

# Requirements Analysis that Works!

**Robert Halligan, FIE Aust**  
**Managing Director, Project Performance International**  
Email: rhalligan@ppi-int.com

## Introduction:

Innumerable studies have concluded that requirements problems are the single biggest contributor to cost overruns, schedule slippages and loss of capability in systems and software projects. Cost impacts alone of 10%, 20%, 50%, 80% and more are regularly reported by practitioners and researchers.

And yet, the cost of making substantial improvements in requirements quality is considerably lower than these cost impacts, typically 0.1 – 2% of total development cost - if appropriate skills and methods are applied.

Requirements analysis (the capture and validation of requirements through analysis of the problem domain) provides the tools for transforming the inadequate to the adequate, requirements-wise.

## Objectives of Requirements Analysis:

The usual criterion for adequacy of a set of requirements is that, if the requirements set is satisfied, the level of risk associated with failing to satisfy the needs of relevant stakeholders is low – typically an expected loss of value of two or three percent, or less.

To this basic criterion can be added the dimension of time. Requirements change with time due to the problem space genuinely changing, and due to “what is possible in technology” triggering perfectly valid new requirements. So, requirements analysis must be an ongoing activity, to a lesser or greater degree.

## Types of Requirements Defects:

Potential areas of defect in requirements, individually and/or as a set, are:

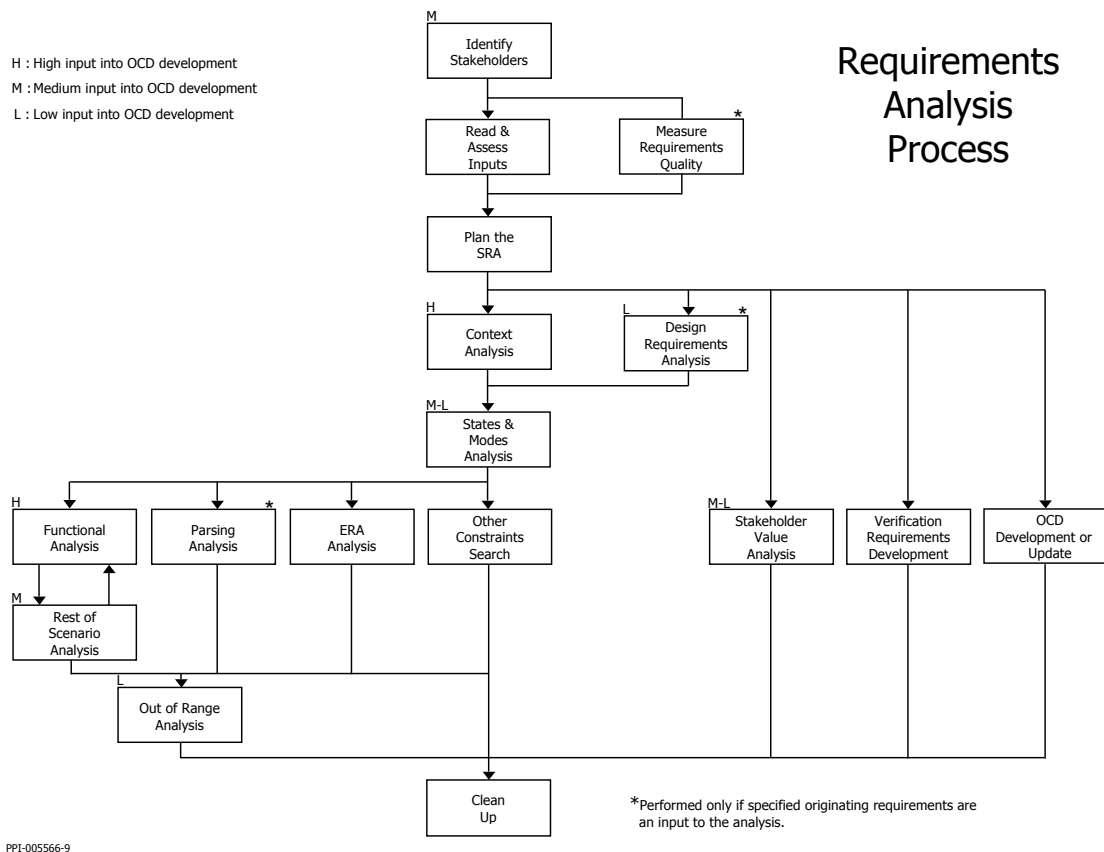
- a. correctness - refers to an absence of errors of fact in the specified requirement;
- b. completeness - for a requirement individually, refers to the inclusion of all necessary information such that if the requirement is satisfied, the need will also be satisfied;  
completeness - for requirements as a set, refers to inclusion of sufficient requirements such that if the set is satisfied, the need will also be satisfied with only a small expected loss due to omissions;
- c. consistency - requires that a requirement not be in conflict with any other requirement, nor be inconsistent internally;
- d. clarity - requires that a requirement be readily understandable without semantic analysis;
- e. non-ambiguity - requires that there be only one semantic interpretation of a requirement;
- f. connectivity - refers to a property whereby all of the terms within a requirement are adequately linked to other requirements and to word and term definitions, causing each individual requirement to properly relate to each other requirement in a set. Connectivity problems typically show up as a result of inconsistent references to the same thing, logical non-sequitur, and incorrect cross-references;
- g. singularity - refers to a property whereby a requirement cannot sensibly be expressed as two or more requirements having different actors (subjects), actions (verbs) and/or objects of action;
- h. verifiability - refers to the existence of a finite and objective process with which to provide adequate evidence that a requirement has been satisfied in a solution;
- i. modifiability - requires that:
  - (i) necessary changes to a requirement can be made completely and consistently; and
  - (ii) the same requirement is specified only once;
- j. feasibility – for a requirement, requires that some means exist whereby the requirement may be satisfied;  
feasibility – for a set of requirements, requires that some means exist whereby the requirements may be satisfied as a set;
- k. balance – for a set of requirements, refers to the set being optimum, i.e., forming a part of an optimum solution to the higher (physical) level problem for which the item which is the subject of the requirements is a part of the solution;

