ABC Bank

Requirements Management Plan

Version 1.0

Revision History

Date	Version	Description	Author
21-Feb-2000	0.1	Initial Session Feedback Applied	Stephen Hunt(slh) (Rational Software)
26-Feb-2000	0.2	Follow-up interview feedback applied	slh
7-Mar-2000	0.3	Added Issues, Assumptions, Biz Rules, and initial trace model	slh
9-Mar-2000	0.4	First crack at all requirement attributes	slh
17-Mar-2000	0.5	Revised requirement attributes	slh
22-Mar-2000	0.6	Added Design Requirements and extension list.	slh
24-Mar-2000	0.7	First Draft Proof	slh
26-Mar-2000	0.8	Revisions from internal review	slh
30-Mar-2000	0.9	Moved assumptions and issues to non-document; Fixed hard/neutral/soft for several attributes	slh

Table of Contents

1. INTRODUCTION 5

1.1 PURPOSE 5
1.2 SCOPE **ERROR! BOOKMARK NOT DEFINED.**
1.3 COLLABORATORS 5
1.4 DEFINITIONS, ACRONYMS AND ABBREVIATIONS 5
1.5 REFERENCES 7
1.6 OVERVIEW 8

2. REQUIREMENT ARTIFACTS AND REQUIREMENT TYPES 9

Table 2-1 Document based Requirement Artifacts and Types 9
Table 2-2 Database Only Requirement Types 10

3. REQUIREMENT ATTRIBUTES 11

3.1 REQUIREMENT ATTRIBUTES FOR IMPACTED GROUP(IG) 11
3.2 REQUIREMENT ATTRIBUTES FOR STAKEHOLDER(STK) 11
3.3 REQUIREMENT ATTRIBUTES FOR STAKEHOLDER NEED(NEED) 11
3.4 REQUIREMENT ATTRIBUTES FOR FEATURE(FEAT) 12
Table 3-1 Status attribute values for FEAT requirement type. 12
Table 3-2 Benefit attribute values for FEAT requirement type. 12
Figure 3-2 Percent engineering hours by phase. [2] 13
Table 3-3 Coordination Complexity attribute values for FEAT requirement type. 14
Table 3-4 Architectual Impact attribut values for FEAT requirement type. 14
Table 3-5 Development Risk Scoping Matrix. [2] 14
Table 3-6 Stability attribute values for FEAT requirement type. 15
Table 3-7 Business Process Impact values for FEAT requirement type. 15
3.5 REQUIREMENT ATTRIBUTES FOR ACTOR(ACTOR) 15
3.6 REQUIREMENT ATTRIBUTES FOR USE-CASE(UC) 15
3.7 REQUIREMENT ATTRIBUTES FOR USE-CASE DETAIL(UCDR) 16
Table 3-8 Location attribute values for UCDR requirement type. 16
Table 3-9 Stability attribute values for UCDR requirement type. 17
3.8 REQUIREMENT ATTRIBUTES FOR SUPPLEMENTAL(SUPP) 17
Table 3-10 Stability attribute values for SUPP requirement type. 18
3.9 REQUIREMENT ATTRIBUTES FOR DESIGN(DE) 18
Table 3-11 Type attribute values for DE requirement type. 18
Table 3-12 Status attribute values for DE requirement type 19
3.10 REQUIREMENT ATTRIBUTES FOR TEST PLAN(TPR) 19
Table 3-13 Status attribute values for TPR requirement type. 19
3.11 REQUIREMENT ATTRIBUTES FOR TEST(TR) 19
Table 3-14 Status attribute values for TR requirement type. 20
3.12 REQUIREMENT ATTRIBUTES FOR ISSUE(ISS) 20
Table 3-15 Status attribute values for ISS requirement type. 20
3.13 REQUIREMENT ATTRIBUTES FOR ASSUMPTION(ASM) 20
Table 3-16 Status attribute values for ASM requirement type. 21
3.14 REQUIREMENT ATTRIBUTES FOR TERM(TERM) 21
Table 3-17 Status attribute values for TERM requirement type. 21
3.15 REQUIREMENT ATTRIBUTES FOR BUSINESS RULE(BR) 21
Table 3-18 Status attribute values for BR requirement type. 21

