U O W

Software Requirements, Specifications and Formal Methods

Dr. Shixun Huang

UNIVERSITY OF WOLLONGONG AUSTRALIA

Writing the Requirements Document

Requirements Agreement and Analysis

- Throughout the elicitation activity and especially before finalizing the systems requirements specification, requirements should be analyzed and reconciled.
- Requirements analysis is the activity of analyzing requirements for problems.
- Problematic requirements include those that are:
  - Confusing
  - Extraneous
  - Duplicated
  - Conflicting
  - Missing

Requirements Agreement and Analysis

- Requirements analysis is usually informal/semiformal
  - Data Flow Diagram
  - Finite State Machine
  - Petri Net
  - Coloured Petri Net

- Requirements agreement is the process of reconciling differences in the same requirement derived from different sources.
  - Is the requirement complete?
  - Is the requirement clear?
  - Is the requirement implementable?
  - Is the qualification plan clear and acceptable?

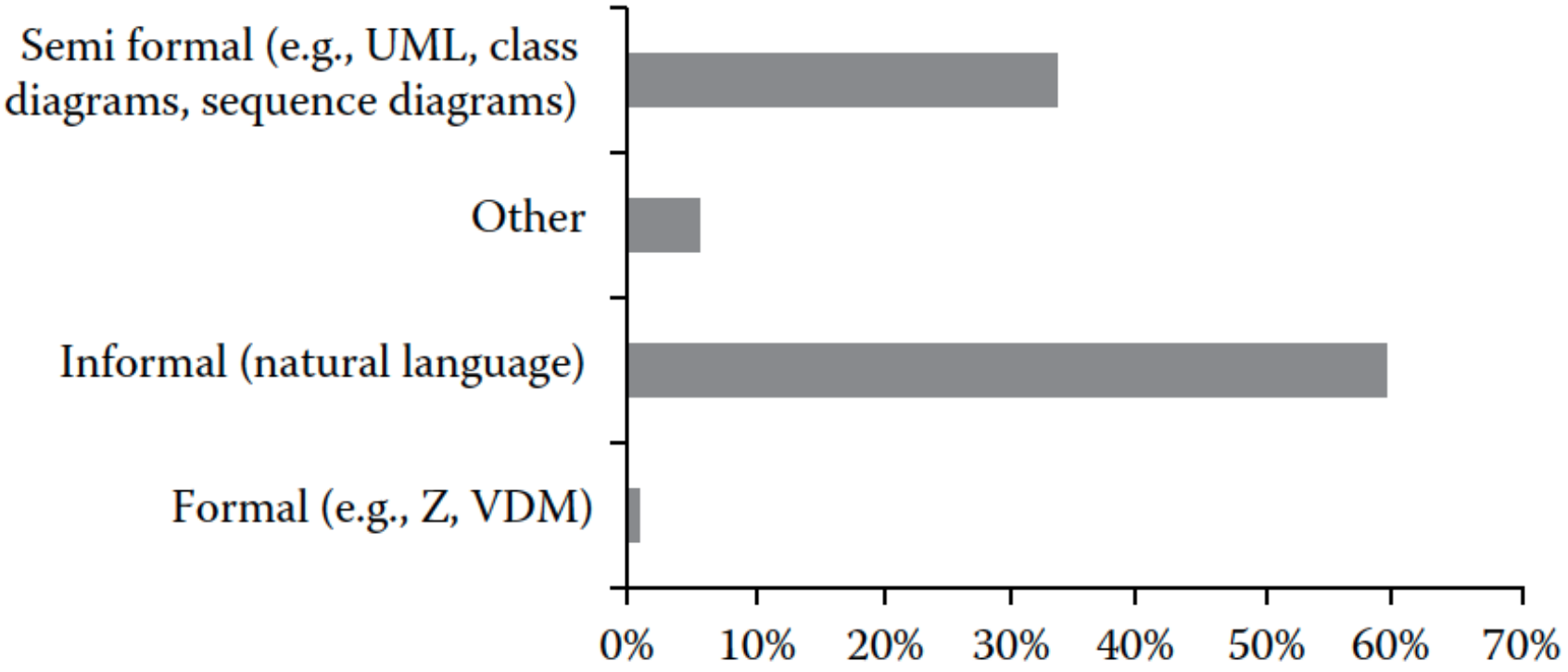
Requirements Representation

- Various techniques can be used to describe functionality in any system, including use cases, user stories, natural languages, formal languages, stylized (restricted natural) languages, and a variety of formal, semiformal, and informal diagrams.
- How to select the right representation style for a project?
  - Technical issues such as the maturity and effectiveness of the technique being considered
  - Complexity of the techniques
  - Social and cultural issues such as organization opposition to change
  - The level of education and knowledge of developers

Requirements Representation

- Generally, there are three forms of requirements representation: formal, informal, and semiformal.
- Formal representations have a rigorous mathematical basis
  - Z, VDM
- Informal techniques include natural language (i.e., human languages), flowcharts, ad hoc diagrams.
- Semiformal representation techniques include those that, while appearing informal, have at least a partial formal basis
  - UML, SysML

Requirements Representation



Requirement specification notation prevalence

ISO/IEC/IEEE Standard 29148

- IEEE 29148 is based on a model that produces a document that helps:
  - To establish the basis for agreement between the acquirers or suppliers on what the product is to do
  - To force a rigorous assessment of requirements before design can begin and reduce later redesign
  - To provide a realistic basis for estimating product costs and risks, and schedules organizations to develop validation and verification plans
  - To provide an informed basis for deploying a product to new users or new operational environments
  - To provide a basis for product enhancement (IEEE 29148 2011)

ISO/IEC/IEEE Standard 29148

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
3.1 External interfaces
