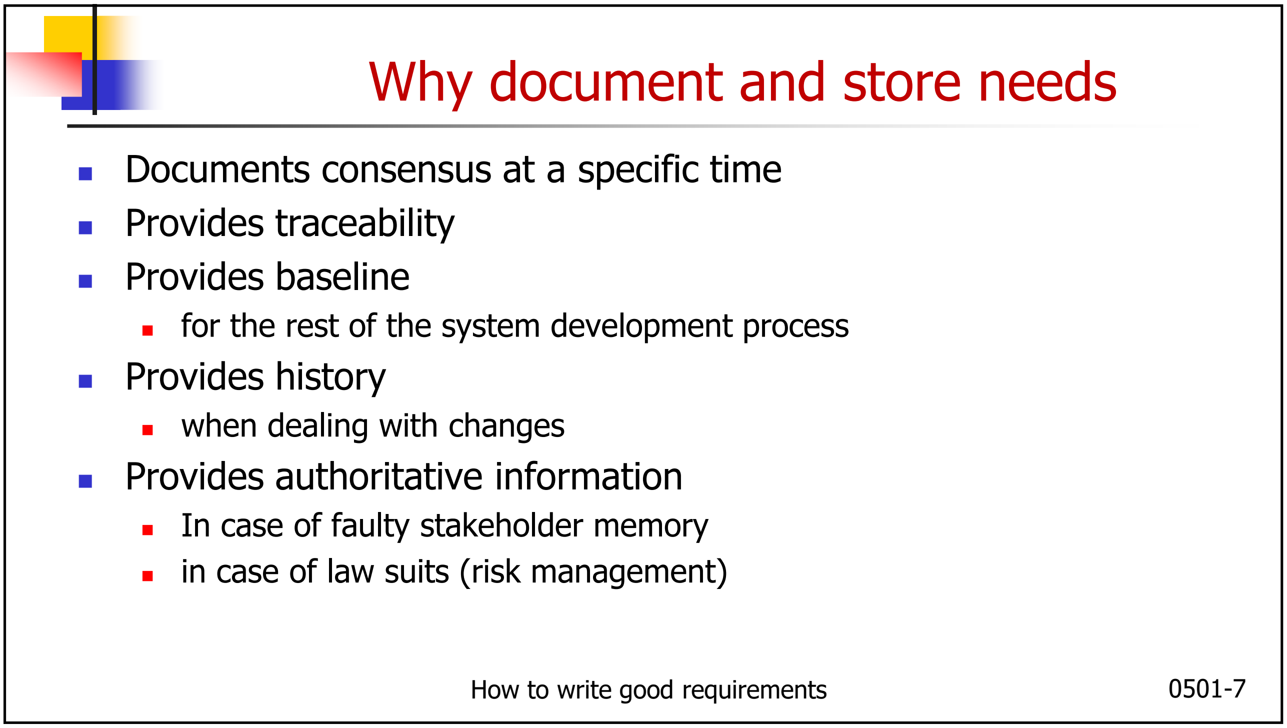
How to write good requirements 0501-4

Knowledge components

- Lecture
  - Sets the context and provides overview
- Readings
  - 0501 Hari, A., Kasser J.E, Weiss M.P., How lessons learnt from using QFD led to the evolution of a process for creating quality requirements for complex systems, *Systems Engineering: The Journal of the International Council on Systems Engineering (INCOSE),* Volume 10, Number 1, 2007.
- Exercises
  - 5-11 Knowledge reading 0501
  - 5-21 Creating scenarios
  - 5-22 Deriving requirements from scenarios
    - The point of this exercise may not be what you think it is
  - 5-23 The real requirement

Module topics

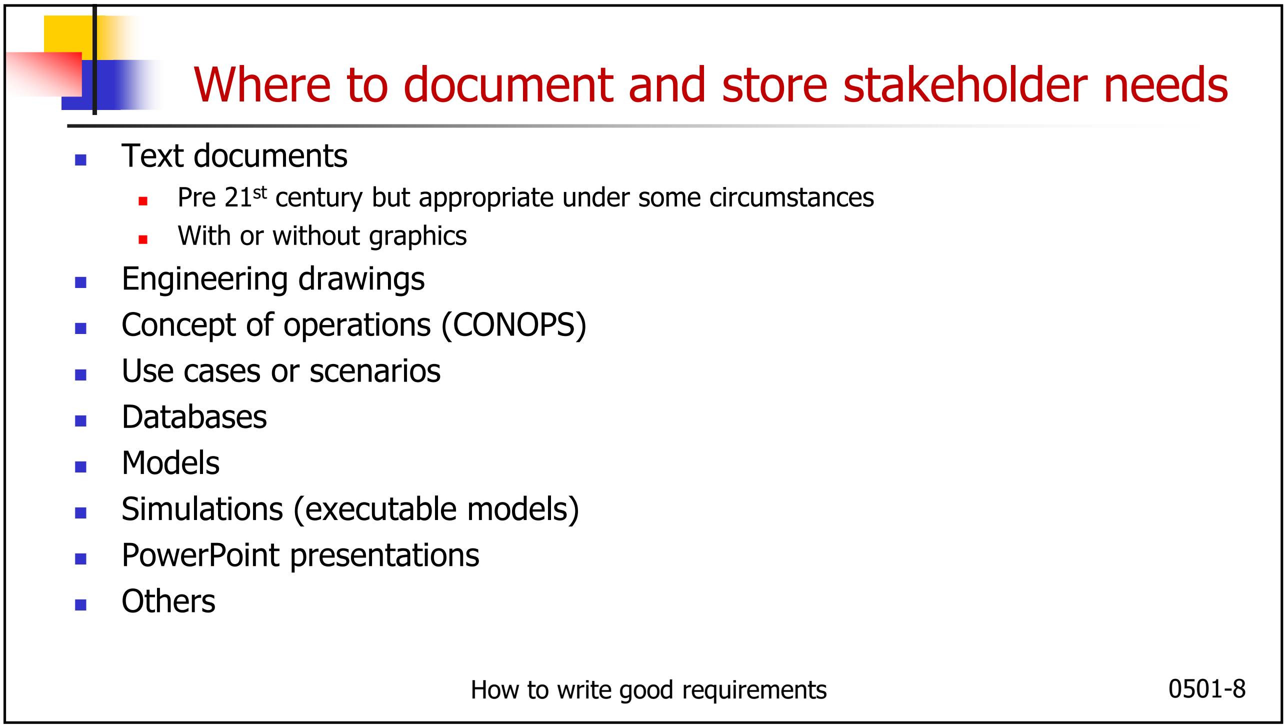
- **Documenting needs**
- Introduction to models
- Some things you should be aware of about modeling tools
- Types of models
- Uses, advantages and limitations of models
- Creating functional models
- Communicating models to stakeholders
- Exercises



Why document and store needs

- Documents consensus at a specific time
- Provides traceability
- Provides baseline
  - for the rest of the system development process
- Provides history
  - when dealing with changes
- Provides authoritative information
  - In case of faulty stakeholder memory
  - in case of law suits (risk management)

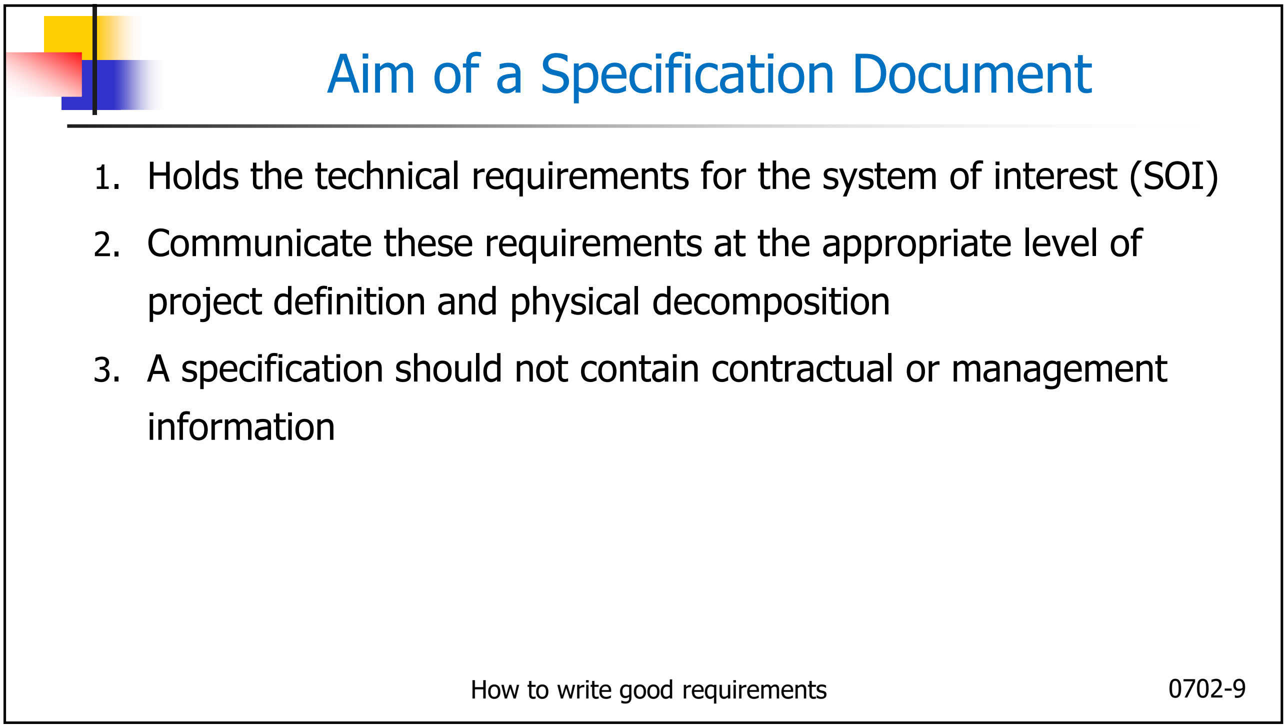
How to write good requirements 0501-7



Where to document and store stakeholder needs

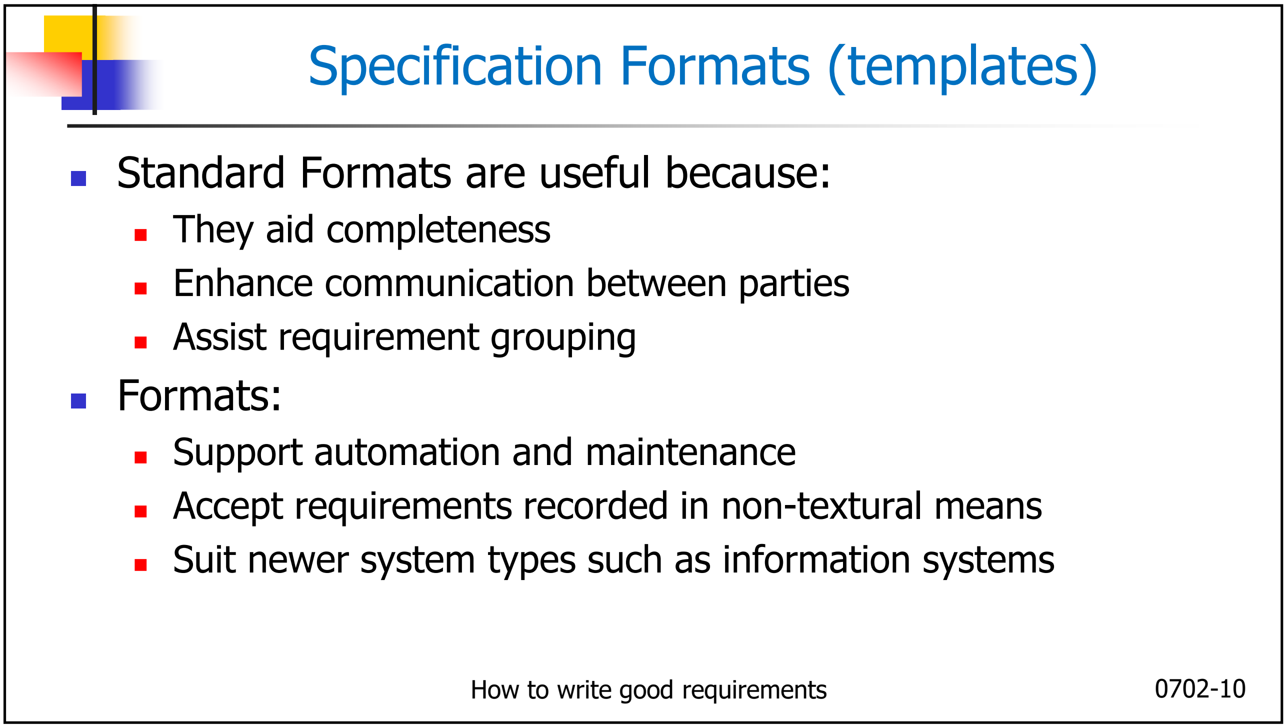
- Text documents
  - Pre 21st century but appropriate under some circumstances
  - With or without graphics
- Engineering drawings
- Concept of operations (CONOPS)
- Use cases or scenarios
- Databases
- Models
- Simulations (executable models)
- PowerPoint presentations
- Others