TRACABILITY CRITERIA 22

FIGURE 4-1 REQUIREMENTS TRACABILITY DIAGRAM 22

4.1 Criteria for Impacted Group 23
4.2 Criteria for Stakeholder 23
4.3 Criteria for Stakeholder Need Requirements 23
4.4 Criteria for Product Feature Requirements 23
4.5 Criteria for Use-Case Requirements 23
4.6 Criteria for Actor Requirements 23
4.7 Criteria for Use-Case Detail Requirements 24
4.8 Criteria for Supplemental Requirements 24
4.9 Criteria for Design Element Requirements 24
4.10 Criteria for Test Plan Requirements 24
4.11 Criteria for Test Requirements 24
4.12 Criteria for Issue Requirements 24
4.13 Criteria for Glossary Requirements 24
4.14 Criteria for Assumption Requirements 24
4.15 Criteria for Business Rule Requirements 25
4.16 Criteria for Supporting Document Requirements 25

5. RATIONAL REQUISITEPRO® VIEWS 25

5.1 Attribute Matrix views for each Requirement Type limiting non-user attributes 25
5.2 Scoping View, see Target Release in 3.4 25

6. RATIONAL REQUISITEPRO® EXTENSIONS 25

6.1 Calculate Feature dependency count 25
6.2 Calculate requirement level counter 25
6.3 Calculate Technology Risk 25
6.4 Trace Check the requirements 25
6.5 Soda document for UCMS 25

APPENDIX 1: TECHNOLOGY RISK ASSESSMENT 26

APPENDIX 2: TRACABILITY DIAGRAMMING NOTATION 27

Requirements Management Plan

1. Introduction

1.1 Purpose

This document describes the guidelines used by the ATM Display project within ABC Bank (ABCB) for establishing the requirements documents, requirement types, requirements attributes, and tracability in order to manage their software project requirements. It will also serve as the configuration document for *Rational Requisite®Pro* and *Rational RequisiteWeb* tools.

1.2 Scope

This plan will be a guideline for all software projects performed by ABCB, IT department.

1.3 Collaborators

- Mark Herbst
- Tina Newell
- Bob Meto
- Stan Johns
- Buck Del'Ort
- Eric Hirch
- Heather Buck
- Vernon Martin
- Dr. Paul Sangerman
- Max Nail

1.4 Definitions, Acronyms and Abbreviations

Baseline

A reviewed and approved release of artifacts that constitutes an agreed basis for further evolution or development and that can be changed only through a formal procedure, such as change management and configuration control.

Business Rule

A formal regulation or bylaw imposed by an organization or simply the standard practices of users governing the way the organization conducts its business. Business rules may be classified as Definitions, Facts (Relationships, Connections), Constraints ('must have' versus 'must not have') and Derivation Rules (inferring new facts from existing ones).

Business Unit Manager

A member of an ABCB business unit. Responsible and accountable for communicating requirements for systems to IT and for accepting delivery of systems.

Customer

The economic buyer of a project developed by IT.

Cyclomatic Complexity

The most widely used member of a class of static software metrics. It may be considered a broad measure of *soundness* and *confidence* for a program. Introduced by Thomas McCabe in 1976, it measures the number of linearly independent paths through a program module. This measure provides a single ordinal number that can be compared to the complexity of other programs. Cyclomatic complexity is often referred to simply as program complexity, or as McCabe's complexity. It is often used in concert with other software metrics. As one of the more widely-accepted software metrics, it is intended to be independent of language and language format.[8]

Engineering Time

A measurement unit describing engineering effort. Usually expressed in units of weeks or months. The move away from terms like man-months, or person-months is deliberate. Men and months are interchangeable commodities only when a task can be partitioned among many workers *with no communication among them*. [12] In most uses, engineering time is used to understand the relative size of something, not as an advertised elapsed time to complete a task.

NCCS

Non-Commented Source Statements. A metric used to estimate project risk, estimate schedules, and most importantly, a component of software release decision when used in defect density calculation.

Pareto Chart

A useful tool for graphically depicting where allocating time, human, and financial resources will yield the best results. Dr. Joseph Juran (of total quality management fame) formulated the Pareto Principle after expanding on the work of Wilfredo Pareto, a nineteenth century economist and sociologist. The Pareto Principle states that a small number of causes is responsible for a large percentage of the effect--usually a 20-percent to 80-percent ratio.

Product Feature

A capability or characteristic of a system that directly fulfills a *Stakeholder Need.* Often thought of as the "advertised benefits" of the system.

Rational Requisite®Pro

Rational Requisite®Pro helps teams organize, prioritize, track and control changing requirements of a system or application.

Rational Requisite®Web

Rational Requisite®Web helps teams organize, prioritize, track and control changing requirements of a system or application via a Web Browser interface.

Rational Rose®

Rational Rose® is a graphical component modeling and development tool, using the industry-standard Unified Modeling Language (UML).

Rational SoDA®

Rational SoDA® provides automatic generation of software documentation.