3.2 Functions
3.3 Usability requirements
3.4 Performance requirements
3.5 Logical database requirements
3.6 Design constraints
3.7 Software system attributes
3.8 Supporting information
4. Verification
(parallel to subsections in Section 3)
5. Appendices
5.1 Assumptions and dependencies
5.2 Acronyms and abbreviations

Table of contents for an SRS document as recommended by IEEE Standard 29148

Recommendations on Representing NFRs

- External interfaces
- Usability
- Performance
- Logical database considerations
- Design constraints
- Standards compliance
- Software system attribute requirements

- External interface
  - Item name, purpose description,
  - input/output,
  - Value range, accuracy
  - Units of measure, timing
  - Screen formats/organization
  - Data/command formats

Recommendations on Representing NFRs

- Usability requirements provide those requirements that meet the user needs.
- In terms of customer satisfaction, usability is often the most important class of requirements.
- These kinds of requirements are hard to discover in novel systems or with regard to new features.
- Prototyping can be helpful in discovering such requirements.

Recommendations on Representing NFRs

- Performance requirements are static and dynamic requirements placed on the software or on human interaction with the software as a whole.

- Typical performance requirements might include the number of simultaneous users to be supported, the numbers of transactions and tasks, and the amount of data to be processed within certain time periods for both normal and peak workload conditions.

Recommendations on Representing NFRs

- Logical database requirements are types of information used by various functions such as:
  - Frequency of use
  - Accessing capabilities
  - Data entities and their relationships
  - Integrity constraints
  - Data retention requirements

Recommendations on Representing FRs

- The functional requirements should capture all system inputs and the exact sequence of operations and responses (outputs) to normal and abnormal situations for every input possibility.
- Functional requirements may use case-by-case description or other general forms of description.
- Functional requirements can be organized by:
  - System mode (e.g., navigation, combat, diagnostic)
  - User class (e.g., user, supervisor, diagnostic)
  - Object (by defining classes/objects, attributes, functions/methods, and messages)
  - Feature (describes what the system provides to the user)
  - Stimulus (e.g., sensor 1, sensor 2, actuator 1, …)
  - Functional hierarchy (e.g., using a top-down decomposition of tasks)

Recommendations on Representing FRs

- Operating system functionality

001 The operating system shall provide a common set of mechanisms necessary to support real-time systems such as multitasking support, CPU scheduling, basic communication, and memory management.

