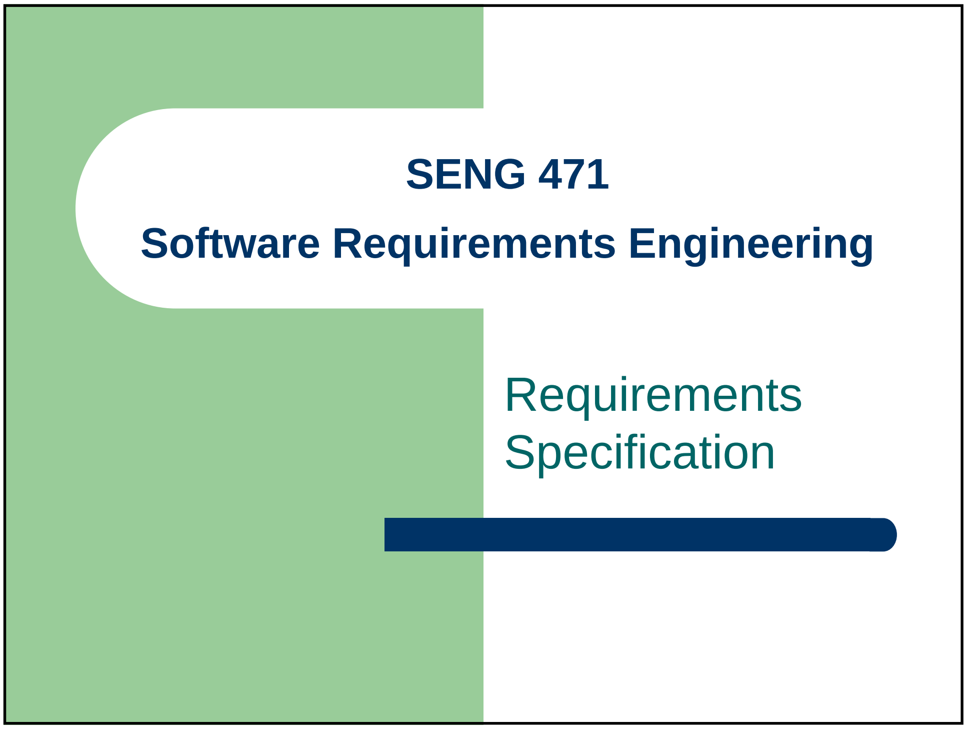
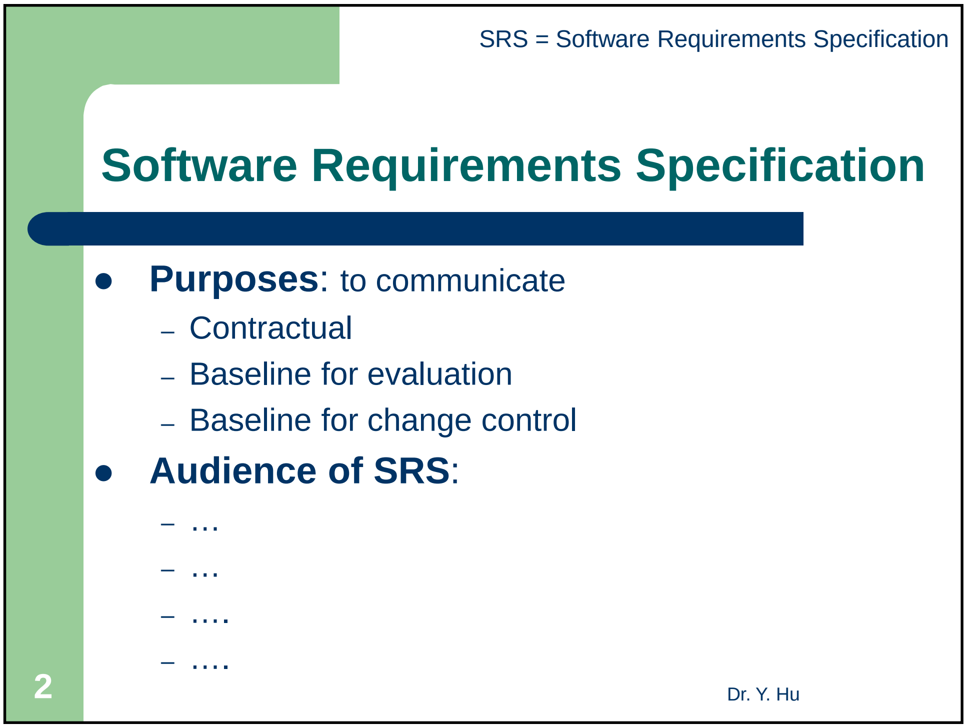