Rational TestManager™

Designed to help you track software testing information through all phases of the software development, test, and revision cycles. You can use TestManager™ to plan testing strategies, and to track information related to test execution.

Stakeholder

A stakeholder is defined as anyone who is materially affected by the outcome of the project. Effectively solving any complex problem involves satisfying the needs of a diverse group of stakeholders. Stakeholders will typically have different perspectives on the problem, and different needs that must be addressed by the solution.

Stakeholder Need

The business or operational problem (opportunity) that must be fulfilled in order to justify purchase or use. Also known as goal or objective.

Vision Document

A general vision of the core project's requirements it provides the contractual basis for the more detailed technical requirements. This is a project management document owned by the IT Project Manager. A System Analyst authors it with primary input educed from the *Stakeholders.*

1.5 References

Applicable references are:

1. Rational Unified Process 5.5, Copyright © 1987 – 1999 Rational Software Corporation
2. Grady R, *Practical Software Metrics For Project Management And Process Improvement*, Englewood Cliffs, NJ: Prentice-Hall, Inc., 1992, pp. 14, 42, 172-174
3. Grady, R. and D. Caswell, *Software Metrics: Establishing a Company-Wide Program*, Englewood Cliffs, NJ: Prentice-Hall, Inc., 1987, pp. 34, 65, 111, 112, 113
4. Card, D., V. Church, and W. Agresti, "An Empirical Study of Software Design Practices," *IEEE Transactions of Software Engineering*, Vol. SE-12, no. 2, (Feb. 1986), pp. 264-271
5. Cash, J., F. McFarlan, J. McKenny, and L. Applegate, *Corporate Information Systems Management: Text and Cases*, Boston, MA: Richard D. Irwin, Inc., 1992, pp. 418-426
6. Spence, I. And L. Probasco, *Tracability Strategies for Managing Requirements with Use-Cases*, Cupertino, CA: Rational Software Corporation, 1998
7. Brooks, F., *The Mythical Man-Month*, Reading, MA: Addison Wesley Longman, Inc., 1998, pp. 16-26
8. Mc McCabe, T. & A. Watson, "Software Complexity," *Crosstalk, Journal of Defense Software Engineering 7*, 12 (December 1994): pp. 5-9.
9. Vision Document Template, V 0.1, Draft, 2000
10. Supplementary Specification Template, V 0.1, Draft, 2000
11. Use-Case Specification Template V 0.1, Draft, 2000
12. Test Plan Template, V 0.1, Draft, 2000
13. Glossary Template, V 0.1, Draft, 2000
14. Assumptions Template, V 0.1, Draft, 2000
15. Issues Template, V 0.1, Draft, 2000
16. Business Rules Template, V 0.1, Draft, 2000
17. Use-Case Model Survey Template, V 0.1, Draft, 2000

1.6 Overview

This Requirements Management Plan is being created to address identified problems in the requirements management process experienced in previously in software projects delivered by ABCB IT. Problems included:

- poor communication of changes to requirements, including the use of email to communicate changes;
- data changes out of sync with code changes;
- lack of formal handoffs between team members for software artifacts;
- minimal contact with stakeholders;
- users not knowing what they want until they see it;
- fast pace of requirements change;
- geographically disbursed team, a problem that has been remedied;
- lack of clear understanding of roles within the requirements process;
- separation of the subject matter experts and developers may result in decreased customer satisfaction;
- inconsistent documentation;
- inability to easily find requirement documents;
- problems are often to largely scoped;

What follows is in response to these problems. A standard set of documents used to express, requirements of all levels will be defined. An established set of requirement types to capture stakeholder problems, needed features of the system, software requirements, test requirements, standard terms and business rules will be described. For each requirement type a collection of requirements attributes, used to manage delivery of the needed system and manage changes to the requirements over the lifecycle of the project, will be identified along with values and ranges appropriate for each. A model of traces between requirements will be established to help communicate requirements change to all members of the project team. An initial list of predefined views of requirements information will be defined. This list must evolve with use. A set of tool extensions to provided needed functionality specifically for ABCB will be created. Finally, a list of roles within the requirements management process will be identified.

The premise of this endeavor from the start was that an initial process and tool configuration for requirements management be defined, and that these would evolve through use. This being the case, this document should be considered a “living” document and changes to it over time and experience are expected. Changes to this document should be made in a controlled manner and only after review by impacted groups.

The project management strategy taken by ABCB IT is one that places an emphasis on customer satisfaction. This plan embraces that strategy.