- l. functional orientation - requires that the set of requirements state what the system is to do, how well it is to do it, necessary external interface characteristics, environmental conditions under which other requirements are to be satisfied, any constraints on the utilization of external resources, any requirements with respect to overall physical characteristics, and any other required qualities such as those related to reliability, maintainability, transportability, growth capability, safety, etc. Functional orientation requires that design requirements (i.e. solution directed in requirements) be confined to those design directions that can be justified on objective criteria. Possible justifications include greater requirer expertise than that of the developer (e.g. in cryptography), or net benefits from standardization.

Requirements analysis aims to reduce the level of risk arising from defects relating to the above characteristics to “low”.

### Techniques of Requirements Analysis

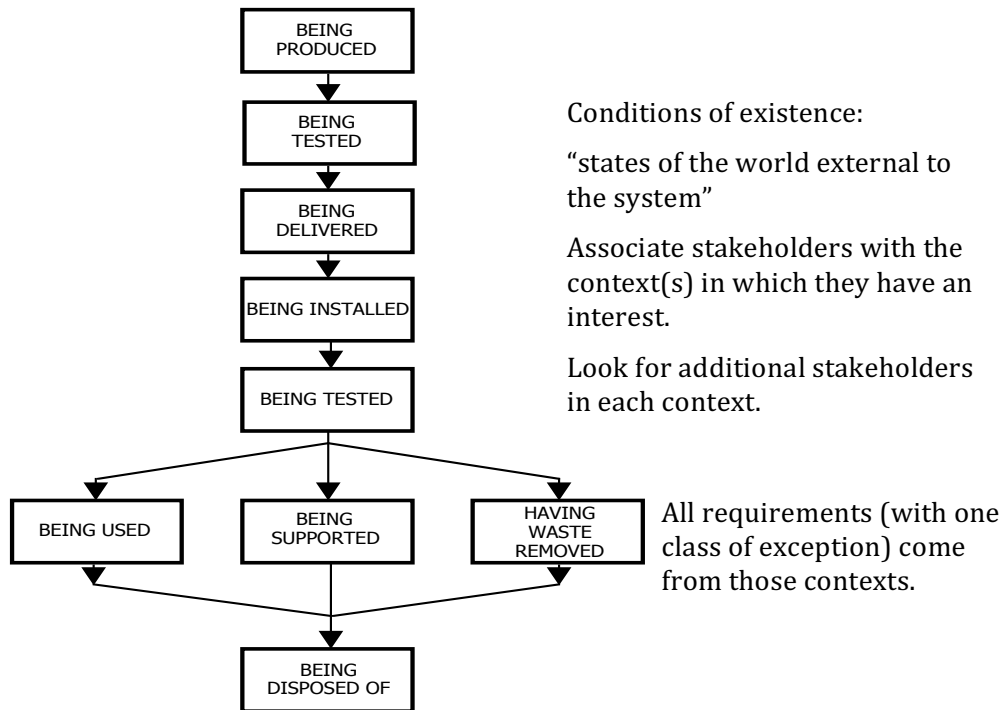
The requirements analysis process used and recommended by the author is illustrated in Figure 1.