How to write good requirements 0501-8



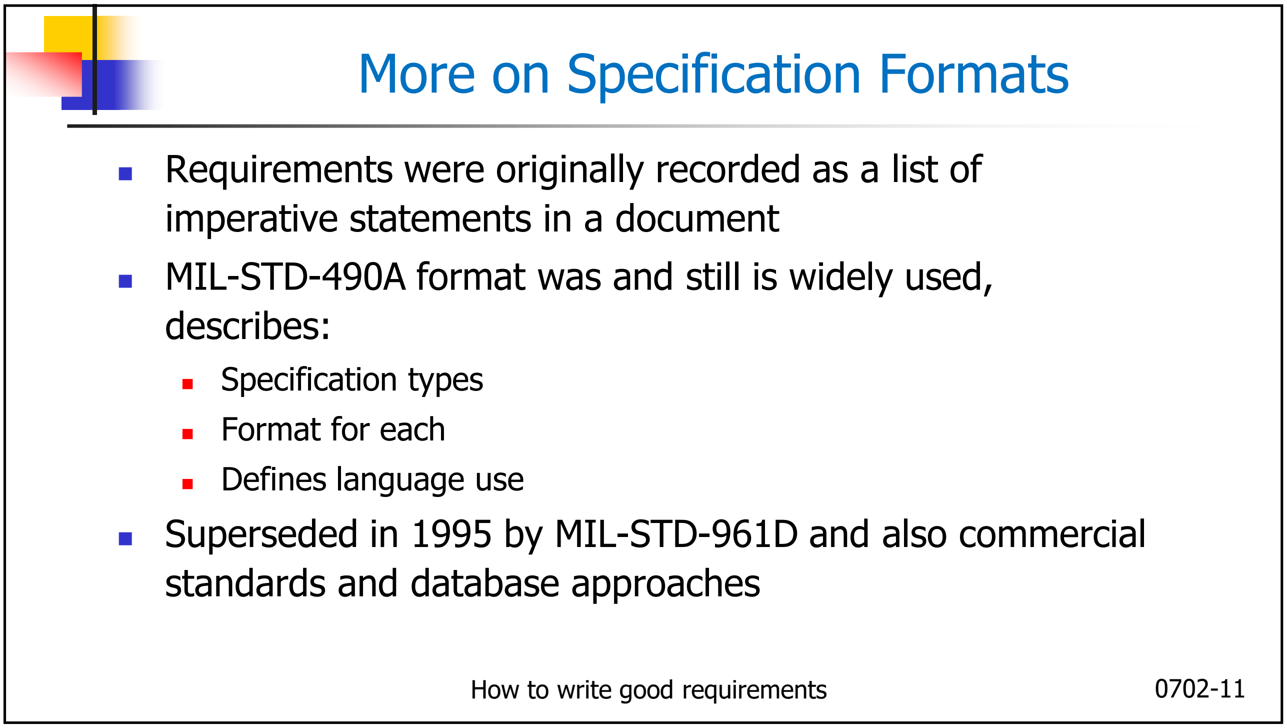
Aim of a Specification Document

1. Holds the technical requirements for the system of interest (SOI)
2. Communicate these requirements at the appropriate level of project definition and physical decomposition
3. A specification should not contain contractual or management information



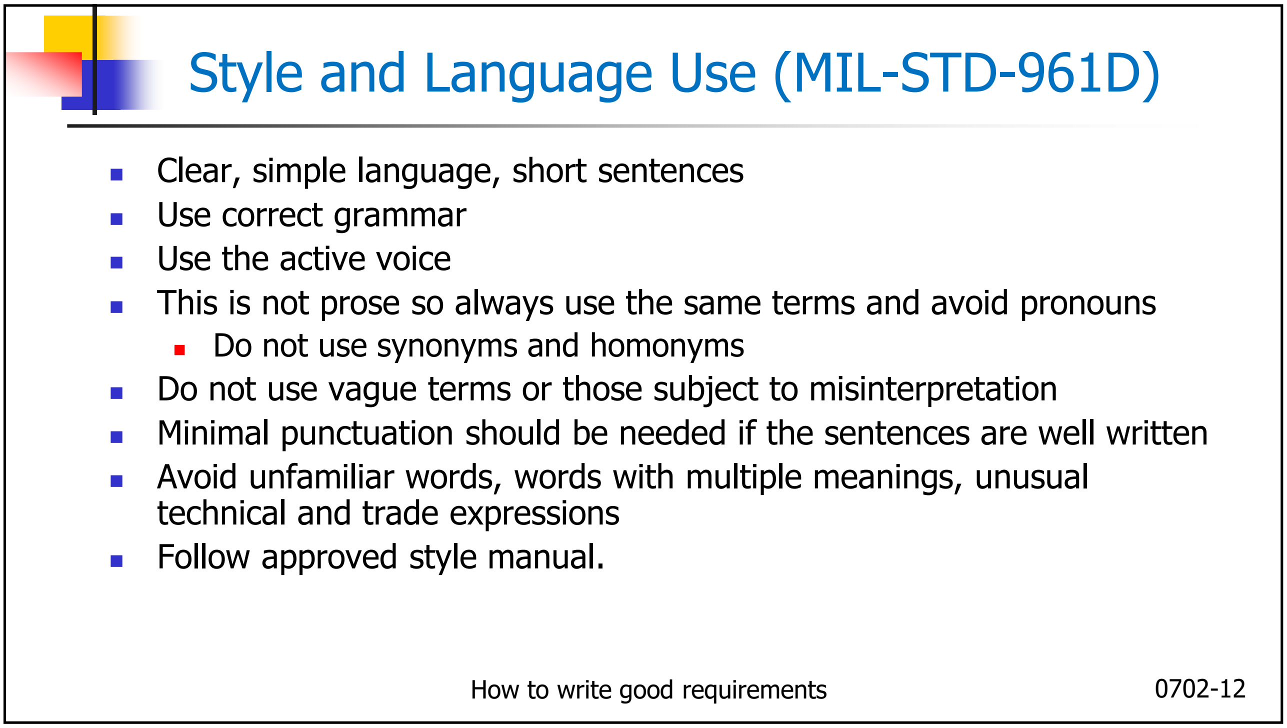
Specification Formats (templates)

- Standard Formats are useful because:
  - They aid completeness
  - Enhance communication between parties
  - Assist requirement grouping
- Formats:
  - Support automation and maintenance
  - Accept requirements recorded in non-textural means
  - Suit newer system types such as information systems



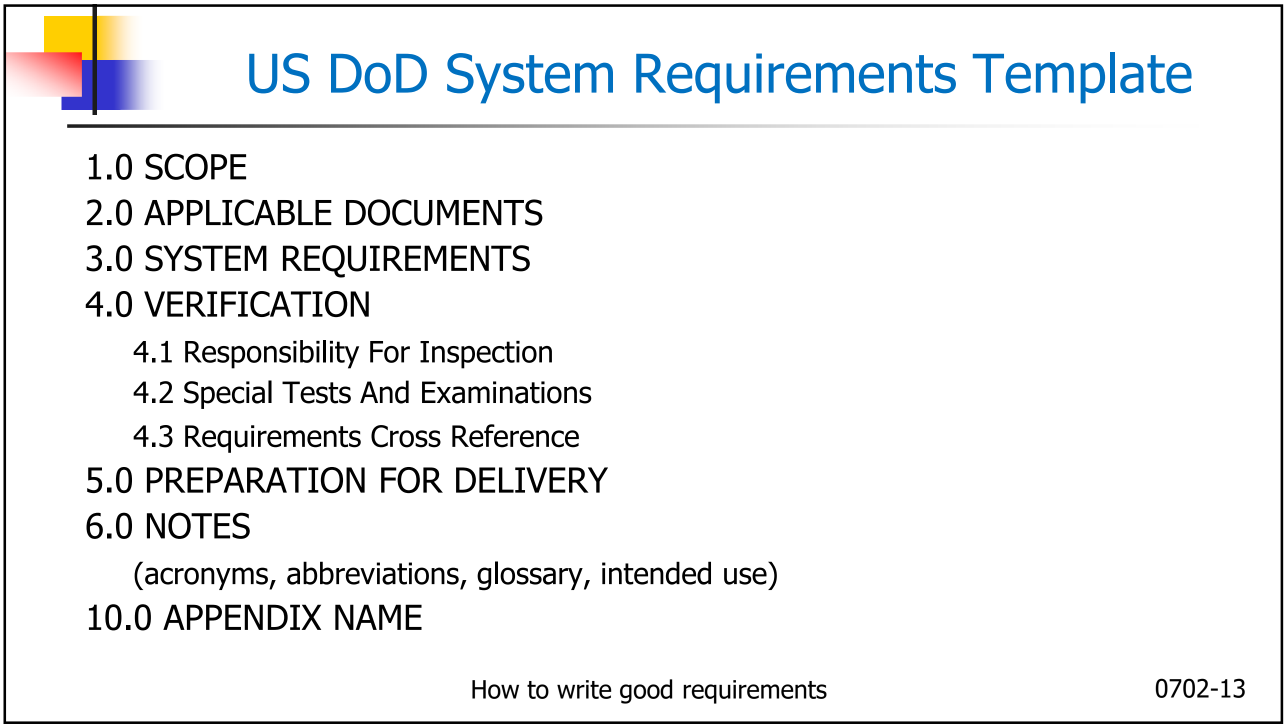
More on Specification Formats

- Requirements were originally recorded as a list of imperative statements in a document
- MIL-STD-490A format was and still is widely used, describes:
  - Specification types
  - Format for each
  - Defines language use
- Superseded in 1995 by MIL-STD-961D and also commercial standards and database approaches



Style and Language Use (MIL-STD-961D)

- Clear, simple language, short sentences
- Use correct grammar
- Use the active voice
- This is not prose so always use the same terms and avoid pronouns
  - Do not use synonyms and homonyms
- Do not use vague terms or those subject to misinterpretation
- Minimal punctuation should be needed if the sentences are well written
- Avoid unfamiliar words, words with multiple meanings, unusual technical and trade expressions
- Follow approved style manual.



US DoD System Requirements Template

1.0 SCOPE

2.0 APPLICABLE DOCUMENTS

3.0 SYSTEM REQUIREMENTS

4.0 VERIFICATION

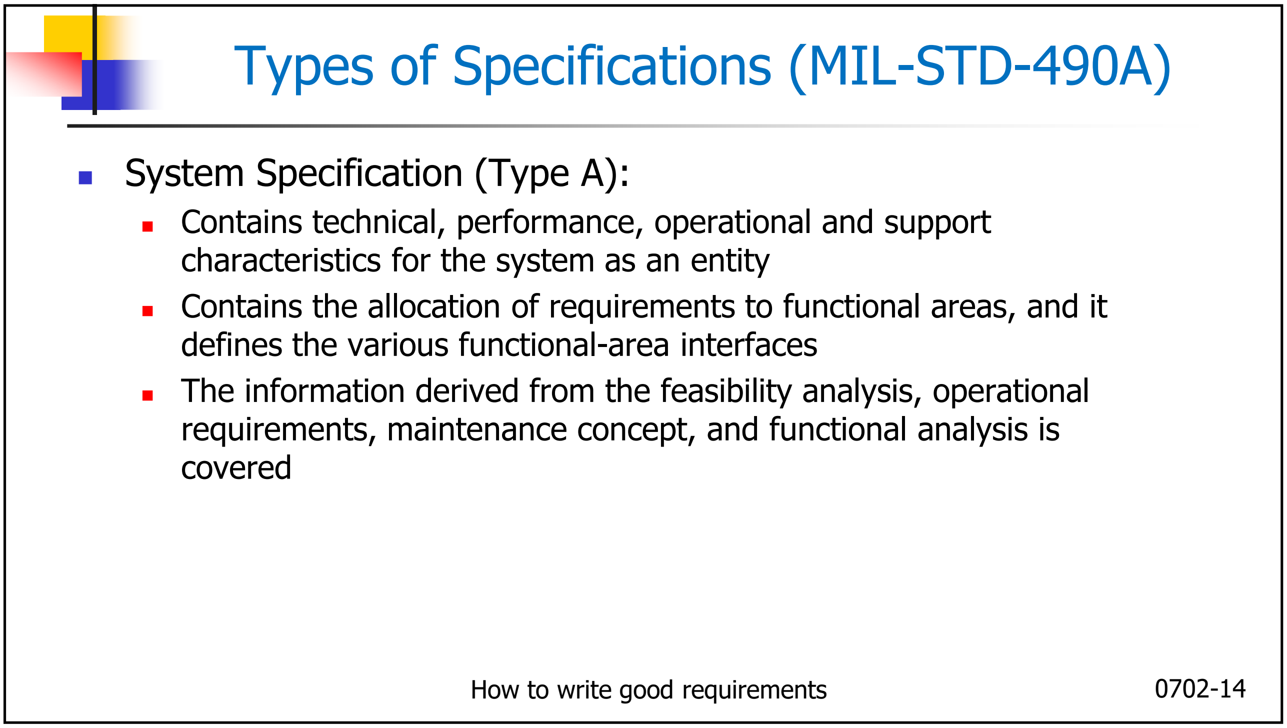
- 4.1 Responsibility For Inspection
- 4.2 Special Tests And Examinations
- 4.3 Requirements Cross Reference