SENG 471

Software Requirements Engineering

Requirements Specification



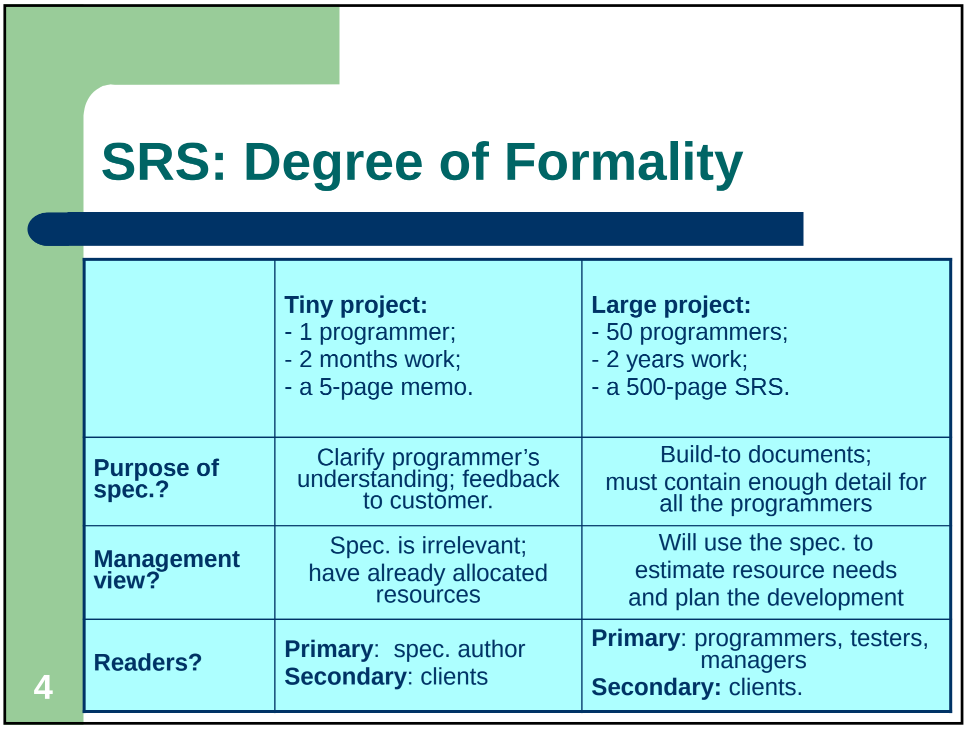
SRS = Software Requirements Specification

Software Requirements Specification

- **Purposes**: to communicate
  - Contractual
  - Baseline for evaluation
  - Baseline for change control
- **Audience of SRS**:
  - …
  - …
  - ….
  - ….