**Figure 1: An Effective Requirements Analysis Process**

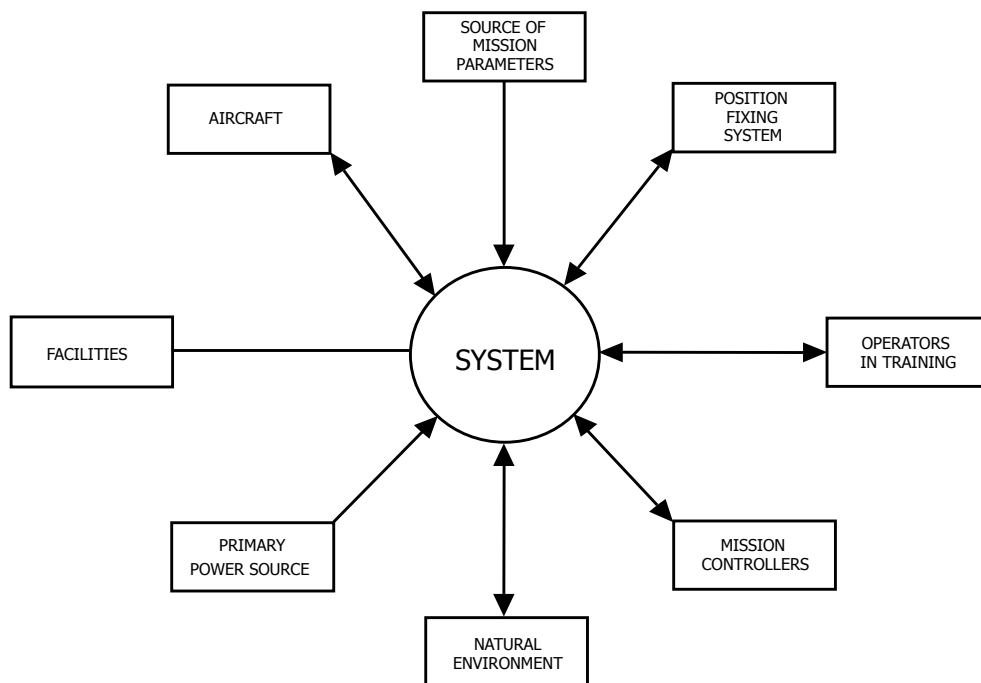
The set of techniques which combine to comprise a very effective and efficient requirements analysis methodology is described below:

- Stakeholder Identification.** The objective of stakeholder identification is to identify stakeholders who are potential “owners” of requirements, or who can facilitate effective communication relating to requirements. These stakeholders are subsequently encouraged to make input into the definition of the requirements, are consulted regarding requirements issues, and are invited to “sign-off” on their subsets of requirements.
- Document Review.** Documents, if any, which contain or relate to intended use, requirements, and goals are examined, with a view to identifying key issues that should be resolved with stakeholders before requirements analysis proceeds too far. This review provides input into the planning for conduct of the requirements analysis.
- Context Flow Analysis.** This analysis tracks the state of the world outside of the system on a whole of life basis, from system cradle to system grave. All requirements of the system originate in these contexts, with one class of exception. Stakeholders are mapped to the contexts, often resulting in the identification of additional stakeholders. The main work product of this analysis is subsequently used to structure analysis work, checks and dialog with stakeholders. See Figure 2.