5.0 PREPARATION FOR DELIVERY

6.0 NOTES

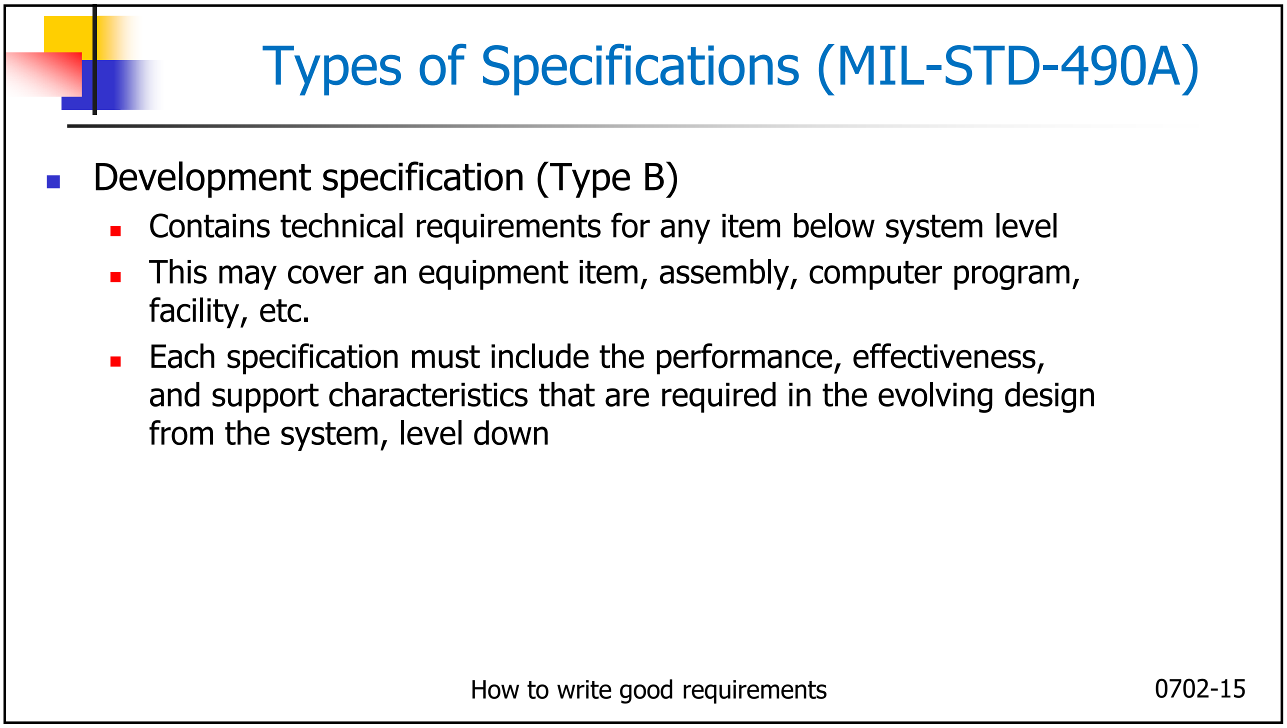
(acronyms, abbreviations, glossary, intended use)

10.0 APPENDIX NAME



Types of Specifications (MIL-STD-490A)

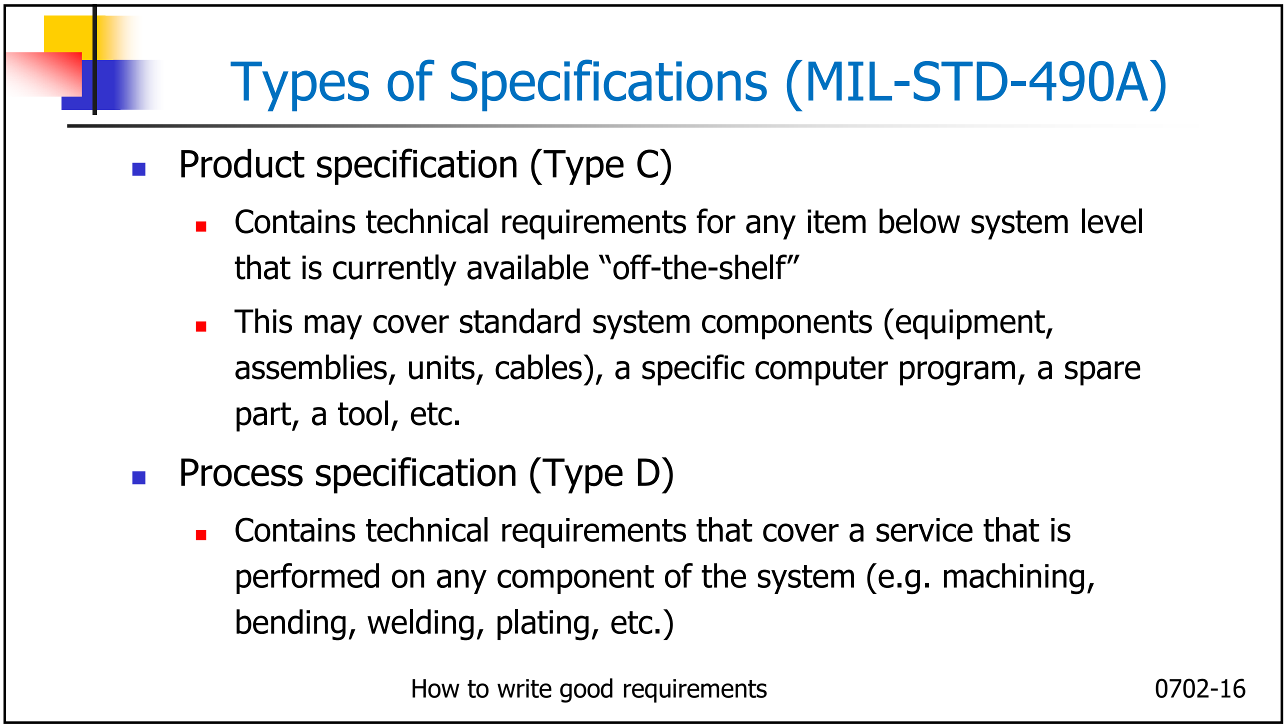
- System Specification (Type A):
  - Contains technical, performance, operational and support characteristics for the system as an entity
  - Contains the allocation of requirements to functional areas, and it defines the various functional-area interfaces
  - The information derived from the feasibility analysis, operational requirements, maintenance concept, and functional analysis is covered



Types of Specifications (MIL-STD-490A)

- Development specification (Type B)
  - Contains technical requirements for any item below system level
  - This may cover an equipment item, assembly, computer program, facility, etc.
  - Each specification must include the performance, effectiveness, and support characteristics that are required in the evolving design from the system, level down

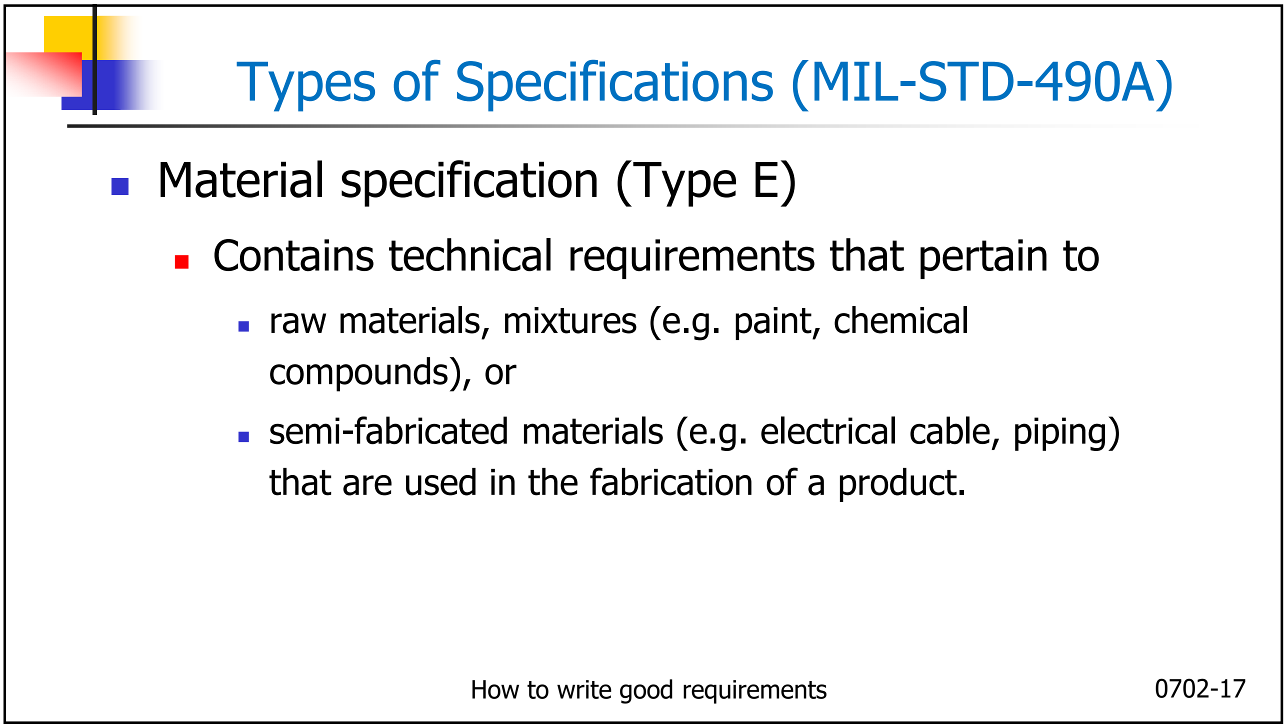
How to write good requirements 0702-15



Types of Specifications (MIL-STD-490A)

- Product specification (Type C)
  - Contains technical requirements for any item below system level that is currently available "off-the-shelf"
  - This may cover standard system components (equipment, assemblies, units, cables), a specific computer program, a spare part, a tool, etc.
- Process specification (Type D)
  - Contains technical requirements that cover a service that is performed on any component of the system (e.g. machining, bending, welding, plating, etc.)

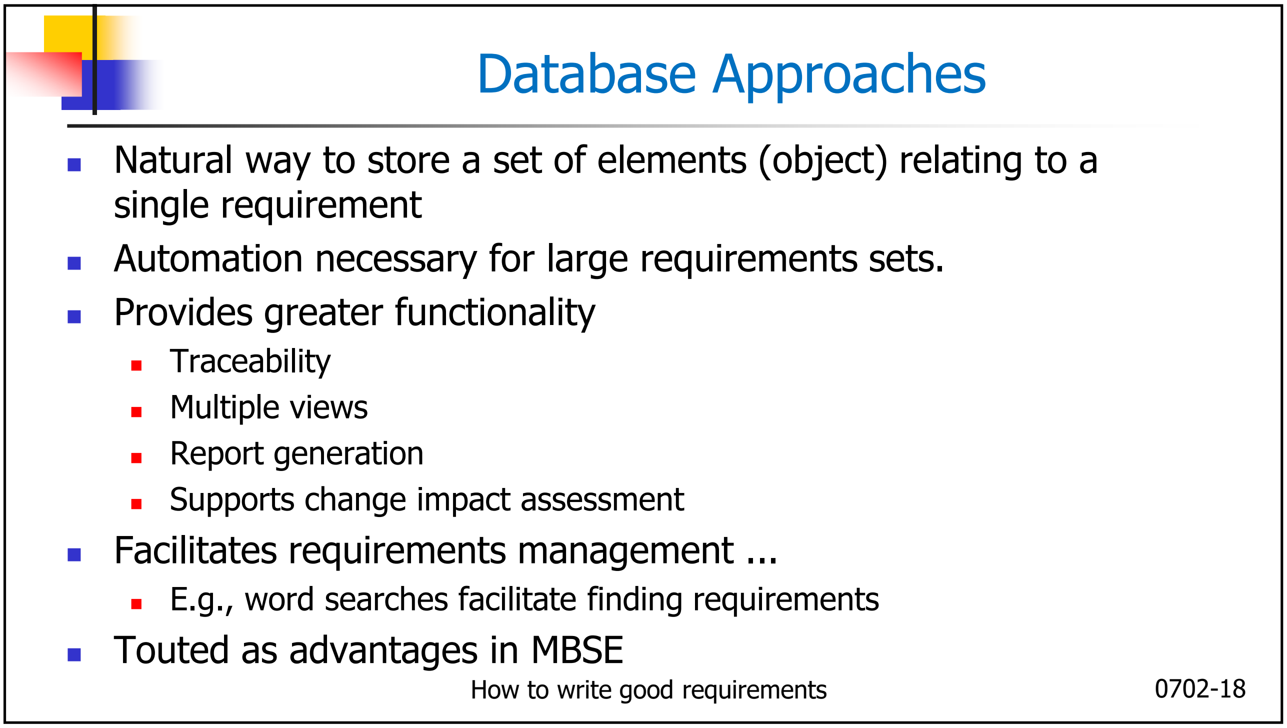
How to write good requirements 0702-16



Types of Specifications (MIL-STD-490A)

- Material specification (Type E)
  - Contains technical requirements that pertain to
    - raw materials, mixtures (e.g. paint, chemical compounds), or
    - semi-fabricated materials (e.g. electrical cable, piping) that are used in the fabrication of a product.