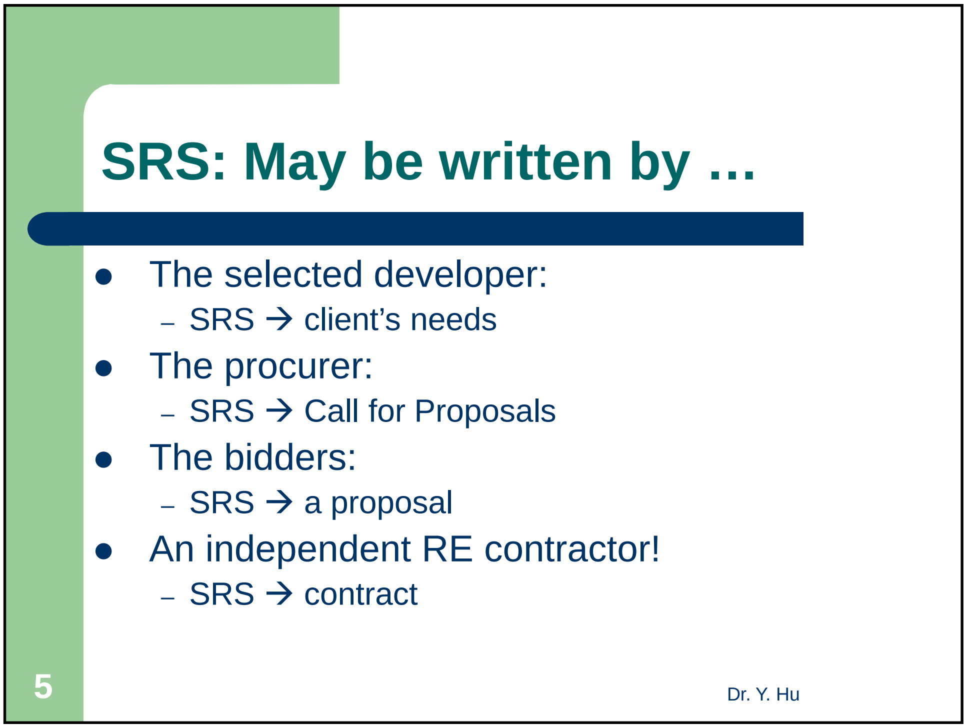
2

Dr. Y. Hu



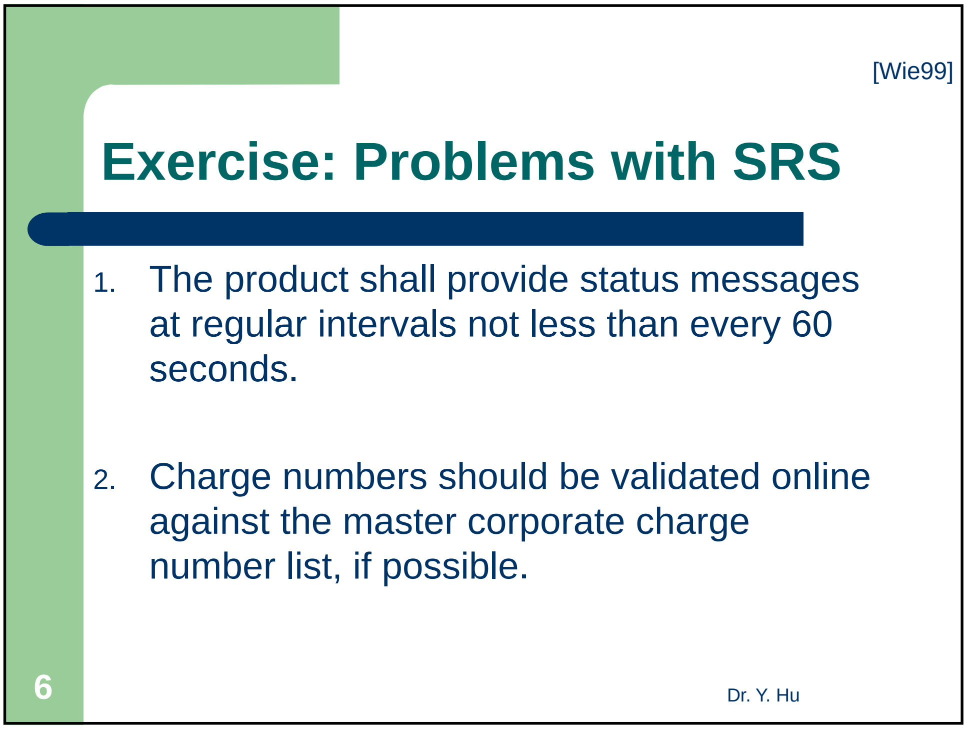
SRS: Degree of Formality

	Tiny project: - 1 programmer; - 2 months work; - a 5-page memo.	**Large project:** - 50 programmers; - 2 years work; - a 500-page SRS.
Purpose of spec.?	Clarify programmer's understanding; feedback to customer.	Build-to documents; must contain enough detail for all the programmers
Management view?	Spec. is irrelevant; have already allocated resources	Will use the spec. to estimate resource needs and plan the development
Readers?	**Primary**: spec. author **Secondary**: clients	**Primary**: programmers, testers, managers **Secondary:** clients.