2. Requirement Artifacts and Requirement Types

ARTIFACT	REQ. TYPE	DESCRIPTION
Vision (VIS)	Stakeholder Need (NEED)	The business or operational problem (opportunity) that must be fulfilled in order to justify purchase or use. Also known as goal or objective. Provided by *Stakeholders.*
	Product Feature (FEAT)	This is the default requirement type for the Vision Document. Conditions or capabilities the system. Some Product Features may be out of Baseline for a particular release of the system.
Glossary (GLS)	Term (TERM)	The Glossary defines important terms used in the project. Owned and authored by the System Analyst. Content provided by *Stakeholders* and ABCB IT.
Business Rule Reference(BR)	Business Rule (BR)	Defines business logic or business data used in the problem domain. Owned and authored by the System Analyst. Content provided by Business.
Use-Case Model Survey		Generated *SoDA®* report providing a high-level view of all Use-Cases and Actors for this release documented in *Rational Rose®.*
Use-Case Specification (UC)	Use-Case Detail Requirement (UCDR)	Individual detailed requirements as specified in the use-case specification. These are also known as software requirements. Owned and authored by the System Analyst. Content provided by *Stakeholders.*
Supplementary Specification (SS)	Supplementary Requirement (SUPP)	The Supplementary Specifications capture the system requirements that are not readily captured in the use cases of the use-case model. Such requirements include: legal and regulatory requirements and application standards; quality attributes of the system to be built, including usability, reliability, performance and supportability requirements; other requirements such as operating systems and environments, compatibility requirements, and design constraints Owned and authored by the System Analyst. Content provided by *Stakeholderss.*
Test Plan(TPL)	Test Plan Requirement (TPR)	The Test Plan contains information about the purpose and goals of testing within the project, identifies the strategies to be used to implement and execute testing, specific high-level test requirements, and resources needed. A Test Plan Requirement (TPR) is owned and authored by the Test Designer. At a minimum, each Use-Case should derive at least one TPR. A more detailed list of TPRs would include at least one TPR for each Use-Case flow of events.
Test Case(TC)	Test Requirement (TR)	A Test Case is a set of test inputs, execution conditions, and expected results developed for particular objective, such as to exercise a particular program path or to verify compliance with a specific requirement. A Test Case implements all or part of a Test Plan Requirement. A Test Requirement (TR) identifies a point in the test case where a verification of state of the System Under Test is needed.

Table 2-1 Document based Requirement Artifacts and Types

REQUIREMENT TYPE	DESCRIPTION
Impacted Group(IG)	A group, area, or division that may request ABCB IT services or may be impacted by one or more systems developed by ABCB IT (e.g. Marketing, Sales Associates, Legal, IT, Accounts Payable). Owned and authored by the System Analyst.
Stakeholder(STK)	An individual stakeholder name. Owned and authored by the System Analyst.
Actor(ACTOR)	Someone or something, outside the system that interacts with the system.
Design Element(DE)	It identifies the design components of the system being built, sometimes down to the method level. Owned and authored by the Architect, Designer or Implementers.
Use-Case (UC)	A high-level reference to a Use-Case.
Issue(ISS)	A matter that is in dispute between two or more parties and, left unsettled might result in poorer product quality or a schedule slip. Authored by anyone on the project team.
Assumption (ASM)	A fact or statement taken for granted. Authored by anyone on the development team. Should be agreed upon by both the Business Unit Manager and the development team.

Table 2-2 Database Only Requirement Types

3. Requirement Attributes

3.1 Requirement Attributes for Impacted Group(IG)

Requirement text is the name of the impacted group.

Level

Calculated by RequisitePro, this attribute represents the height of the requirement in a hierarchy. The higher the number, the more levels of requirements that this requirement traces to. The calculation reports the deepest trace network.

3.2 Requirement Attributes for Stakeholder(STK)

Requirement text is the name of a Stakeholder.

3.3 Requirement Attributes for Stakeholder Need(NEED)

Requirement text consists of one or more phrases which describe the problem or need.

Problem Analyzed

Indicates whether problem analysis has been performed on the requirement. Yes or No.

Contribution

Indicates the problem contribution to the overall business opportunity or problem being addressed by the project. Percentage (0%-100%). All contributions should sum no greater than 100%.