How to write good requirements 0702-17



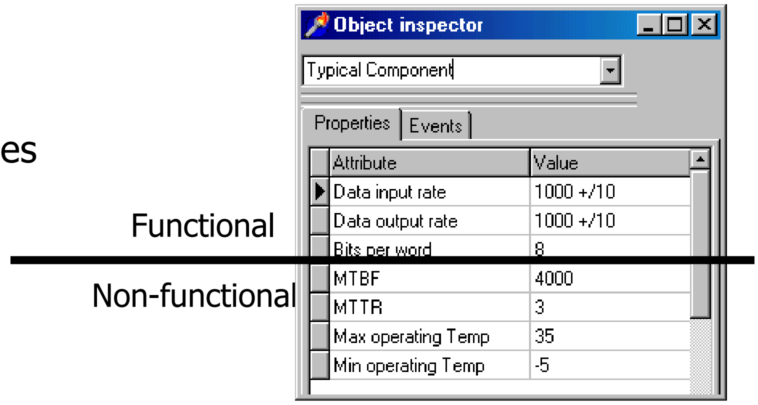
Database Approaches

- Natural way to store a set of elements (object) relating to a single requirement
- Automation necessary for large requirements sets.
- Provides greater functionality
  - Traceability
  - Multiple views
  - Report generation
  - Supports change impact assessment
- Facilitates requirements management ...
  - E.g., word searches facilitate finding requirements
- Touted as advantages in MBSE

How to write good requirements 0702-18

Documenting non-functional needs

- Non-functional models
- Text statements
- Tables and databases
- Properties and attributes



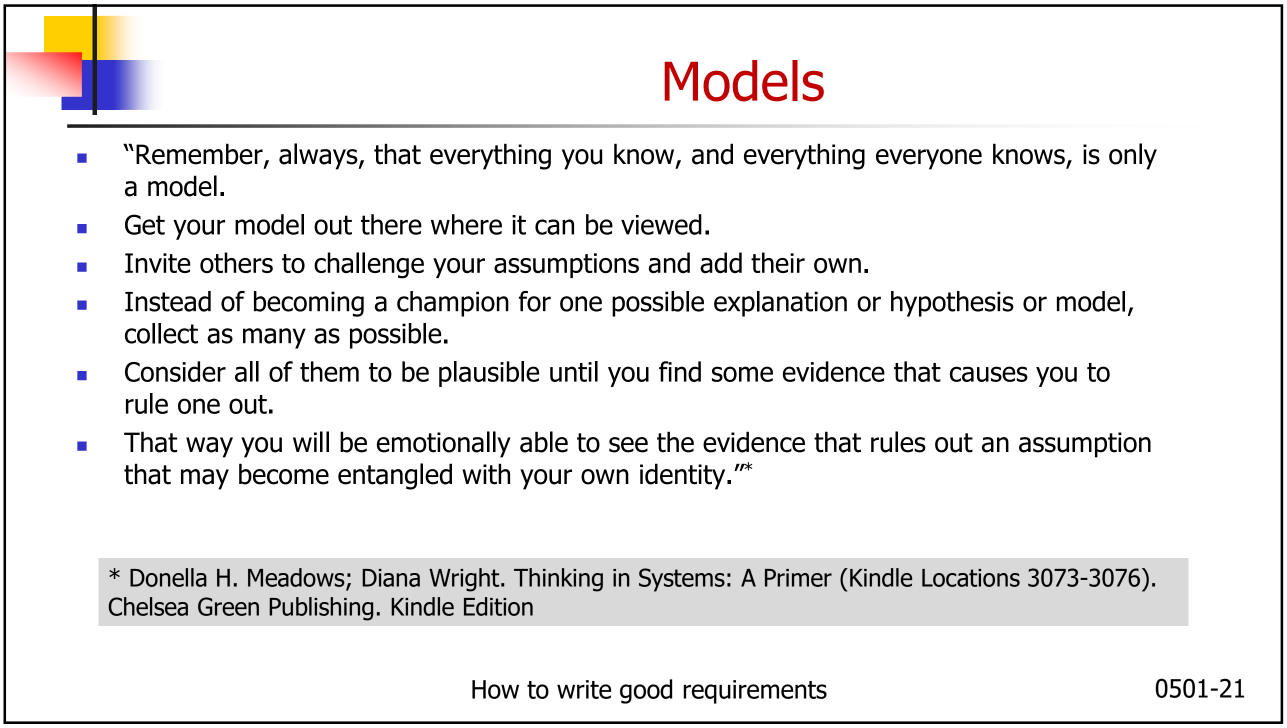
Attribute	Value
Data input rate	1000 +/10
Data output rate	1000 +/10
Bits per word	8
MTBF	4000
MTTR	3
Max operating Temp	35
Min operating Temp	-5

How to write good requirements 0501-19

Module topics

- Documenting needs
- **Introduction to models**
- Some things you should be aware of about modeling tools
- Types of models
- Uses, advantages and limitations of models
- Creating functional models
- Communicating models to stakeholders
- Exercises

How to write good requirements 0501-20

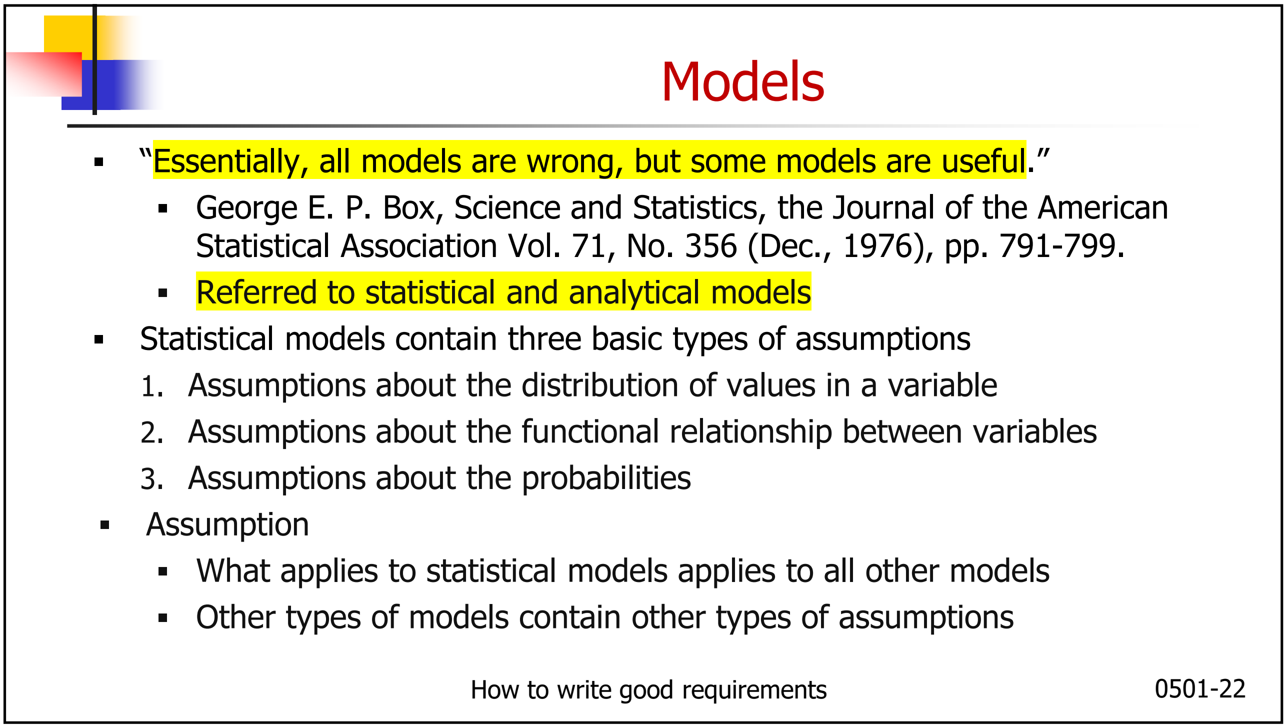


Models

- “Remember, always, that everything you know, and everything everyone knows, is only a model.
- Get your model out there where it can be viewed.
- Invite others to challenge your assumptions and add their own.
- Instead of becoming a champion for one possible explanation or hypothesis or model, collect as many as possible.
- Consider all of them to be plausible until you find some evidence that causes you to rule one out.
- That way you will be emotionally able to see the evidence that rules out an assumption that may become entangled with your own identity.”*

* Donella H. Meadows; Diana Wright. Thinking in Systems: A Primer (Kindle Locations 3073-3076). Chelsea Green Publishing. Kindle Edition

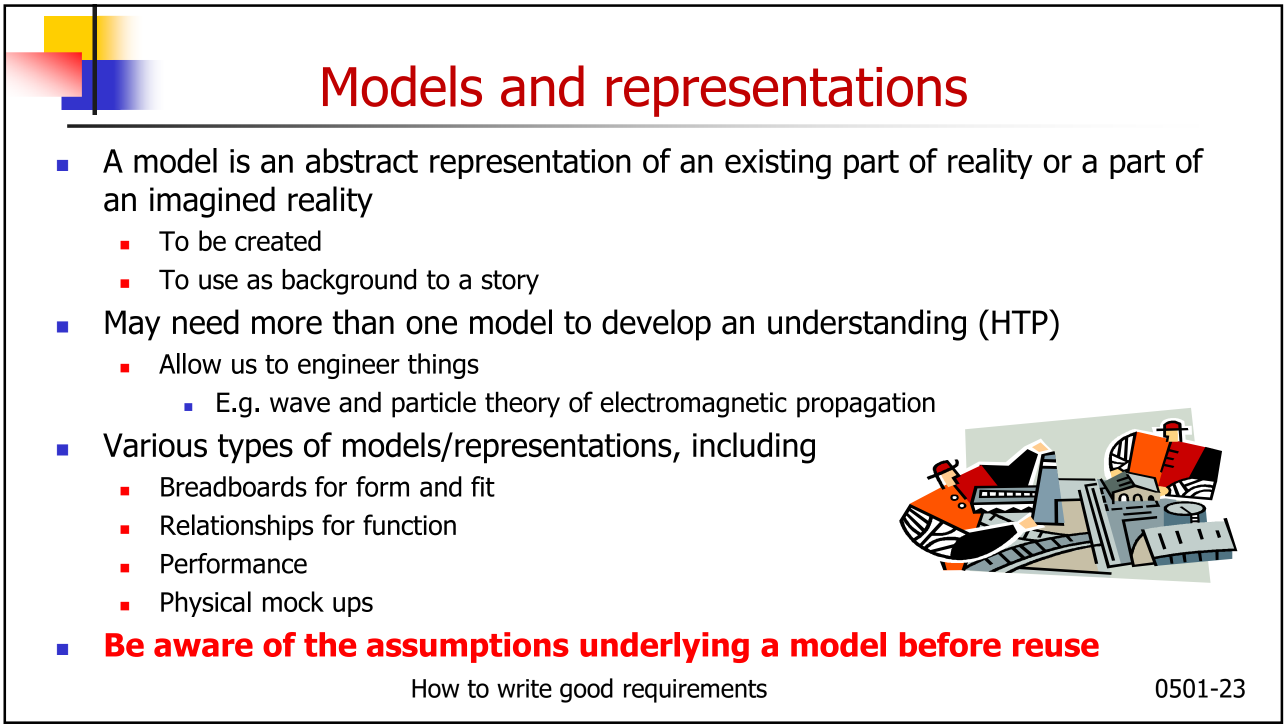
How to write good requirements 0501-21



Models

- “Essentially, all models are wrong, but some models are useful.”
  - George E. P. Box, Science and Statistics, the Journal of the American Statistical Association Vol. 71, No. 356 (Dec., 1976), pp. 791-799.
  - Referred to statistical and analytical models
- Statistical models contain three basic types of assumptions
  1. Assumptions about the distribution of values in a variable
  2. Assumptions about the functional relationship between variables
  3. Assumptions about the probabilities
- Assumption
  - What applies to statistical models applies to all other models
  - Other types of models contain other types of assumptions