SRS: May be written by …

- The selected developer:
  - SRS → client's needs
- The procurer:
  - SRS → Call for Proposals
- The bidders:
  - SRS → a proposal
- An independent RE contractor!
  - SRS → contract

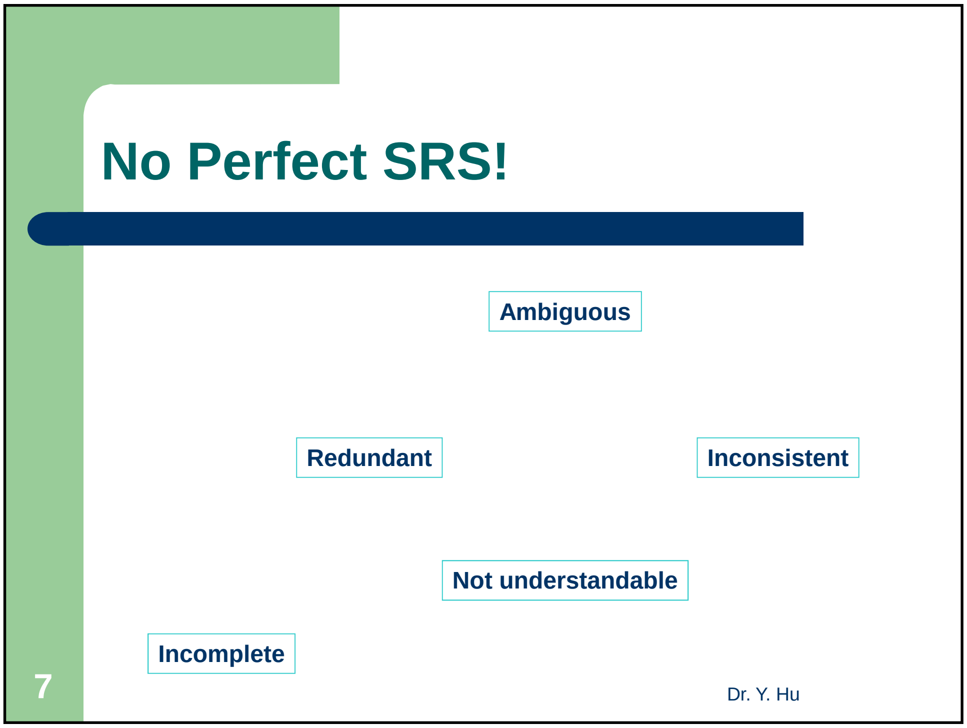


[Wie99]

Exercise: Problems with SRS

1. The product shall provide status messages at regular intervals not less than every 60 seconds.

2. Charge numbers should be validated online against the master corporate charge number list, if possible.



No Perfect SRS!