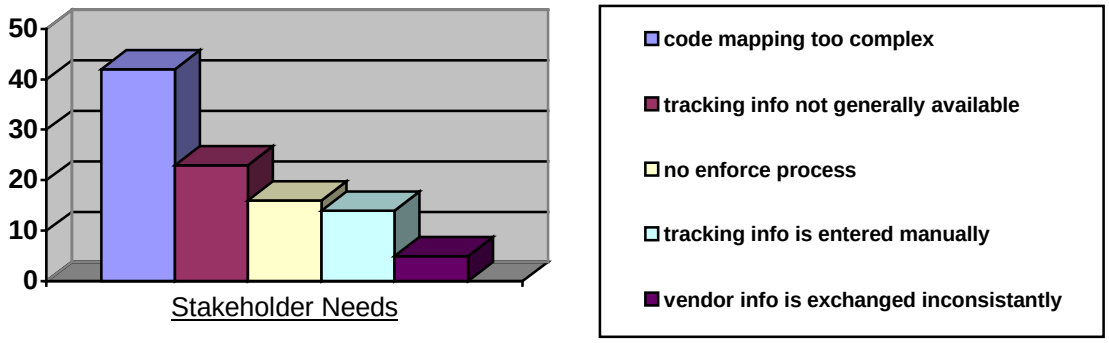

Figure 3-1 *Pareto Chart* for root problems of Special Orders management.

Level

Calculated by RequisitePro, this attribute represents the height of the requirement in a hierarchy. The higher the number, the more levels of requirements that this requirement traces to. The calculation reports the deepest trace network.

Dependency

Calculated by RequisitePro, this attribute represents the number of peer requirements to which this requirement traces-to.

3.4 Requirement Attributes for Feature(FEAT)

Requirement Text is the feature description.

Status

Set after negotiation and review by the project management team and Business Unit Managers. Tracks progress during definition of the project baseline. Used to control scope.

Proposed	Used to describe features that are under discussion but have not yet been reviewed and accepted by a working group consisting minimally of Business Unit Managers and IT Project Management. **Not in the project baseline.**
Approved	Capabilities that are deemed useful and feasible and have been approved for implementation by the working group. **Not in the project baseline.**
Incorporated	Features incorporated into the product baseline at a specific point in time.
Validated	Features have been implemented and validated.

Table 3-1 Status attribute values for FEAT requirement type.

Benefit

Set Business System Manager. Ranking requirements by their relative benefit to the business opens a dialogue with customers, analysts and members of the development team. Used in managing scope and determining development priority.

Critical	Essential features. Failure to implement means the system will not meet customer needs. All critical features must be implemented in the release.
Important	Features important to the effectiveness and efficiency of the system for most applications. The functionality cannot be easily provided in some other way. Lack of inclusion of an important feature may affect customer or user satisfaction, or even revenue, but release will not be delayed due to lack of any important feature.
Useful	Features that are useful in less typical applications, will be used less frequently, or for which reasonably efficient workarounds can be achieved. No significant revenue or customer satisfaction impact can be expected if such an item is not included in a release.

Table 3-2 Benefit attribute values for FEAT requirement type.

Effort

Set by the development team. Because some features require more time and resources than others, estimating the engineering time is the best way to gauge complexity and set expectations of what can and cannot be accomplished in a given time frame. Used in managing scope and determining development priority.

When estimating effort keep in mind all of the activities associated with the production of a software product. Figure 3-1 shows a breakdown that may be used until ABCB IT develops metrics and adjusts the graph accordingly. One use of the graph is to start with the estimated level of effort to perform one of the graphed activities, extrapolate the remaining activities, and total them for the Effort attribute. Rule of thumb: Projects created primarily from reused software take about one fourth the time and resources of those that are new. [3,4] Measurement will be integer value, unit of engineering weeks.

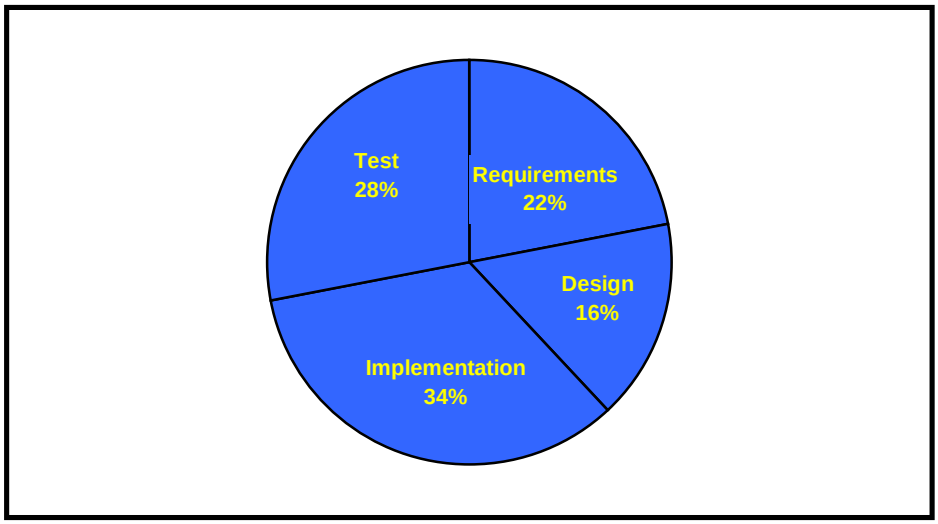

Figure 3-2 Percent engineering hours by phase. [2]

Size

Set by the development team based on the estimated number of non-commented source statements needed to implement the feature. Used to help establish development risk. The greater the number of lines of code the greater the complexity of the project. Reused software lines of code should be counted at a quarter of their number. Measurement will be an integer value, number of non-commented source statements/1000(KNCSS).

