Document important information elicited or developed when performing a core the RE activity
• *i.e.*, documentation, elicitation, negotiation, validation and/or management
"Requirements Lifecycle"

Source: Adapted from Pohl, CAISE 1993
Requirements Specification

• What is Requirements Specification?
Requirements Specification

→ How do we communicate the Requirements to others?
  ✧ It is common practice to capture them in a specification
  ➢ But an specification does not need to be a single paper document...

→ Purpose
  ✧ Communication
    ➢ explains the application domain and the system to be developed
  ✧ Contractual
    ➢ May be legally binding!
    ➢ Expresses agreement and a commitment
  ✧ Baseline for evaluating the software
    ➢ supports testing, V&V
    ➢ “enough information to verify whether delivered system meets requirements”
  ✧ Baseline for change control

→ Audience
  ✧ Customers & Users
    ➢ interested in system requirements…
    ➢ …but not detailed software requirements
  ✧ Systems (Requirements) Analysts
    ➢ Write other specifications that inter-relate
  ✧ Developers, Programmers
    ➢ Have to implement the requirements
  ✧ Testers
    ➢ Have to check that the requirements have been met
  ✧ Project Managers
    ➢ Have to measure and control the project
Appropriate Specification

A) Tiny project, 1 programmer, 2 months work
programmer talks to customer, then writes up a 2-page memo

<table>
<thead>
<tr>
<th>Purpose of spec?</th>
<th>Project A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Crystalizes programmer’s understanding; feedback to customer</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Management view?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spec is irrelevant; have already allocated resources</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Readers?</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Primary</strong>: Spec author; <strong>Secondary</strong>: Customer</td>
</tr>
</tbody>
</table>
Appropriate Specification

B) Large project, 50 programmers, 2 years work

- team of analysts model the requirements, then document them in a 500-page document

<table>
<thead>
<tr>
<th>Purpose of spec?</th>
<th>Project B</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Build-to document; must contain enough detail for all the programmers</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Management view?</th>
<th>Will use the spec to estimate resource needs and plan the development</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Primary</strong>: programmers, testers, managers; <strong>Secondary</strong>: customers</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Readers?</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- programmer talks to customer, then writes up a 2-page memo
Procurement

→ A requirements specification may be written by...

- **the procurer:**
  - specification is really a call for proposals
  - Must be general enough to yield a good selection of bids...
  - …and specific enough to exclude unreasonable bids

- **the bidders:**
  - specification is a proposal to implement a system to meet the CfP
  - must be specific enough to demonstrate feasibility and technical competence
  - …and general enough to avoid over-commitment

- **the selected developer:**
  - reflects the developer’s understanding of the customer’s needs
  - forms the basis for evaluation of contractual performance

- **...or by an independent RE contractor!**

→ Choice over what point to compete the contract

- **Early (conceptual stage)**
  - can only evaluate bids on apparent competence & ability

- **Late (detailed specification stage)**
  - more work for procurer; appropriate RE expertise may not be available in-house
Specification Contents

Specification should address:

- **Functionality**
  - What is the software supposed to do?

- **External interfaces**
  - How does the software interact with people, the system's hardware, other hardware, and other software?
  - What assumptions can be made about these external entities?

- **Performance**
  - What is the speed, availability, response time, recovery time of various software functions, and so on?

- **Attributes**
  - What are the portability, correctness, maintainability, security, and other considerations?

- **Design constraints imposed on an implementation**
  - Are there any required standards in effect, implementation language, policies for database integrity, resource limits, operating environment(s) and so on?
Specification should not include…

→ **Project development plans**
  - E.g. cost, staffing, schedules, methods, tools, etc
  - Lifetime of SRS is until the software is made obsolete
  - Lifetime of development plans is much shorter

→ **Product assurance plans**
  - V&V, test, QA, etc
  - Different audiences
  - Different lifetimes