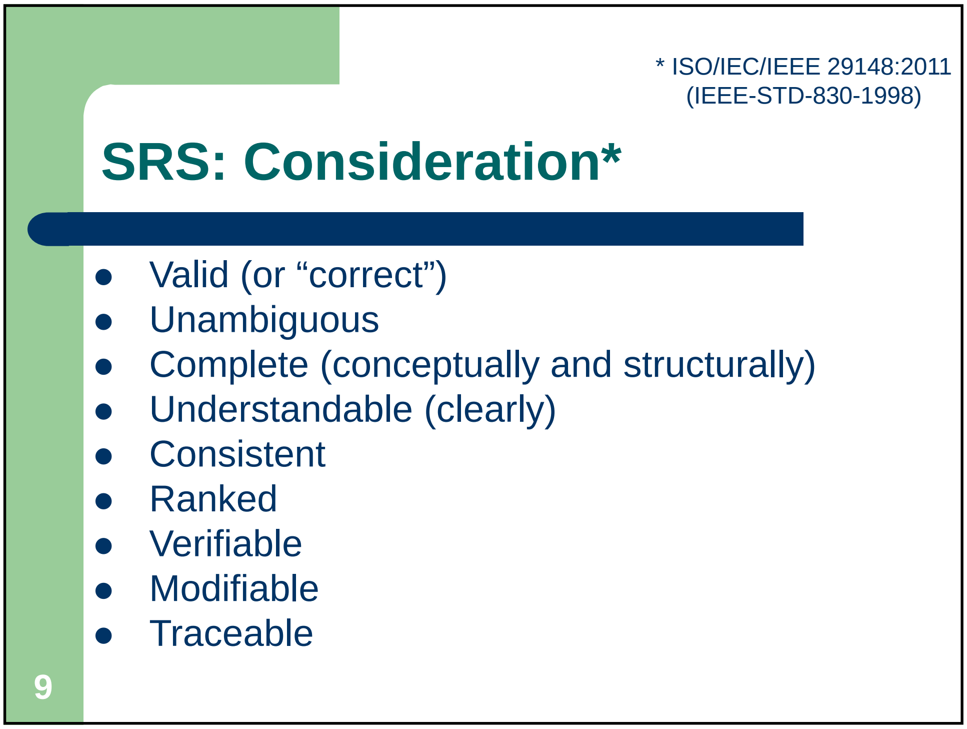
Ambiguous

Redundant

Inconsistent

Not understandable

Incomplete

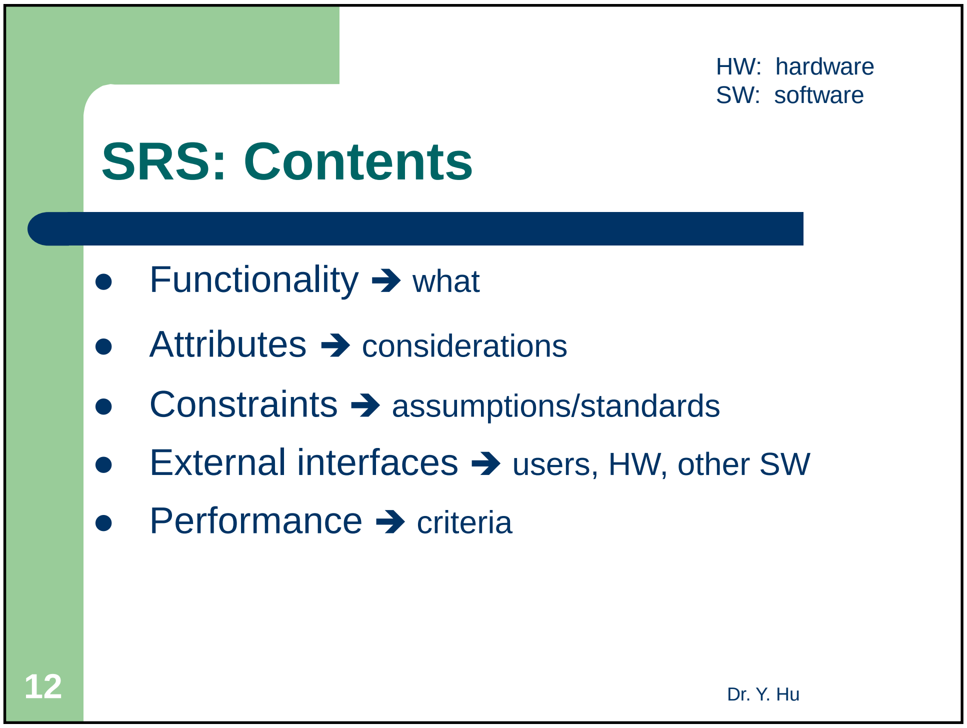


* ISO/IEC/IEEE 29148:2011
(IEEE-STD-830-1998)

SRS: Consideration*

- Valid (or “correct”)
- Unambiguous
- Complete (conceptually and structurally)
- Understandable (clearly)
- Consistent
- Ranked
- Verifiable
- Modifiable
- Traceable

