

Requirements Categorisation

**A Report from the INCOSE
Requirements Working Group**

Toby Waters

INCOSE UK Autumn Assembly 2001

13th November 2001

Introduction

- Software Requirements Engineering (SRE) mailing list question:
 - *Where can I find guidance in categorising requirements?*
- Handled at a high level or poorly in most texts - sometimes almost trivially.
- Project set up in INCOSE RWG.

Project Team

Andrew Gabb (Editor)

George Caple

Paul Davies

Steve Eppig

David Haines

Anthony Hall

David Jones

Dave Lamont

Jim van Gaasbeek

Bill Vietinghoff

Consultant, Australia

GECM

Racal, UK

Northrop Grumman

Boeing

Praxis Critical Systems, UK

Raytheon

BAE Systems

Northrop Grumman

Boeing

Objectives

- Clarify the categorisation of requirements for complex projects.
- Suggest different category sets and categories.
- Improve the standardisation and communication of requirements.
- Provide guidance to practitioners in the use of categories.
- Provide a cookbook for categorisation, *not* a recipe.

Why Categorise Requirements?

- Present/list requirements according to different viewpoints and audiences.
- Determine the requirements applicable to different aspects of development.
- Assess relationships between associated requirements - conflicts, overlaps, completeness, consistency.
- Plan, model and manage requirements related activities.
- Use as a basis for requirements attributes in RM tools.

What is a Requirement? A new definition!

Definition: An expression of a perceived need that something be accomplished or realised.

Includes:

- Product, work, programmatic, service and other requirements (including 'constraints').
- Incorrect requirements.
- Poor requirements and poorly expressed requirements.
- Not just statements in natural language.
- Technical and non-technical requirements.
- Wants and desires.
- Non-binding provisions and prioritised requirements.

Alternative views of requirements

- A requirement is mandatory. Other provisions are guidance. [Mil-Std-961]
- A requirement is clear, consistent, testable, etc. (i.e. *good*).
- Needs, expectations, 'desirements', constraints are not requirements.
- Product is everything - service is a product.

I agree with some of these some of the time

Drawing the boundaries - Scope

- Prior to detailed design.
- Avoids much of the lower level design issues.
- Very difficult, because of different approaches to and levels of requirements efforts.
- Not included: assigned to, allocated to, verified by, etc.
- Categories are not RM tool *attributes* as such, cf. rationale, ID etc. but a likely subset of attributes.

Overview of category sets

Intrinsic Characteristics

- Basic Type
- Product/Process (PP) Type
- Quantitative/Qualitative (QQ) Type
- Life Cycle Phase

Priority and Importance

- Priority (Compliance Level)
- Importance

Source and Ownership

- Derivation status
- Source (Origin)
- Stakeholder
- Owner (Approval Authority)

Context

- Requirements Set / Document
- Subject
- Scope

Verification and Validation

- Verification Method
- Verification Stage
- Verification Status
- Validation Method
- Validation Stage
- Validation Status

Miscellaneous Category Sets

- Status
- Maturity (Stability)
- Risk Level
- Cost
- Product Release

Intrinsic characteristics group

Category Set	Categories
Basic Type	Functional, performance, quality factor, environment, interface, constraint, non-requirement. ...
Product/Process Type	Product, process, data, service.
Quantitative/Qualitative Type	Quantitative, qualitative.
Life Cycle Phase	Pre-concept, concept, development, manufacturing, integration/test, deployment/delivery/installation, operation, support, disposal.

Basic types

Functional	What is to be accomplished.
Performance	How well the functions are accomplished.
Quality Factor	Addresses other factors of product or process quality. ...
Environment	Requirements defining the physical (and possibly socio-political-economic) environment for system operation, or in which the work is done.
Physical	Requirements for the form of the product.
Interface	Requirements affecting product interfaces.
Constraint	Constraints on how and where the other requirements apply, or how the work is to be performed.
Non-requirement	Provided for completeness or to provide clarification or context, e.g. scenarios.

Basic types - Quality Factor sub-categories (examples)

Workmanship

Reliability

Availability

Maintainability

Supportability

Portability

Flexibility

Usability

Safety

Security

Integrity

Comments on Basic Types

- By the far the hardest to determine and get agreement on.
- The Functional/Performance separation is traditional, but flawed.
- 'Non-functional requirements' has low meaning and is highly variable in scope - omitted.
- Constraints includes design constraints.
- Functional and Performance requirements may include support requirements.
- Service requirements would generally be classified as either Functional or Performance requirements.
- Overlap between the {Functional, Performance} and {Environment, Interface, Quality Factors}.

Lessons Learned

- Many hands don't make light work.
- Good hands add significantly to completeness and quality.
- Significantly different opinions and viewpoints, within the team and in standard texts.
- Agreement on concepts can be easy - agreement on words can be impossible.
- Requirements engineering is very young and ill-defined.
- *RE has a long and hard way to go!*

Definition: An expression of a perceived need that something be accomplished or realised.

Intrinsic Characteristics

- Basic Type
- Product/Process (PP) Type
- Quantitative/Qualitative (QQ) Type
- Life Cycle Phase

Priority and Importance

- Priority (Compliance Level)
- Importance

Source and Ownership

- Derivation
- Source (Origin)
- Stakeholder
- Owner (Approval Authority)

Context

- Requirements Set / Document
- Subject
- Scope

Verification and Validation

- Verification Method
- Verification Stage
- Verification Status
- Validation Method
- Validation Stage
- Validation Status

Miscellaneous Category Sets

- Status
- Maturity (Stability)
- Risk Level
- Cost
- Product Release