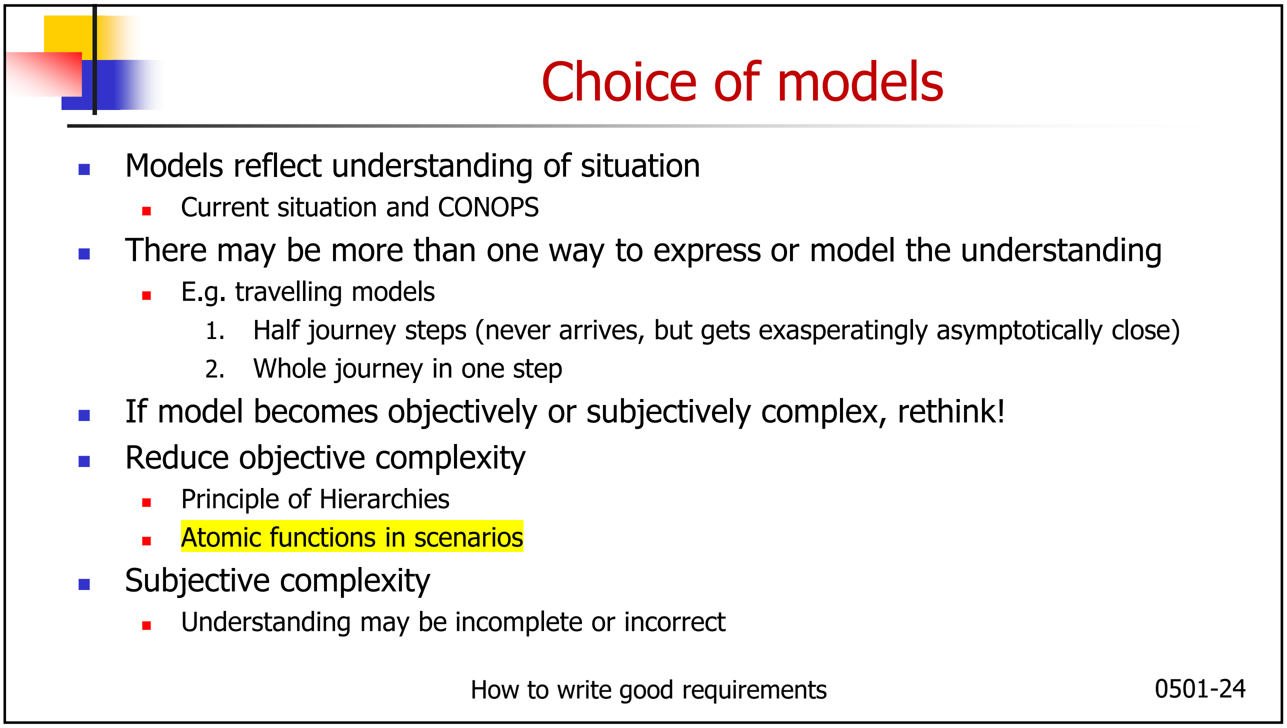
How to write good requirements 0501-22



Models and representations

- A model is an abstract representation of an existing part of reality or a part of an imagined reality
  - To be created
  - To use as background to a story
- May need more than one model to develop an understanding (HTP)
  - Allow us to engineer things
    - E.g. wave and particle theory of electromagnetic propagation
- Various types of models/representations, including
  - Breadboards for form and fit
  - Relationships for function
  - Performance
  - Physical mock ups
- **Be aware of the assumptions underlying a model before reuse**

How to write good requirements 0501-23

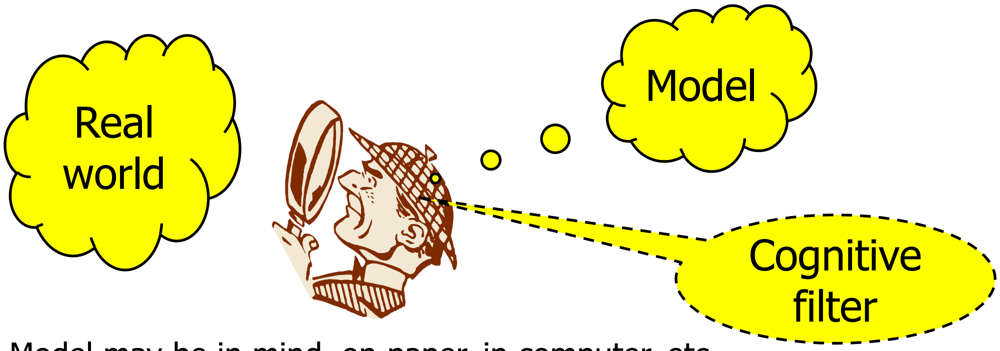


Choice of models

- Models reflect understanding of situation
  - Current situation and CONOPS
- There may be more than one way to express or model the understanding
  - E.g. travelling models
    1. Half journey steps (never arrives, but gets exasperatingly asymptotically close)
    2. Whole journey in one step
- If model becomes objectively or subjectively complex, rethink!
- Reduce objective complexity
  - Principle of Hierarchies
  - Atomic functions in scenarios
- Subjective complexity
  - Understanding may be incomplete or incorrect

How to write good requirements 0501-24

Building models



Real world

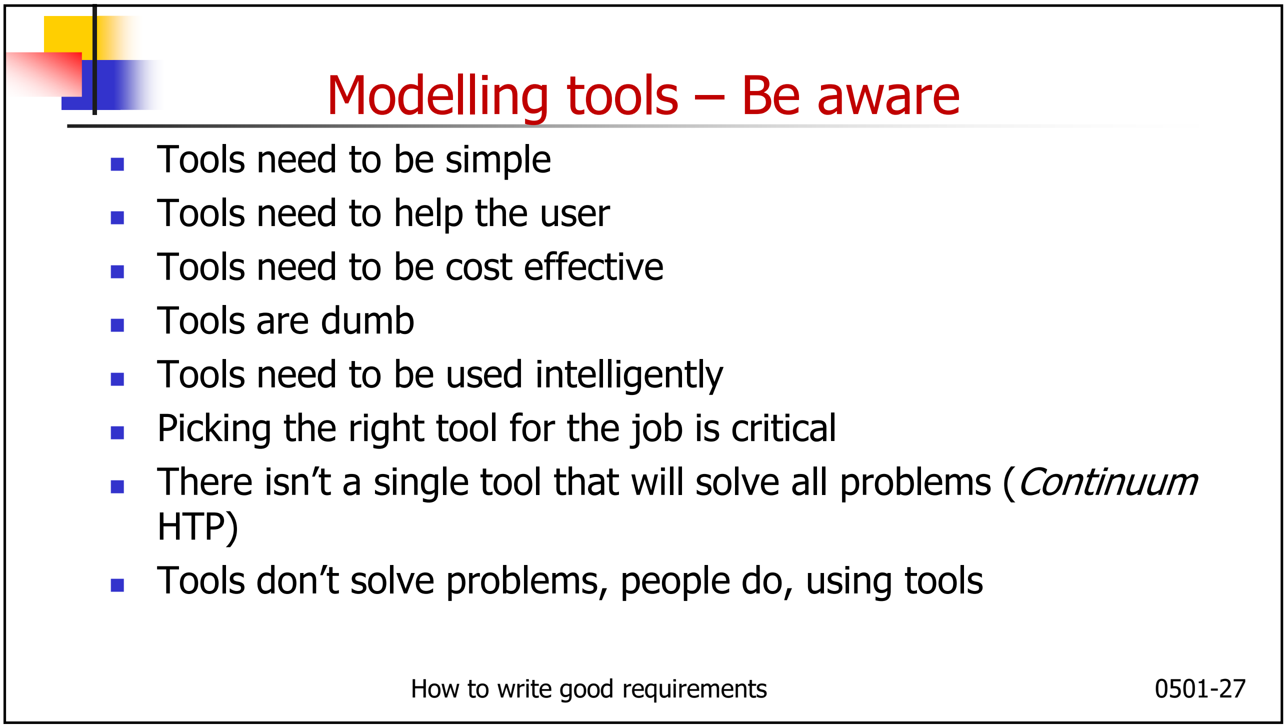
Model

Cognitive filter

- Model may be in mind, on paper, in computer, etc.
- Cognitive filters (Holistic Thinking page 22) are one cause of the model diverging from the real world
- Models based on observation cannot be validated without a reference
- Generic flaw in Standards

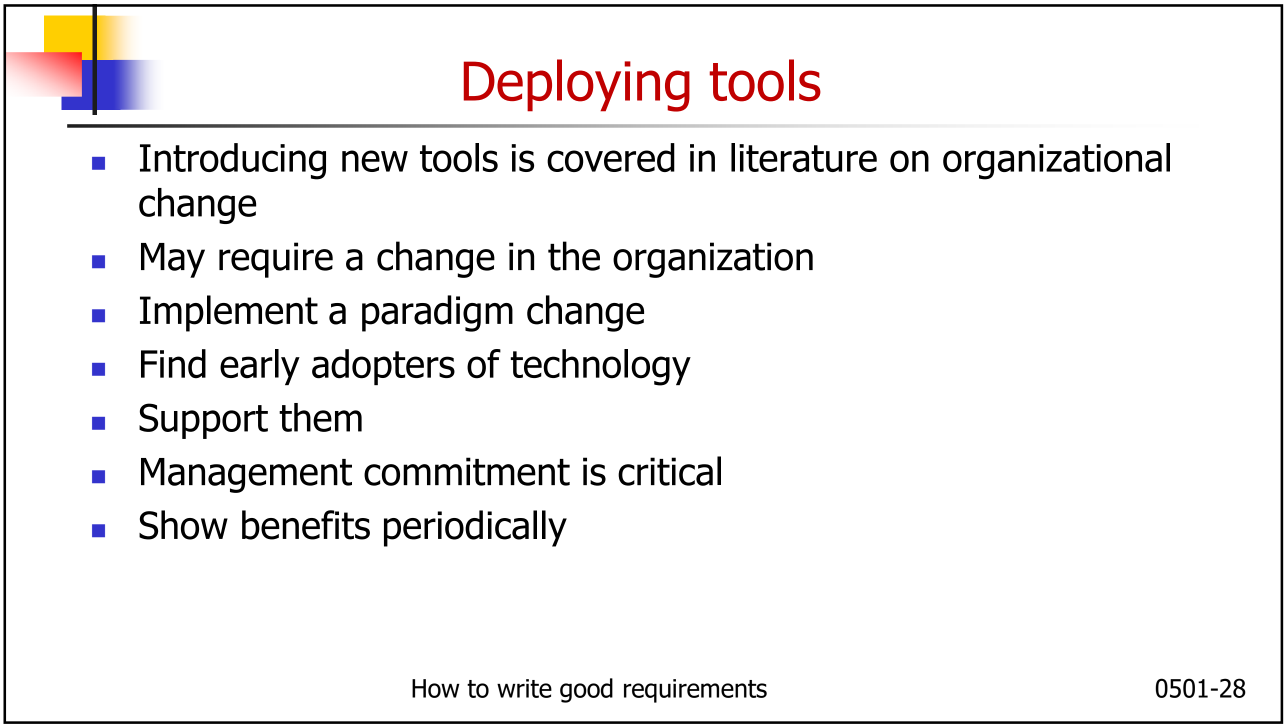
Module topics

- Documenting needs
- Introduction to models
- **Some things you should be aware of about modeling tools**
- Types of models
- Uses, advantages and limitations of models
- Creating functional models
- Communicating models to stakeholders
- Exercises



Modelling tools – Be aware

- Tools need to be simple
- Tools need to help the user
- Tools need to be cost effective
- Tools are dumb
- Tools need to be used intelligently
- Picking the right tool for the job is critical
- There isn’t a single tool that will solve all problems (*Continuum* HTP)
- Tools don’t solve problems, people do, using tools



Deploying tools

- Introducing new tools is covered in literature on organizational change
- May require a change in the organization
- Implement a paradigm change
- Find early adopters of technology
- Support them
- Management commitment is critical
- Show benefits periodically

Compatibility issues

- Types of tools
- Types of knowledge
- Representation of knowledge
- Communications between tools
- Different tools in
  - Customer-contractor
  - Prime contractor-subcontractor
  - Within distributed organizations
  - Other places

Module topics

- Documenting needs
- Introduction to models
- Some things you should be aware of about modeling tools
- **Types of models**
- Uses, advantages and limitations of models
- Creating functional models
- Communicating models to stakeholders
- Exercises