9



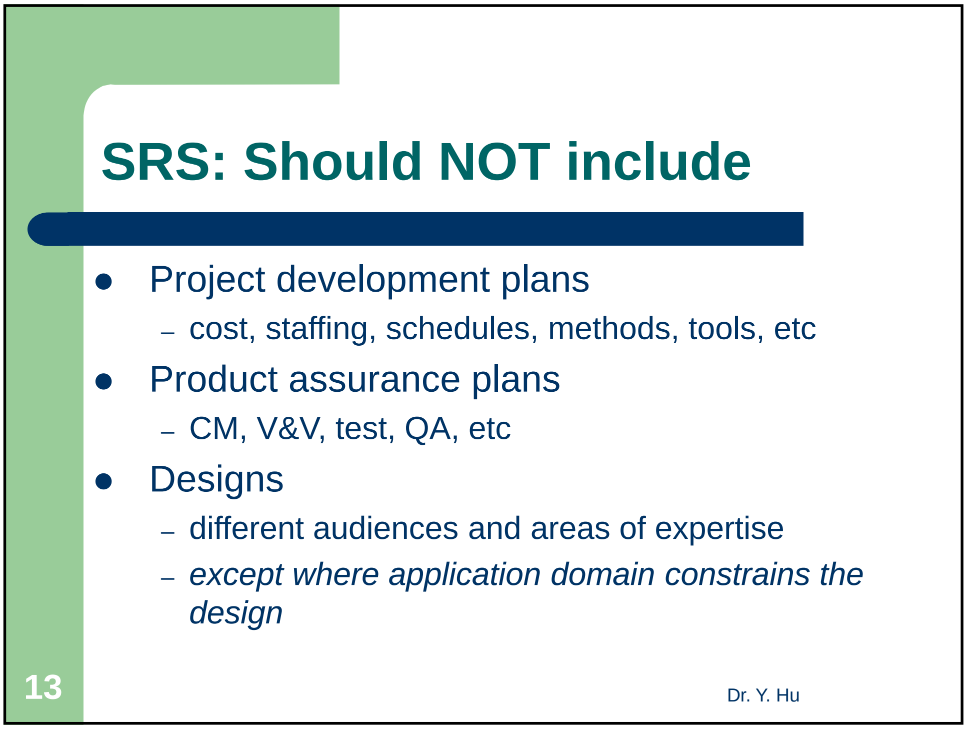
HW: hardware
SW: software

SRS: Contents

- Functionality ➔ what
- Attributes ➔ considerations
- Constraints ➔ assumptions/standards
- External interfaces ➔ users, HW, other SW
- Performance ➔ criteria