- 001.1 The operating system shall provide multitasking capabilities.
- 001.2 The operating system shall provide event-driven, priority-based scheduling.
  - 001.2.1 Task execution shall be controlled by a task's priority and the availability of resources required for its execution.
  - 001.3 The operating system shall provide support for intertask communication and synchronization.
  - 001.4 The operating system shall provide real-time clock support.
  - 001.5 The operating system software shall provide task-level context switching for the 80387 math coprocessor.

Recommendations on Representing FRs

- Command Validation

211 The flight software shall perform CCSDS command structure validation.

- *211.1 The flight software shall implement CCSDS Command Operations Procedure number 1 (COP-1) to validate that CCSDS transfer frames were received correctly and in order.*
- *211.2 The flight software shall support a fixed-size frame acceptance and reporting mechanism (FARM) sliding window of 127 and a fixed FARM negative edge of 63.*
- *211.3 The flight software shall telemeter status and discard the real-time command packet if any of the following errors occur:*
  - *Checksum fails validation prior to being issued*
  - *An invalid length is detected*
  - *An invalid application ID is detected*
- *211.4 The flight software shall generate and maintain a command link control word (CLCW). Each time an update is made to the CLCW, a CLCW packet is formatted and routed for possible downlink (NASA WIRE 1996).*

Users of a Requirements Document

There are several users of the requirements document, each with a unique perspective, needs, and concerns.

Typical users of the document include:

- Customers
- Managers
- Developers
- Test engineers
- Maintenance engineers
- Stakeholders

SRS Requirements

- IEEE 29148 template, but the "right" format depends on what the sponsor, customer, employer, and other stakeholders require.
- In terms of general organization, writing approach, and discourse, best practices include:
  - Using consistent modeling approaches and techniques throughout the specification, for example, a top-down decomposition, structured, or object-oriented approaches
  - Separating operational specification from descriptive behavior
  - Using consistent levels of abstraction within models and conformance between levels of refinement across models
  - Modeling nonfunctional requirements as a part of the specification models—in particular timing properties
  - Omitting hardware and software assignments in the specification (another aspect of design rather than specification)

Preferred Writing Style

- A well-organized SRS document should have a consistent level of detail
- A simplified standard requirement form is:
  - The [noun phrase] shall (not) [verb phrase]
  - Where [noun phrase] describes the main subject of the requirement, the shall (or shall not) statement indicates a required or prohibited behavior and [verb phrase] indicates the actions of the requirement.
- Consider these two requirements for the baggage handling system:
  - The system shall reject untagged baggage
  - The system shall not reject damaged baggage
- Use of imperatives: shall, should, must, will, responsible for, and etc.

Preferred Writing Style

Imperative	*Most Common Use*
Are applicable	To include, by reference, standards, or other documentation as an addition to the requirements being specified
Is required to	As an imperative in specifications statements written in the passive voice
Must	To establish performance requirements or constraints
Responsible for	In requirements documents that are written for systems whose architectures are predefined
Shall	To dictate the provision of a functional capability
Should	Not frequently used as an imperative in requirement specification statements
Will	To cite things that the operational or development environment are to provide to the capability being specified

Preferred Writing Style

Best Practices and Recommendations

- Writing effective requirements specifications can be very difficult even for trivial systems because of the many challenges that we have noted.
- Some of the more common dangers in writing poor SRS documents include:
  - Mixing of operational and descriptive specifications
  - Combining low-level hardware functionality and high-level systems and software functionality in the same functional level
  - Omission of timing information

Tools

- Microsoft Words
- Freemind
- IBM Rational Doors (https://www.ibm.com/products/requirements-management-doors-next?r=rmt)