Some types of models

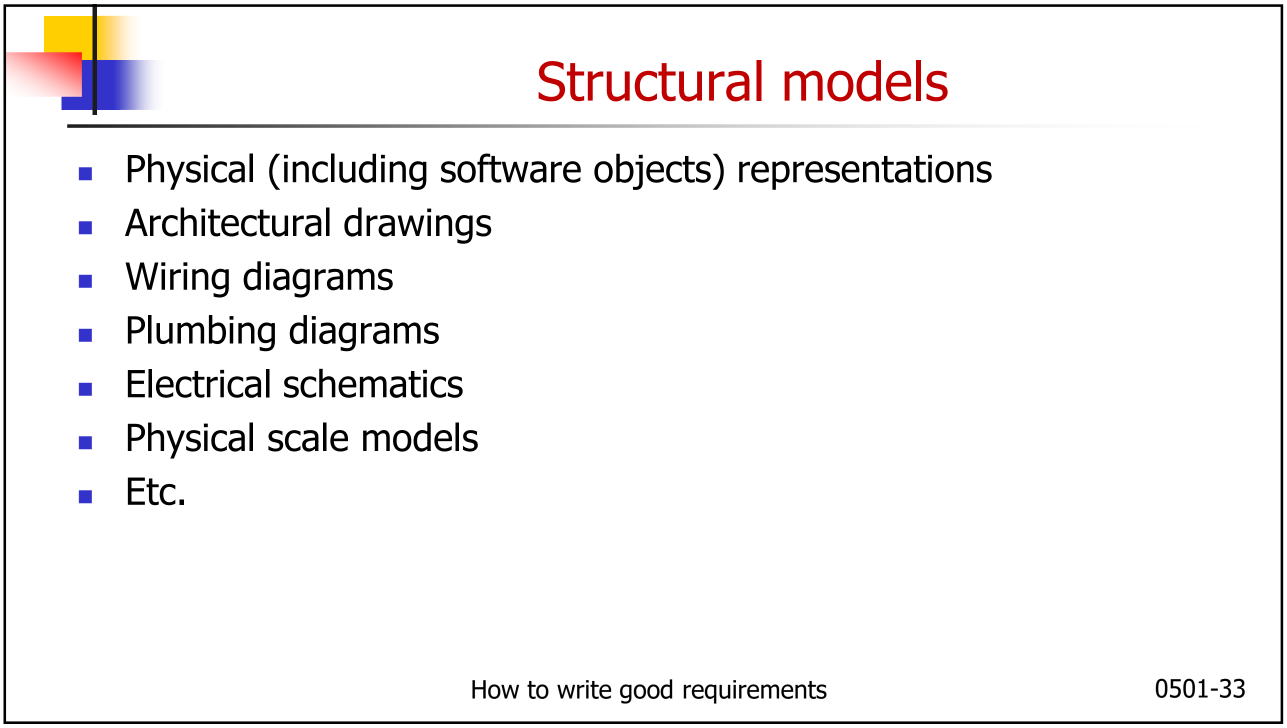
- Context models
- Structural models
- Function and flow models
- Pace layer model
- KANO model of customer satisfaction
- Financial models
- Others

How to write good requirements 0501-31

Context models

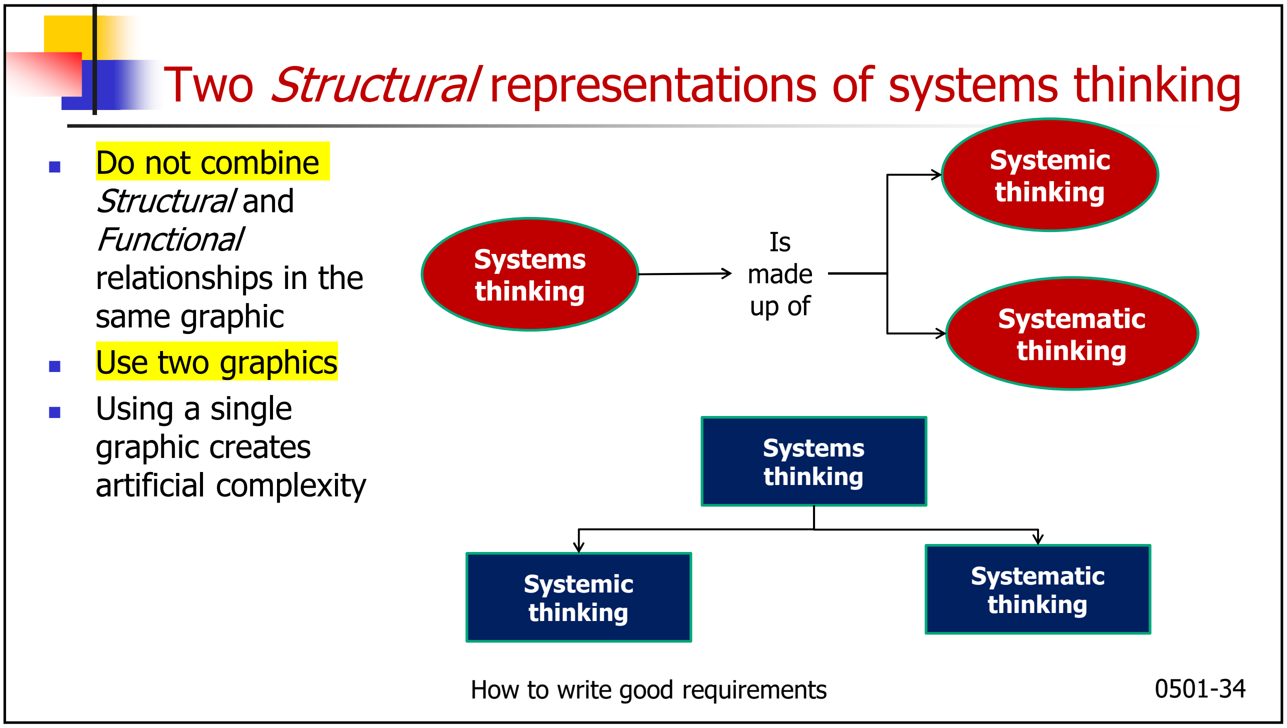
- *Big Picture* and *Operational*
- Describe a system and the adjacent systems that interact with the system
- Sketch the interfaces between a system and its context (e.g., in terms of what information is exchanged)
- Contain context diagrams
    - Used for expressing use cases or scenarios
    - Have no standardized notation
    - UML and SysML used by INCOSE

How to write good requirements 0501-32



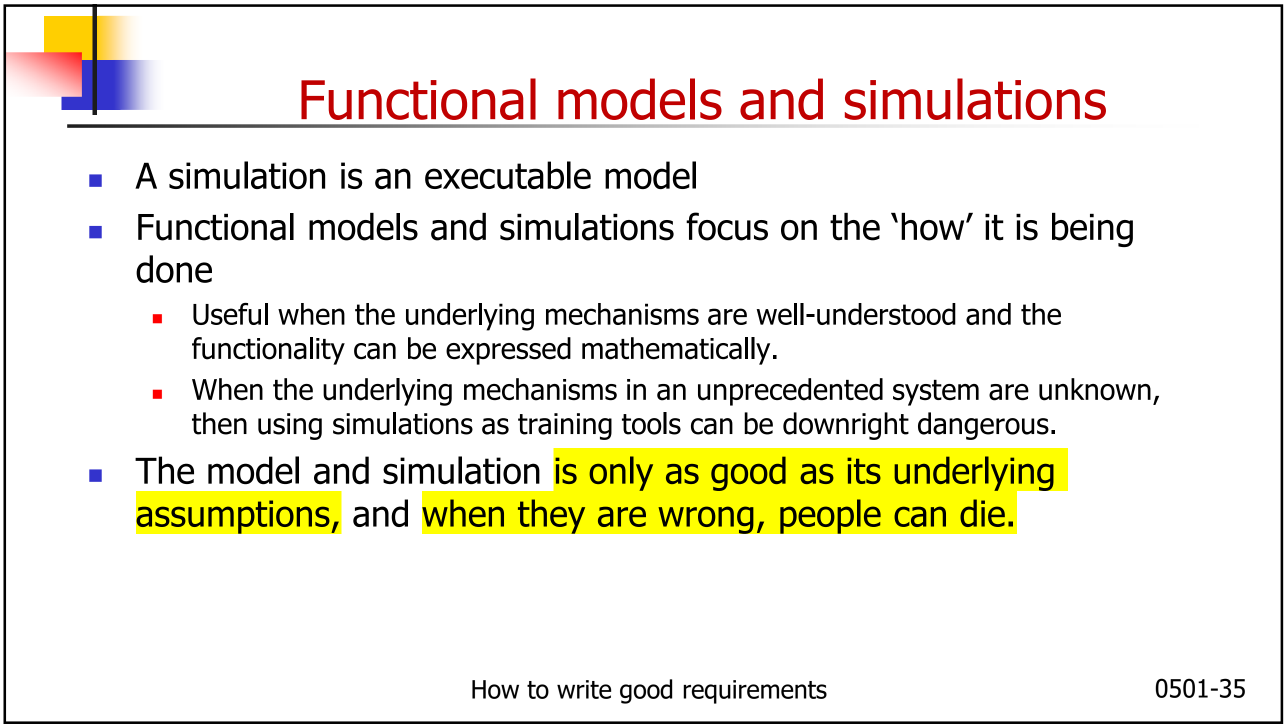
Structural models

- Physical (including software objects) representations
- Architectural drawings
- Wiring diagrams
- Plumbing diagrams
- Electrical schematics
- Physical scale models
- Etc.



Two *Structural* representations of systems thinking

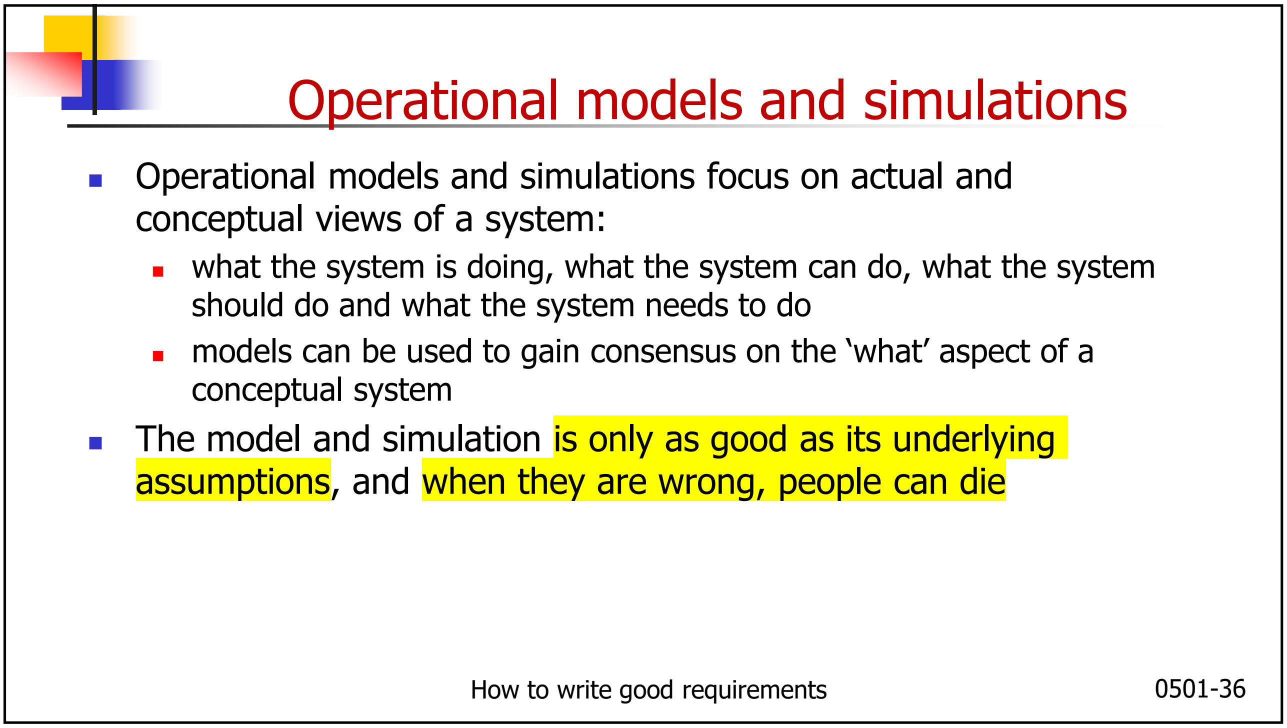
- Do not combine *Structural* and *Functional* relationships in the same graphic
- Use two graphics
- Using a single graphic creates artificial complexity



Functional models and simulations

- A simulation is an executable model
- Functional models and simulations focus on the ‘how’ it is being done
  - Useful when the underlying mechanisms are well-understood and the functionality can be expressed mathematically.
  - When the underlying mechanisms in an unprecedented system are unknown, then using simulations as training tools can be downright dangerous.
- The model and simulation is only as good as its underlying assumptions, and when they are wrong, people can die.