Coordination Complexity

Set by analyst and development team based on the reliance on organizations outside their control needed to implement the feature. Used to help establish development risk.

Internal	ABCB will be the sole source for input and integration.
External	One or mores ABC Co. business areas other than ABCB will be the source for input and the product will integrate with systems outside ABCB IT control.
Geographic	The source for input and/or the product will integrate with systems geographically distant from ABCB IT.
Vendor	A vendor will be responsible for development of a significant portion of the system.

Table 3-3 Coordination Complexity attribute values for FEAT requirement type.

Technology Risk

Set by the development team based on a Technology Risk Assessment (TRA), Appendix 1, for the requirement. Normally performed at the project level, the TRA allows the development team to understand the slope they must climb to deliver the goods. Used to help establish development risk. Measurement will be an integer value between zero and one hundred-fifty. A value of zero indicates no assessment has been performed.

Architectural Impact

Set by the Software Architect, this indicates how this feature will impact the architecture.

None	The feature should use the existing software architecture without modification.
Extends	The feature will require extending the current software architecture.
Modifies	The feature will require a refactoring of the software architecture.

Table 3-4 Architectual Impact attribut values for FEAT requirement type.

Development Risk

Calculated by the *RequisitePro® tool*. Each risk component requirement attribute(i.e. effort, size, coordination complexity, technology risk, architectural impact) is evaluated for its value in the matrix and the resulting value is multiplied by the weight and totaled. The result is an integer valued between zero and one hundred. This measures the probability the project will experience undesirable events, such as cost overruns, schedule delays or even cancellation.

Weight (x4)	0	2	5
Effort	0 – 6 Eng. Mo.	7 – 18 Eng. Mo.	> 18 Eng. Mo.
Size	< 3KNCSS	3KNCSS – 15KNCSS	> 15KNCSS
Coord. Complexity	Internal	External or Geographic	Vendor
Technology Risk	< 20	20 - 60	> 60
Architectural Impact	None	Extends	Modifies

Table 3-5 Development Risk Scoping Matrix. [2]

Stability

Set by the Business System Manager and development. Used to help establish development priorities and determine the items for which additional exploration and discovery is the appropriate next action.

Hard	It is very unlikely that this feature will change, or that the development team's understanding of the feature will change.
Neutral	No indicator exists to predict the likelihood of change for this feature. DEFAULT
Soft	It is very likely that this feature will change, or that the development team's understanding of the feature will change.

Table 3-6 Stability attribute values for FEAT requirement type.

Target Release

Records the intended product version in which the feature will first appear. This field can be used to allocate features from a Vision document into a particular baseline release. When combined with the Status attribute, the team can propose, record and discuss various features of the release without committing them to development. Only features whose Status is set to Incorporated and whose Target Release is defined will be implemented. When scope management occurs, the Target Release Version Number can be increased so the item will remain in the Vision document but will be scheduled for a later release. Values: enumerates of pre-determined release names.

Impact to Business Process

Set by the Business System Manager. Indicates the implementation effort and risk. Is not used by IT.

High	Changes or creates a business process.
Medium	Replaces a business process.
Low	Automates a piece of a business process.

Table 3-7 Business Process Impact values for FEAT requirement type.

Level

Calculated by RequisitePro, this attribute represents the height of the requirement in a hierarchy. The higher the number, the more levels of requirements that this requirement traces to. The calculation reports the deepest trace network.

Dependency

Calculated by RequisitePro, this attribute represents the number of peer requirements to which this requirement traces-to.

3.5 Requirement Attributes for Actor(ACTOR)

Requirement text is the name of the actor.

Brief Description

A few sentences which describe what or whom the actor represents, why the actor is needed, and what interests the actor has in the system.

3.6 Requirement Attributes for Use-Case(UC)

Requirement text is the Use-Case name.

Brief Description

A few sentences that describe what an actor would use the Use-Case for. It should reflect the role and purpose of the Use-Case.

Affects Architecture

Set by the Software Architect, this indicates that the Use-Case flow of events touches a delicate point in the architecture or exercises a significant portion of the architecture. Values: True or False.