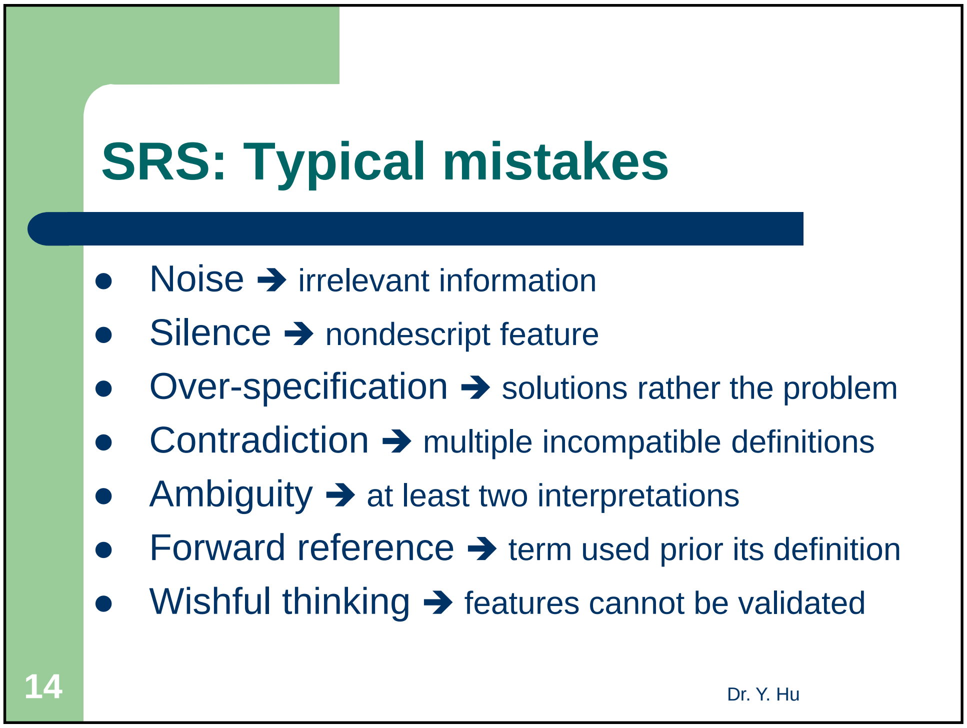
12

Dr. Y. Hu



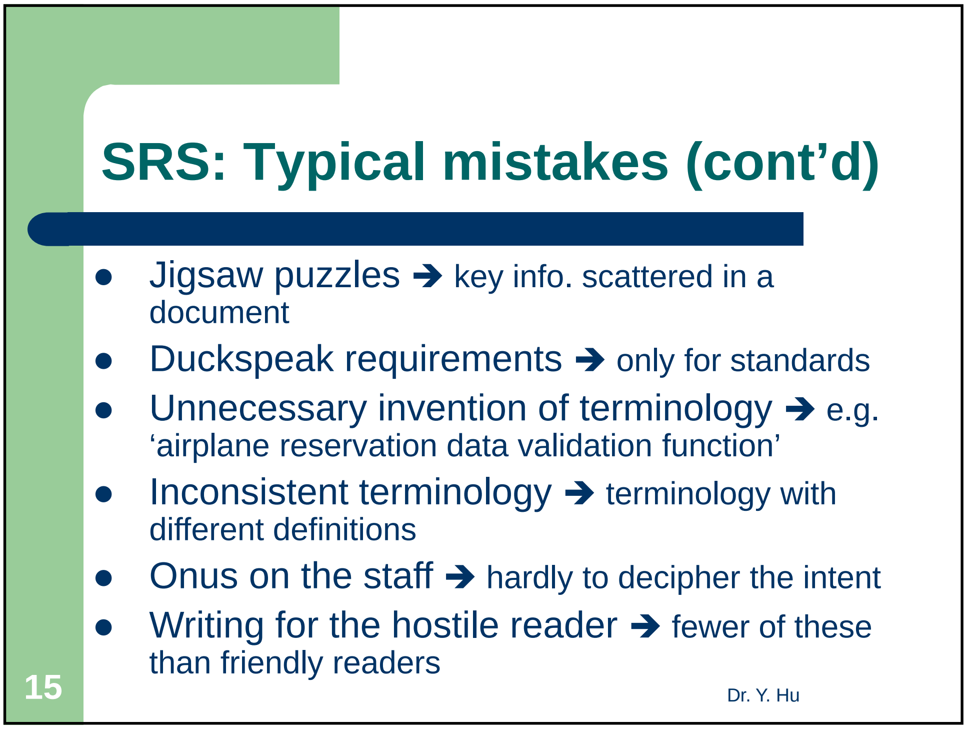
SRS: Should NOT include

- Project development plans
  - cost, staffing, schedules, methods, tools, etc
- Product assurance plans
  - CM, V&V, test, QA, etc
- Designs
  - different audiences and areas of expertise
  - *except where application domain constrains the design*



SRS: Typical mistakes

- Noise ➔ irrelevant information
- Silence ➔ nondescript feature
- Over-specification ➔ solutions rather the problem
- Contradiction ➔ multiple incompatible definitions
- Ambiguity ➔ at least two interpretations
- Forward reference ➔ term used prior its definition
- Wishful thinking ➔ features cannot be validated