→ **Designs**
  - Requirements and designs have different audiences
  - Analysis and design are different areas of expertise
    - I.e. requirements analysts shouldn’t do design!
# 1 Introduction
- **Purpose**
- **Scope**
- Definitions, acronyms, abbreviations
- Reference documents
- Overview

# 2 Overall Description
- Product perspective
- Product functions
- User characteristics
- Constraints
- Assumptions and Dependencies

# 3 Specific Requirements

Appendices

Index

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Introduction</td>
<td>Identifies the product, &amp; application domain</td>
</tr>
<tr>
<td></td>
<td>Describes contents and structure of the remainder of the SRS</td>
</tr>
<tr>
<td></td>
<td>Describes all external interfaces: system, user, hardware, software; also operations and site adaptation, and hardware constraints</td>
</tr>
<tr>
<td></td>
<td>Summary of major functions, e.g. use cases</td>
</tr>
<tr>
<td></td>
<td>Anything that will limit the developer’s options (e.g. regulations, reliability, criticality, hardware limitations, parallelism, etc)</td>
</tr>
<tr>
<td></td>
<td>All the requirements go in here (i.e. this is the body of the document). IEEE STD provides 8 different templates for this section</td>
</tr>
</tbody>
</table>
IEEE STD Section 3 (example)

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 Functional Requirements
   *this section organised 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
      3.2.1.1 Functional Requirement 1.1
         ...
   3.2.2 Mode n
      ...

3.3 Performance Requirements
   *Remember to state this in measurable terms!*

3.4 Design Constraints
   3.4.1 Standards compliance
   3.4.2 Hardware limitations
      etc.

3.5 Software System Attributes
   3.5.1 Reliability
   3.5.2 Availability
   3.5.3 Security
   3.5.4 Maintainability
   3.5.5 Portability

3.6 Other Requirements
Organizing the Requirements

→ Example Structures - organize by…

- **External stimulus or external situation**
  - e.g., for an aircraft landing system, each different type of landing situation: wind gusts, no fuel, short runway, etc

- **System feature**
  - e.g., for a telephone system: call forwarding, call blocking, conference call, etc

- **System response**
  - e.g., for a payroll system: generate pay-cheques, report costs, print tax info;

- **External object**
  - e.g. for a library information system, organize by book type

- **User type**
  - e.g. for a project support system: manager, technical staff, administrator, etc.

- **Mode**
  - e.g. for word processor: page layout mode, outline mode, text editing mode, etc

- **Subsystem**
  - e.g. for spacecraft: command&control, data handling, comms, instruments, etc.

→ Requirements documents templates

https://www.volere.org/templates/volare-requirements-specification-template/
Documenting requirements artefacts

- Identifier
- Description
- Requirements artefact
  - Goal
  - Solution-oriented requirement
  - Scenario
    - contributes to realisation of
    - derived from
  - example of satisfaction
  - has
    - Criticality
    - Priority
    - Risk
    - {complete, disjoint}

1..* has

1

1..*
# Requirement Shell

Volere template, 2010

<table>
<thead>
<tr>
<th>Requirement # : Unique id</th>
<th>Requirement Type</th>
<th>Event/use case # :</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description : <em>A one sentence statement of the intention of the requirement</em></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rationale : <em>A justification of the requirement</em></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Originator : <em>Who raised this requirement?</em></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fit Criterion : <em>A measurement of the requirement such that it is possible to test if the solution matches the original requirement</em></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Customer Satisfaction :</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Priority : <em>The relative urgency of this requirement</em></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Supporting Materials : <em>Pointer to documents that illustrate and explain this requirement</em></td>
<td></td>
<td></td>
</tr>
<tr>
<td>History : <em>Creation, changes, deletions, etc.</em></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Degree of stakeholder happiness if this requirement is successfully implemented.**

Scale from 1 = uninterested to 5 = extremely pleased.

**Measure of stakeholder unhappiness if this requirement is not part of the final product.**

Scale from 1 = hardly matters to 5 = extremely displeased.

**The type from the template**

**List of events/use cases that need this requirement**
Things to Take Home

- Requirements specification
- Criteria for good requirements