Planned Iteration

The iteration when this Use-Case will begin to be implemented. Integer value.

3.7 Requirement Attributes for Use-Case Detail(UCDR)

Requirement text describes what the system should do.

Section

Set by the Systems Analyst. Indicates the location within a Use-Case Report where the requirement lives.

Name	The requirement is found in the name of the Use-Case.
Brief Description	The requirement is found in the brief description of the Use-Case.
Basic Flow	The requirement is found in the basic flow of the Use-Case.
Alternate Flow	The requirement is found in an alternate flow of the Use-Case.
Special Requirements	The requirement is found in the special requirements section of the Use-Case.
Pre-Condition	The requirement is found in the pre-condition section of the Use-Case.
Post-Condition	The requirement is found in the post-condition section of the Use-Case.

Table 3-8 Location attribute values for UCDR requirement type.

Affects Architecture

Set by the Software Architect, this indicates the requirement has an influence on the software architecture. Values: True or False.

Effort

Set by the development team, it indicates the estimated effort required implementing and validating the requirement. Used in determining development priority. Measurement will be integer value, unit of engineering weeks.

Size

Set by the development team based on the estimated number of non-commented source statements estimated to implement the feature. Used in determining development priority. The greater the number of lines of code the greater the complexity and difficulty of the requirement. Measurement will be an integer value, number of non-commented source statements/1000(KNCSS).

Reviewed Ambiguity

Set by the Systems Analyst after the review of the encompassing Use-Case. It represents a count of different interpretations the reviewers had when looking at the requirement in context. Integer value. Zero default.

Stability

Set by Business Unit Manager and the Systems Analyst based on the probability the behavior expressed by the requirement will change or the team's understanding of that behavior will change. Used to help establish development priorities and determine those items for which additional elicitation is the appropriate next action.

Hard	It is very unlikely that this behavior will change, or that the development team's understanding of the behavior will change.
Neutral	No indicator exists to predict the likelihood of change for this behavior. Default value.
Soft	It is very likely that this behavior will change, or that the development team's understanding of the behavior will change.

Table 3-9 Stability attribute values for UCDR requirement type.

Level

Calculated by RequisitePro, this attribute represents the height of the requirement in a hierarchy. The higher the number, the more levels of requirements that this requirement traces to. The calculation reports the deepest trace network.

Dependency

Calculated by RequisitePro, this attribute represents the number of peer requirements to which this requirement traces-to.

3.8 Requirement Attributes for Supplemental(SUPP)

Requirement text describes what the system should do.

Affects Architecture

Set by the Software Architect, this indicates the requirement has an influence on the software architecture. Values: True or False.

Effort

Set by the development team, it indicates the estimated effort required implementing and validating the requirement. Used in determining development priority. Measurement will be integer value, unit of engineering weeks.

Size

Set by the development team based on the estimated number of non-commented source statements estimated to implement the feature. Used in determining development priority. The greater the number of lines of code the greater the complexity and difficulty of the requirement. Measurement will be an integer value, number of non-commented source statements/1000(KNCSS).

Reviewed Ambiguity

Set by the system analyst after the review of the encompassing Use-Case. It represents a count of different interpretations the reviewers had when looking at the requirement in context. Integer value. Zero default.

Stability

Set by Business Unit Manager and the Systems Analyst based on the probability the behavior expressed by the requirement will change or the team's understanding of that behavior will change. Used to help establish development priorities and determine those items for which additional elicitation is the appropriate next action.

Hard	It is very unlikely that this behavior will change, or that the development team's understanding of the behavior will change.
Neutral	No indicator exists to predict the likelihood of change for this behavior. Default value.

Soft	It is very likely that this behavior will change, or that the development team's understanding of the behavior will change.

Table 3-10 Stability attribute values for SUPP requirement type.

Level

Calculated by RequisitePro, this attribute represents the height of the requirement in a hierarchy. The higher the number, the more levels of requirements that this requirement traces to. The calculation reports the deepest trace network.

Dependency

Calculated by RequisitePro, this attribute represents the number of peer requirements to which this requirement traces-to.

3.9 Requirement Attributes for Design(DE)

Requirement text is the name of the design element.

Type

Set by Software Architect, Designer, or Implementers. Identifies the design element's level in the system.