How to write good requirements 0501-35



Operational models and simulations

- Operational models and simulations focus on actual and conceptual views of a system:
  - what the system is doing, what the system can do, what the system should do and what the system needs to do
  - models can be used to gain consensus on the ‘what’ aspect of a conceptual system
- The model and simulation is only as good as its underlying assumptions, and when they are wrong, people can die

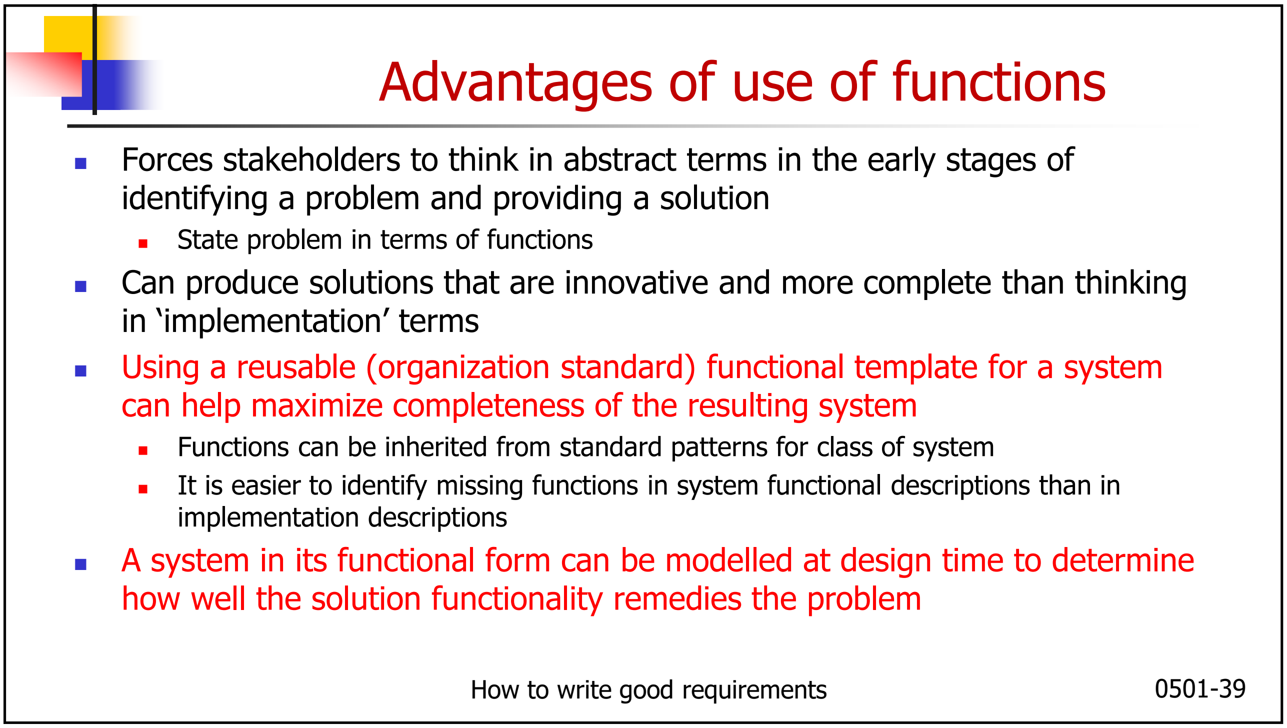
How to write good requirements 0501-36

Simulations (executable models)

- *Functional*, *Operational* and *Quantitative* HTPs
- Relationships
- Effects
- Dynamic
- Games
- Training
- Others

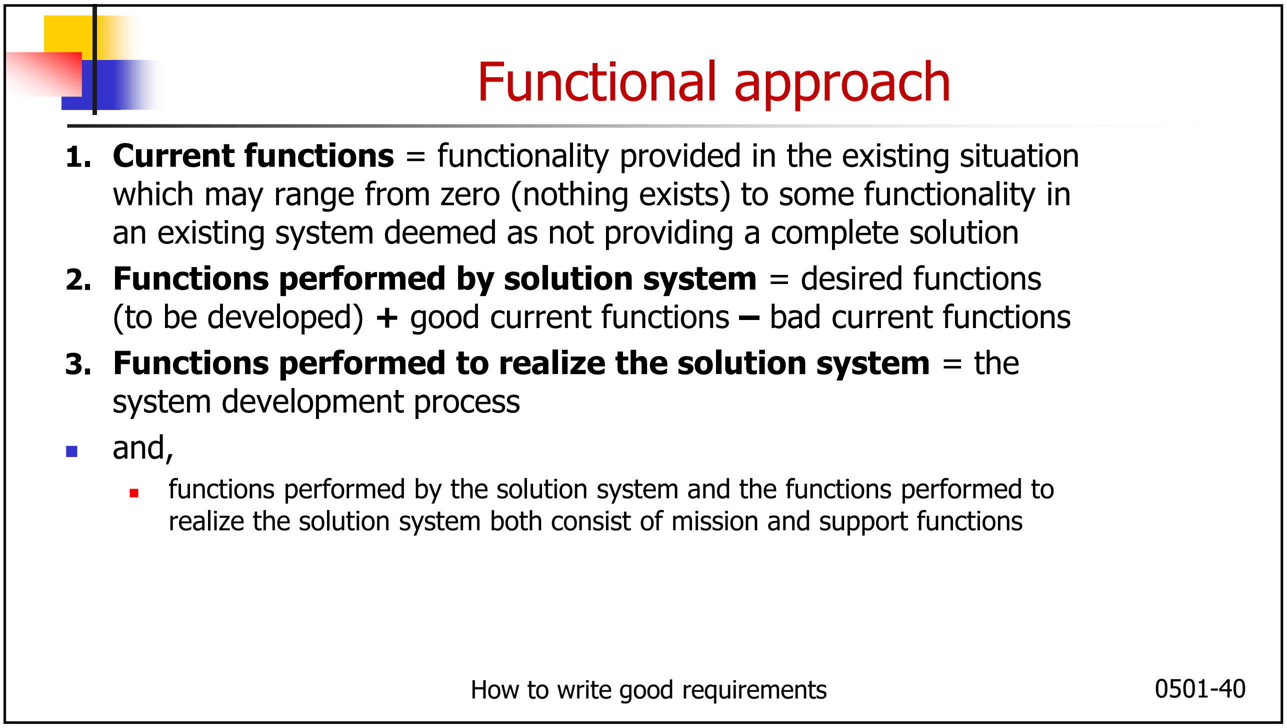
Function and flow models

- Models that focus on function and flow aspects specify
  - Requirements for the sequence of actions required to produce the required results from given inputs
  - The actions required to execute a (business) process, including the flow of control and data between the actions and who is responsible for which action.
- Activity models are used to specify system functions
- Activity diagrams can also express who is responsible for which action
- Process models are used to describe business processes or technical processes
- Domain story models
  - Specify visual stories about how ‘actors’ interact with devices, artifacts and other items in a domain
  - Are a means for understanding the application domain in which a system will operate.



Advantages of use of functions

- Forces stakeholders to think in abstract terms in the early stages of identifying a problem and providing a solution
  - State problem in terms of functions
- Can produce solutions that are innovative and more complete than thinking in ‘implementation’ terms
- Using a reusable (organization standard) functional template for a system can help maximize completeness of the resulting system
  - Functions can be inherited from standard patterns for class of system
  - It is easier to identify missing functions in system functional descriptions than in implementation descriptions
- A system in its functional form can be modelled at design time to determine how well the solution functionality remedies the problem



Functional approach

1. **Current functions** = functionality provided in the existing situation which may range from zero (nothing exists) to some functionality in an existing system deemed as not providing a complete solution
2. **Functions performed by solution system** = desired functions (to be developed) **+** good current functions **–** bad current functions
3. **Functions performed to realize the solution system** = the system development process

- and,
  - functions performed by the solution system and the functions performed to realize the solution system both consist of mission and support functions

Pace Layer model*

- In any complex system, there are different layers or components that change at different rates or paces. These layers range through
  1. Fast
     - Includes fashion, versions of technological products (cell phones, software product versions), teenager’s moods
  2. Medium
     - Includes management and engineering fads; e.g. Management by Objectivities, Business Process Reengineering, Model-based Systems Engineering, Network-centric-Warfare, (Systems Engineering ?), computer operating systems
  3. Slow
     - Long term stability, e.g. building and logical infrastructure
- Relevance to requirements elicitation and elucidation
  1. Probability of the stability of the requirement (how quickly might it change)
  2. May be more applicable in the Subsystem Design States of the System Development Process

* Brand, S., The Clock Of The Long Now: Time and Responsibility, Phoenix Paperbacks, 2000

How to write good requirements 0501-41

KANO model*

- A model of customer satisfaction with a simple ranking scheme to distinguish between essential and differentiating attributes related to concepts of customer satisfaction**
- Applicable to new product development for a mass market
- Two dimensions
  1. Achievement
     - Ranges from (low) the supplier didn’t do it at all, to the supplier did it very well (high)
  2. Satisfaction
     - Ranges from total dissatisfaction with the product or service (low), to total satisfaction with the product or service (high)

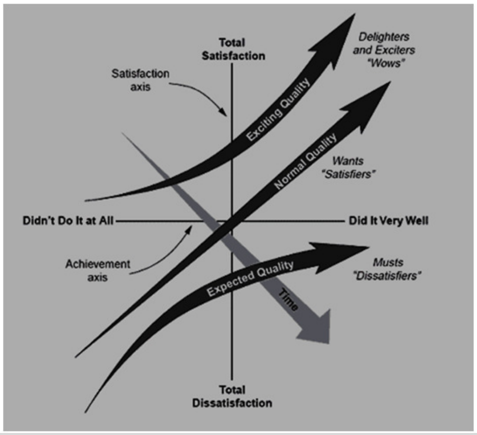
* *Kano, Noriaki; Nobuhiku Seraku; Fumio Takahashi; Shinichi Tsuji (April 1984).* *"Attractive quality and must-be quality"*. Journal of the Japanese Society for Quality Control 14(2), 147-156, 1984-04-15, The Japanese Society for Quality Control
** ChatGPT, accessed 8 October 2023

How to write good requirements 0501-42

Kano Model: Types of customer requirements

- Different versions, different terminology

1. The five–level version (original by Kano)
    1. Musts (expected)
    2. One-Dimensional (wants)
    3. Delighter (exciting)
    4. Don’t care
    5. Reverse (dissatisfiers)
2. A three-level version of needs*
    1. Expected (musts)
    2. Normal needs (wants)
    3. Exciting needs (delighters or wows)