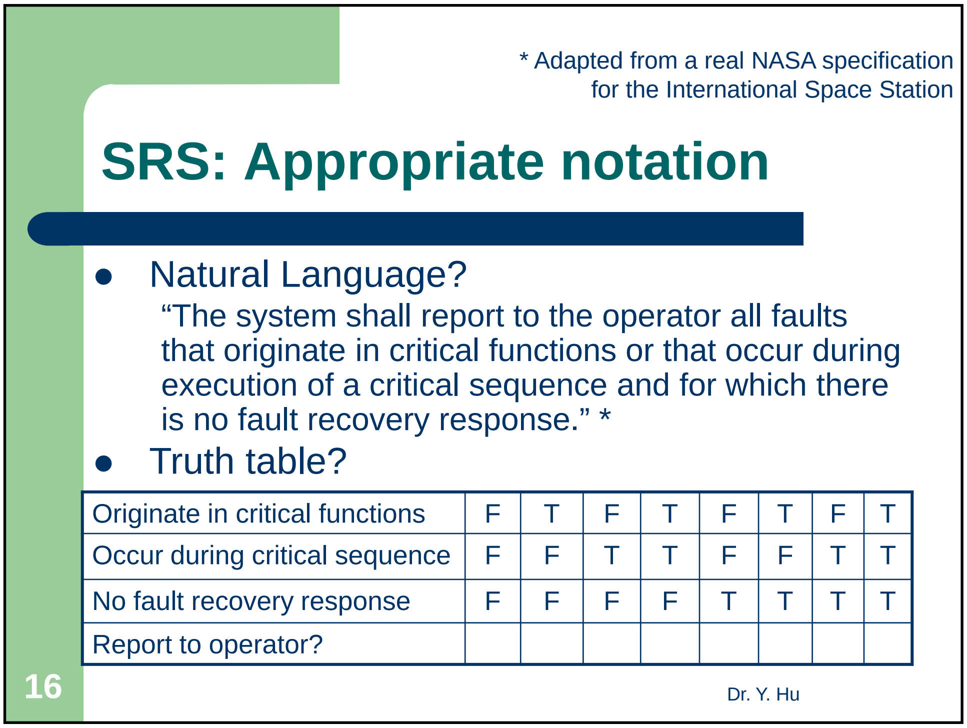


SRS: Typical mistakes (cont'd)

- Jigsaw puzzles ➔ key info. scattered in a document
- Duckspeak requirements ➔ only for standards
- Unnecessary invention of terminology ➔ e.g. 'airplane reservation data validation function'
- Inconsistent terminology ➔ terminology with different definitions
- Onus on the staff ➔ hardly to decipher the intent
- Writing for the hostile reader ➔ fewer of these than friendly readers

15

Dr. Y. Hu



* Adapted from a real NASA specification for the International Space Station

SRS: Appropriate notation

- Natural Language?

  "The system shall report to the operator all faults that originate in critical functions or that occur during execution of a critical sequence and for which there is no fault recovery response." *

- Truth table?

Originate in critical functions	F	T	F	T	F	T	F	T
Occur during critical sequence	F	F	T	T	F	F	T	T
No fault recovery response	F	F	F	F	T	T	T	T
Report to operator?								

16

Dr. Y. Hu

SRS: Organizing requirements

- Example ways of organizing the document
  - External stimulus or external situation
  - System feature
  - System response
  - External object
  - User type
  - Mode
  - Subsystem

SRS Example Outline

1. Introduction
   - 1.1 Purpose
   - 1.2 Scope
   - 1.3 Product overview
     - 1.3.1 Product perspective
     - 1.3.2 Product functions
     - 1.3.3 User characteristics
     - 1.3.4 Limitations
   - 1.4 Definitions
2. References
3. Specific requirements
4. Verification

   (parallel to subsections in Section 3)
5. Appendices
   - 5.1 Assumptions and dependencies
   - 5.2 Acronyms and abbreviations

ISO/IEC/IEEE 29148:2011

SRS: STD Section 3

3.1 External Interface Requirements
- 3.1.1 User Interfaces
- 3.1.2 Hardware Interfaces
- 3.1.3 Software Interfaces
- 3.1.4 Communication Interfaces

3.2 Functions

[this section organized by mode, user class, feature, etc. For example:]
- 3.2.1 Mode 1
  - 3.2.1.1 Functional Requirement 1.1
  - ………
- 3.2.2 Mode 2
  - 3.2.2.1 Functional Requirement 2.1
  - ………
- ………
- 3.2.n Mode n
  - ………

3.3 Usability requirements

[state this in measureable terms!]

3.4 Performance requirements

[state this in measureable terms!]

3.5 Logical database requirements

[state this in measureable terms!]

3.6 Design Constraints
- 3.6.1 Standards Compliance
- 3.6.2 Hardware Limitations
- … …

3.7 Software system attributes
- 3.7.1 Reliability
- 3.7.2 Availability
- 3.7.3 Security
- 3.7.4 Maintainability
- 3.7.5 Portability

3.8 Supporting information

[Other requirements]

21

Recap

- Software Requirements Specification
  - Purpose
  - Contents
  - Typical mistakes
  - IEEE standard (reading)

23

Dr. Y. Hu