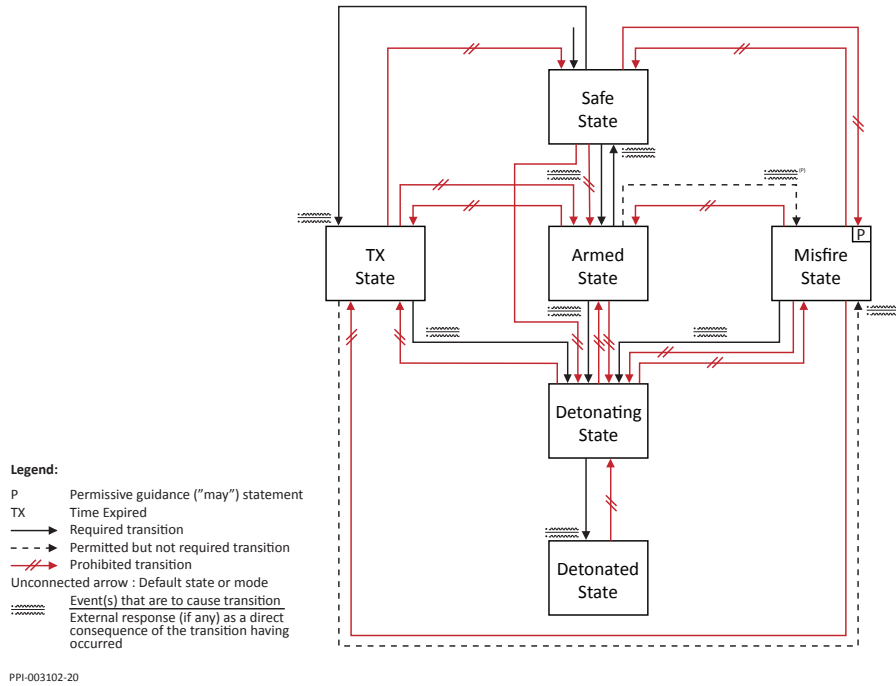
**Figure 2: Context Flow Diagram**

d. **Context Analysis.** This analysis identifies/validates mainly external interface requirements. The analysis also contributes to environmental requirements. Context analysis helps identify additional stakeholders in the system: owners of interoperating systems; individuals who will interact with the system; and organizational entities with which the system will interface. Context analysis sets the foundation for subsequent capture and validation of required functionality. See Figure 3.