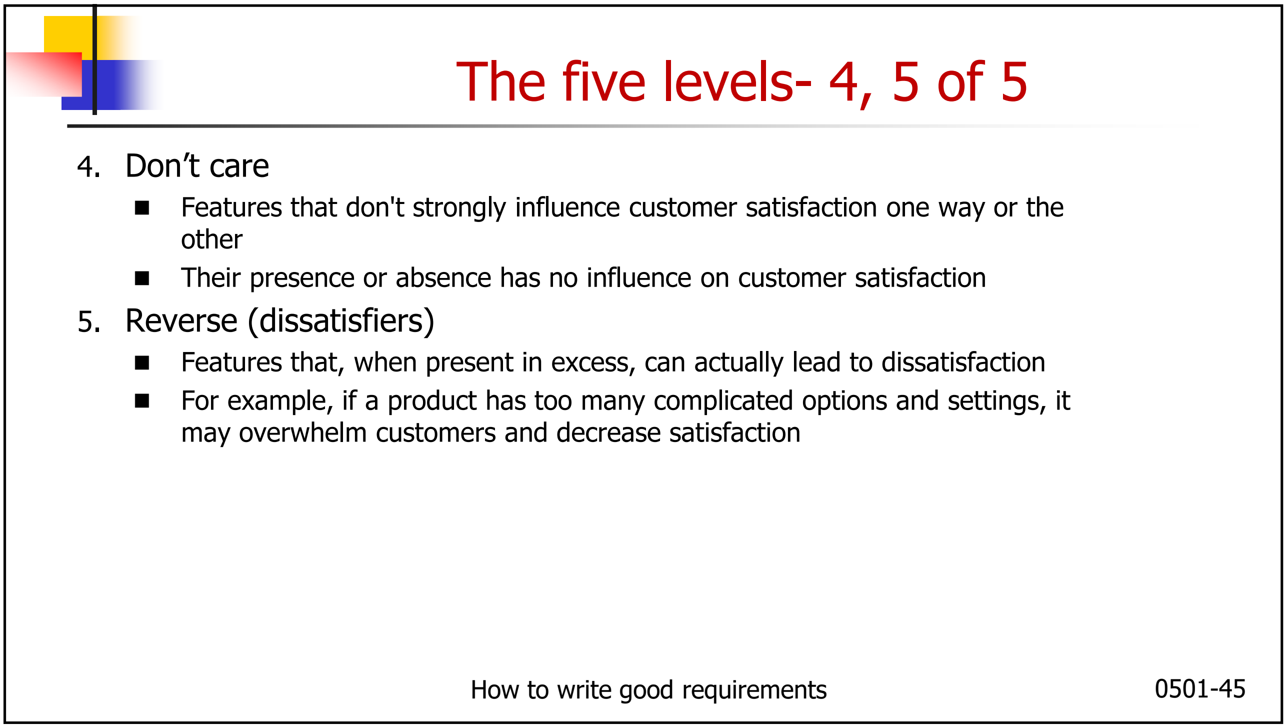


* https://asq.org/quality-resources/kano-model

Highlights show comparison between versions

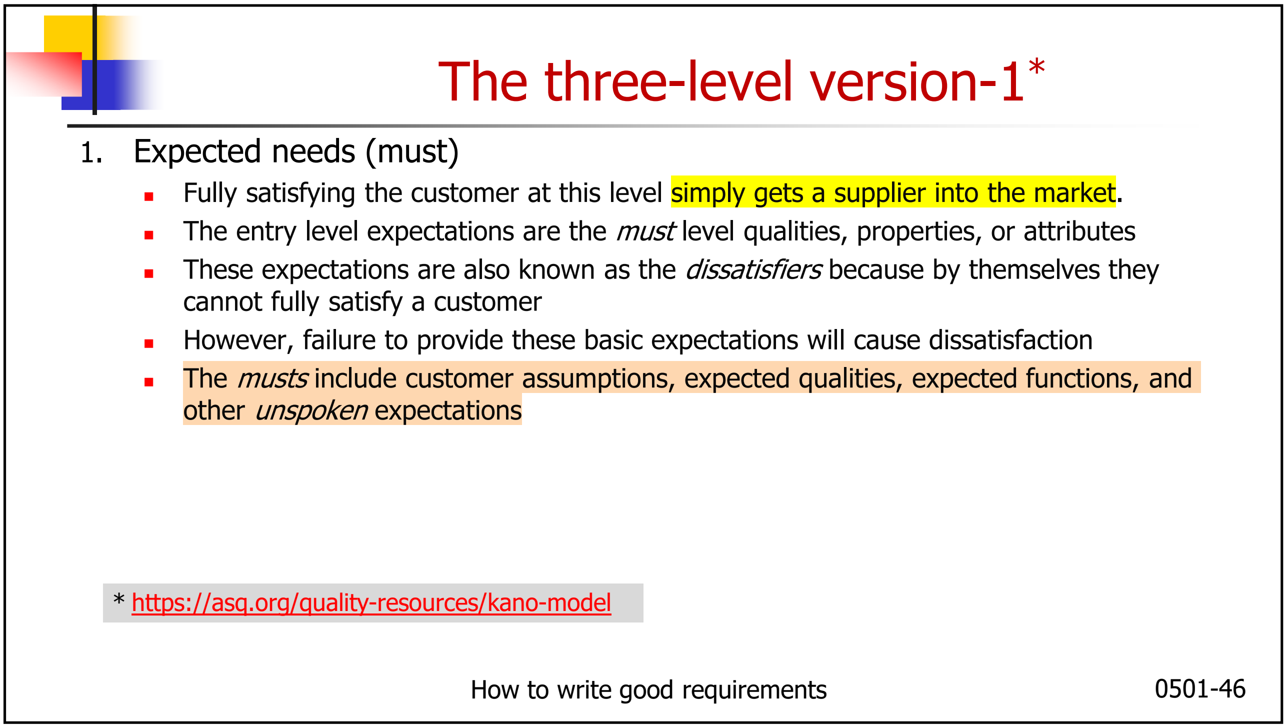
The five levels-1, 2, 3 of 5

1. Musts (Expected)
    - Basic requirements or features that customers expect as a minimum
    - Leads to dissatisfaction if they are missing, but their presence doesn't significantly increase customer satisfaction because they are considered basic and essential.
2. One-Dimensional (wants)
    - Features that linearly correlate with customer satisfaction.
    - Features that directly impact customer satisfaction in a predictable way
    - The more you improve these features, the more satisfied customers become
3. Delighter (Exciting)
    - Features that were not asked for but when present, significantly enhance customer satisfaction
    - Their absence does not necessarily lead to dissatisfaction
    - Customers may not even be aware they want these features until they experience them



The five levels- 4, 5 of 5

4. Don’t care
   - Features that don't strongly influence customer satisfaction one way or the other
   - Their presence or absence has no influence on customer satisfaction
5. Reverse (dissatisfiers)
   - Features that, when present in excess, can actually lead to dissatisfaction
   - For example, if a product has too many complicated options and settings, it may overwhelm customers and decrease satisfaction



The three-level version-1*

1. Expected needs (must)
   - Fully satisfying the customer at this level simply gets a supplier into the market.
   - The entry level expectations are the *must* level qualities, properties, or attributes
   - These expectations are also known as the *dissatisfiers* because by themselves they cannot fully satisfy a customer
   - However, failure to provide these basic expectations will cause dissatisfaction
   - The *musts* include customer assumptions, expected qualities, expected functions, and other *unspoken* expectations

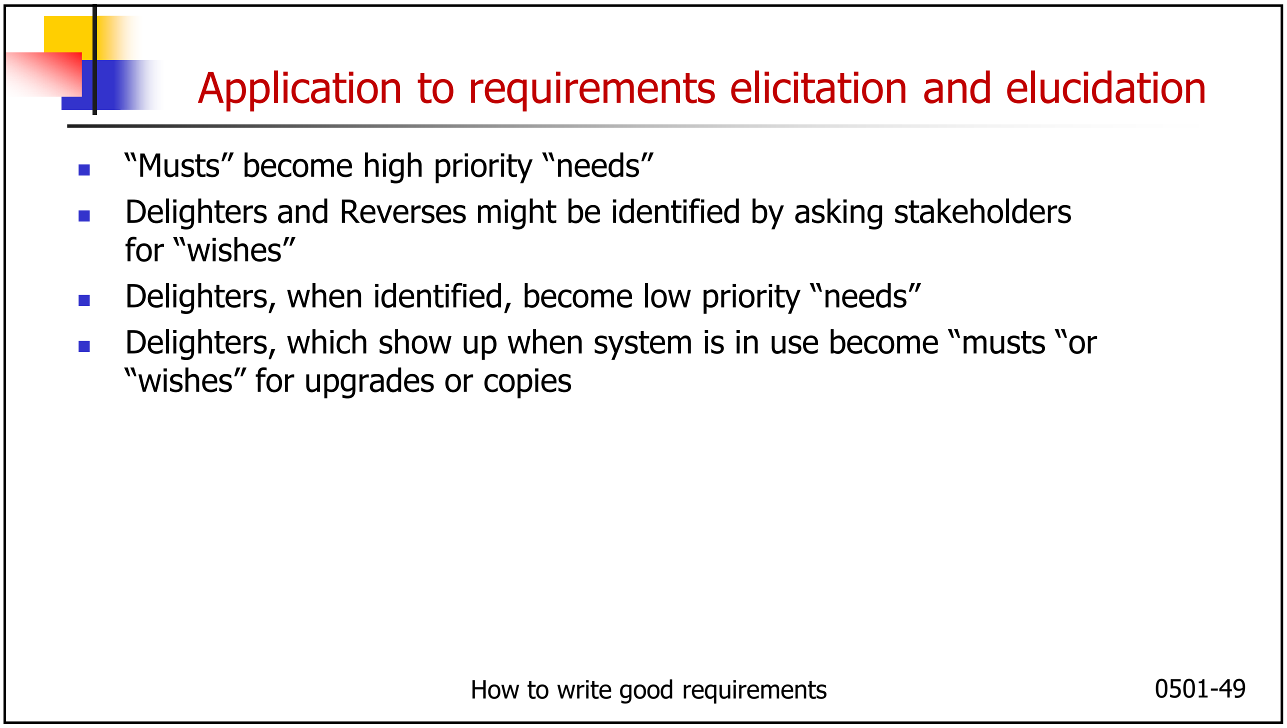
* https://asq.org/quality-resources/kano-model

The three-level version-2

2. Normal needs (wants)
   - the qualities, attributes, and characteristics that keep a supplier in the market
   - These next higher level expectations are known as the wants or the satisfiers because they are the ones that customers will specify
   - They can either satisfy or dissatisfy the customer depending on their presence or absence
   - The wants include voice of the customer requirements and other spoken expectations

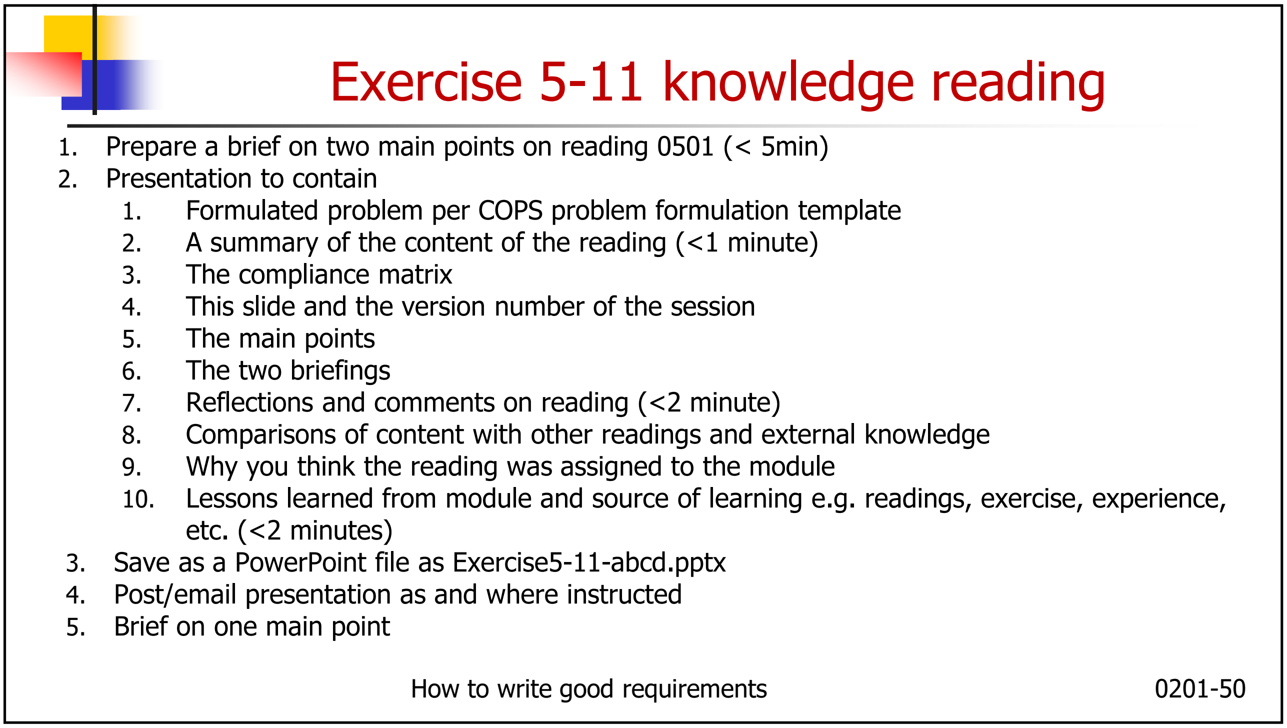
The three-level version-3

3. Exciting needs (delighters or wows)
   - These are features and properties that make a supplier a leader in the market
   - The highest level of customer expectations, as described by Kano, is termed the *wow* level qualities, properties, or attributes
   - These expectations are also known as the *delighters* or *exciters* because they go well beyond anything the customer might imagine and ask for
   - Their absence does nothing to hurt a possible sale, but their presence improves the likelihood of purchase
   - *Wows* not only excite customers to make on-the-spot purchases but make them return for future purchases
   - These are *unspoken* ways of delighting the customer
   - Over time, *wows* become *wants* become *musts*



Application to requirements elicitation and elucidation

- “Musts” become high priority “needs”
- Delighters and Reverses might be identified by asking stakeholders for “wishes”
- Delighters, when identified, become low priority “needs”
- Delighters, which show up when system is in use become “musts “or “wishes” for upgrades or copies



Exercise 5-11 knowledge reading

1. Prepare a brief on two main points on reading 0501 (< 5min)
2. Presentation to contain
    1. Formulated problem per COPS problem formulation template
    2. A summary of the content of the reading (<1 minute)
    3. The compliance matrix
    4. This slide and the version number of the session
    5. The main points
    6. The two briefings
    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 module and source of learning e.g. readings, exercise, experience, etc. (<2 minutes)
3. Save as a PowerPoint file as Exercise5-11-abcd.pptx
4. Post/email presentation as and where instructed
5. Brief on one main point

Any questions ?

1. Best
2. Worst
3. Missing

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