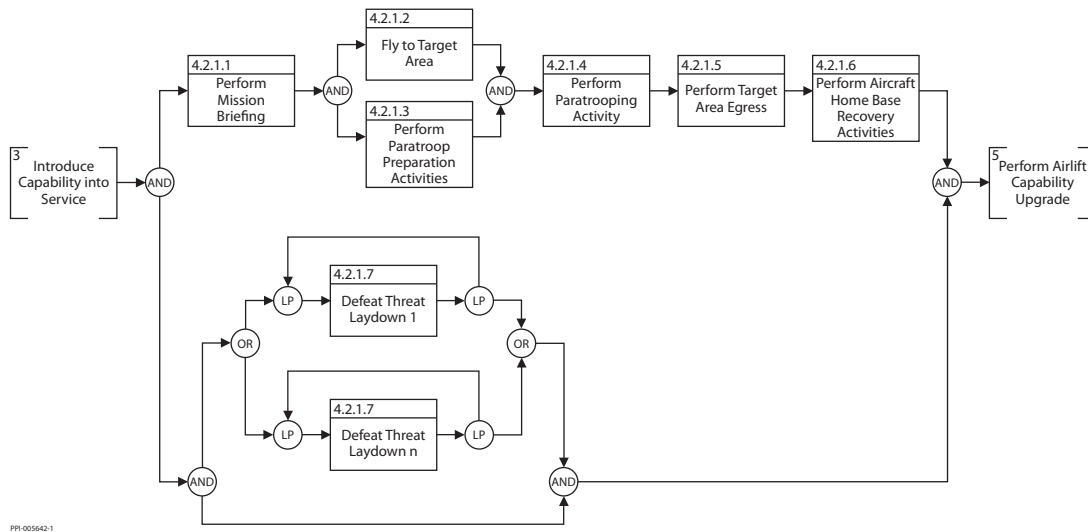
**Figure 3: Context Diagram**

e. **States and Modes Analysis.** This is a high ROI analysis, which establishes the big-picture dynamics required of the system, expressed in terms of states & modes. States and modes analysis often identifies major requirements issues. The analysis also establishes preconditions for subsequent precise and concise specification of the requirements captured in other analyses. See Figure 4.



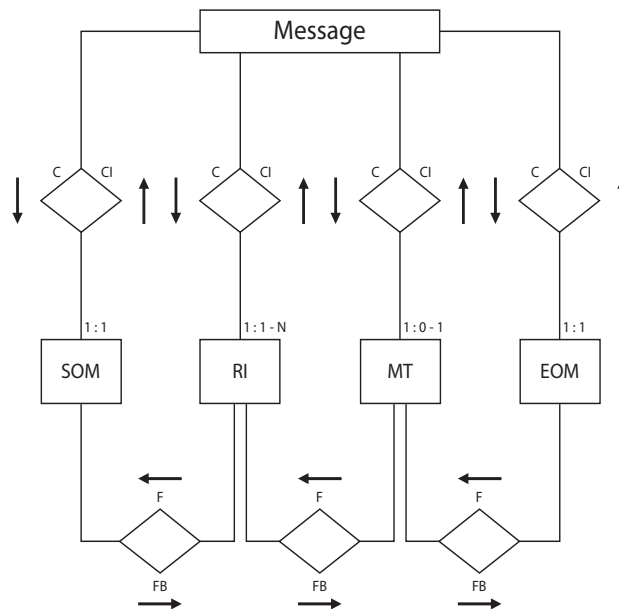
**Figure 4: States Transition Diagram**

f. **Functional Analysis.** This analysis is conducted within a modeling boundary which encapsulates enough of the problem, including functional aspects of operational scenarios, to capture and validate the required system functional and performance requirements. The result is a set of functional and performance requirements which is sufficiently complete and is at precisely the correct level of abstraction, neither too broad nor at a level of abstraction which directs the implementation of the system, as opposed to capturing the need. Use cases are a basic form of functional analysis; more robust functional modeling techniques can be used for more demanding applications.



**Figure 5: Functional Flow Block Diagram**

- g. **Rest of Scenario Analysis.** This analysis, conducted iteratively with functional analysis, identifies/validates environmental requirements, physical requirements, resource requirements and contributes additional content to external interface requirements.
- h. **Entity Relationship Attribute Analysis.** ERA analysis provides input to capture/validation of additional information content of external interface requirements, and some aspects of functional requirements. The analysis is most relevant to data-oriented systems.



**Figure 6: Entity Relationship Attribute Diagram**

- i. **Parsing Analysis.** Parsing analysis is a text analysis technique for identification of errors, incompleteness, inconsistency, lack of clarity, ambiguity, lack of verifiability, and infeasibility, in textually stated requirements. The basis of the technique is illustrated below:

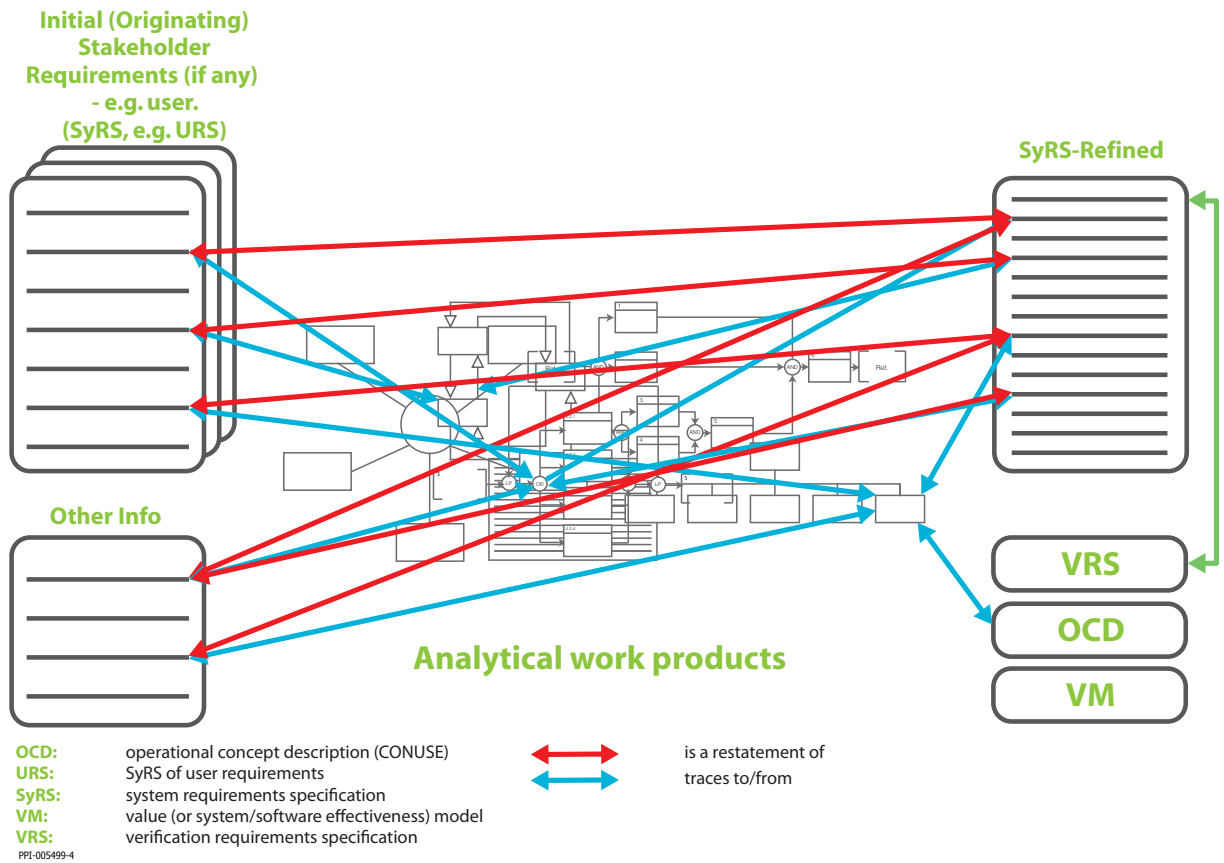
|                                 |   |
|---------------------------------|---|
| Actor:                          | The system,   |
| Condition:                      | upon receipt of a message,                                      |
| Action:                         | shall switch  |
| Object of Action:               | that message  |
| Constraints of Action:          | within 10 milliseconds of receipt,                              |
| Refinement of Object:           | for messages in ACPI28 format having a valid routing indicator, |
| Source of Object                | from the message input port,                                    |
| Destination of Action:          | to a message output port,                                       |
| (Further) Refinement of Action: | corresponding to the routing indicator in the message.          |

**Figure 7: Parsing Template**

The parsing template also provides an excellent aid to writing good requirements the first time, and for rewriting defective requirements.

- j. **Out-of-Range analysis.** This analysis captures and validates any requirements that relate to defective inputs or outputs or abnormal conditions of use/support/disposal. The requirements from this analysis can make the difference between a system that will be effective in the real world, and a system that could be effective only in the ideal world.
- k. **Other Constraints Search.** This activity looks for requirements which are ordained from on high (such as from statute law, applicable regulations, policy, governing standards, directives).
- l. **Clean-Up.** This activity verifies the refined requirements set, looking for residual defects in the work products of the analysis. Keyword searching is used in combination with specific verification criteria.

**Conclusion:**



**Figure 8: Approach to Requirements Analysis**

Methods exist to perform requirements capture and validation both efficiently and very effectively. The methods rely, not on requirements elicitation per se (which is neither efficient nor effective), but on elicitation of responses from stakeholders to specific requirements issues identified mainly through effective analysis of the problem domain.