Executable	A unit of compiled and linked code visible at the file system level.
Subsystem	A model element which has the semantics of a package, such that it can contain other model elements, and a class, such that it has behavior. (The behavior of the subsystem is provided by classes or other subsystems it contains). A subsystem realizes one or more interfaces, which define the behavior it can perform. A subsystem is a grouping of model elements, of which some constitute a specification of the behavior offered by the other contained model elements.
Package	A general purpose mechanism for organizing elements into groups. Packages may be nested within other packages.
Class	A description of a set of objects that share the same attributes, operations, methods, relationships and semantics. A class may use a set of interfaces to specify collections of operations it provides to its environment.
Method	The implementation of an operation. It specifies the algorithm or procedure associated with an operation.

Table 3-11 Type attribute values for DE requirement type.

Actual NCSS

Set by Software Architect, Designer, or Implementers. Used to calculate defect density trend for a release decision by the Project Manager. Actual number of non-commented source statements in the design element.

Status

Set by Software Architect, Designer, or Implementers. Used to track objective project status at the implementation level Minimizes repeated discussions or miss-directed expectations.

No Design	No design is complete for the element.
Designed	The design has been completed for the element.
Design Inspected	The element design has been inspected and accepted.
Coding	Coding for the element is in progress.
Code Inspected	Code for the element has been inspected.
Tested	Unit test has been performed on the element.

Table 3-12 Status attribute values for DE requirement type

Current Complexity

The current cyclomatic complexity level of the design element's code. Used as a predictor of risk. Also used as a prioritization method for the Quality Assurance group. Can usually be assessed automatically by a tool. Numeric.

Defects

Number of defects reported against the design element including those identified in design reviews, code reviews unit testing, integration testing, and system testing. Used as a predictor of risk in the software.

Test Coverage

Percentage of code covered by the unit test. Used as a predictor of risk in the software.

3.10 Requirement Attributes for Test Plan(TPR)

Requirement text is a few sentences that describe what the test requirement will do.

Assigned To

The test engineer responsible to write the Test Cases which will verify the requirement. Ideally, an enumerated list of test engineers.

Status

Set by the test assigned test engineer. Tracks progress during test development.

No Activity	No work has been accomplished in implementation of the requirement.
In Progress	Test Case(s) that verify the requirement are being written.
Written	Test Case(s) have been written but have not passed inspection.
Inspected	Test Case(s) have been written and have been inspected and accepted as valid.

Table 3-13 Status attribute values for TPR requirement type.

Planned Build

The build when the requirement will implemented.

3.11 Requirement Attributes for Test(TR)

Requirement text is one or more sentences that describe what the Test Requirement will verify.

Assigned To

The test engineer responsible to write the Test Case. Ideally, an enumerated list of test engineers.

Status

Set by the test assigned test engineer. Tracks progress during test development.

No Activity	No work has been accomplished in implementation of the requirement.
Planned	A script, either manual or automated has been identified and associated with the requirement.
Manual	A manual script has been created and validated as capable of proving the requirement.
Automated	An automated script has been created and validated as capable of proving the requirement.

Table 3-14 Status attribute values for TR requirement type.

Planned Build

The build when the Test Case must be written.

3.12 Requirement Attributes for Issue(ISS)

Requirement text defines the issue.

Created

Date the Issue was started.

Due Date

Date the Issue should be resolved.

Resolved

Date the Issue was started.

Assigned To

Set by the project manager or lead. The person responsible to resolve the issue. Ideally, enumerated list of development team members.

Status

Set by the test assigned development team member. Tracks progress during issue resolution.

No Activity	No work has been accomplished in the issue resolution.
In Progress	The Issue is being worked.
Resolved	The Issue has been resolved.

Table 3-15 Status attribute values for ISS requirement type.

3.13 Requirement Attributes for Assumption(ASM)

Requirement text defines the assumption.

Created

Date the Assumption was stated.

Status

Set by any team member. Minimizes repeated discussions or miss-directed expectations.

Stated	The Assumption has been documented.
Accepted	Team and business have accepted the Assumption.
Rejected	The Assumption has been rejected.

Table 3-16 Status attribute values for ASM requirement type.

3.14 Requirement Attributes for Term(TERM)

Requirement text includes a term and its definition.

Created

Date the Term and its definition were approved for inclusion in the Glossary.

Status

Set by any team member. Minimizes repeated discussions or miss-directed expectations.

Stated	The Term has been initially defined..
Accepted	Team and business have accepted the Term and its definition.

Table 3-17 Status attribute values for TERM requirement type.

3.15 Requirement Attributes for Business Rule(BR)

Requirement text includes the name of a Business Rule and its definition.

Created

Date the Business Rule was stated.

Status

Set by Systems Analyst. Minimizes repeated discussions or miss-directed expectations.

Stated	The Business Rule has been documented.
Accepted	Team and business have accepted the Business Rule.

Table 3-18 Status attribute values for BR requirement type.

4. Tracability Criteria

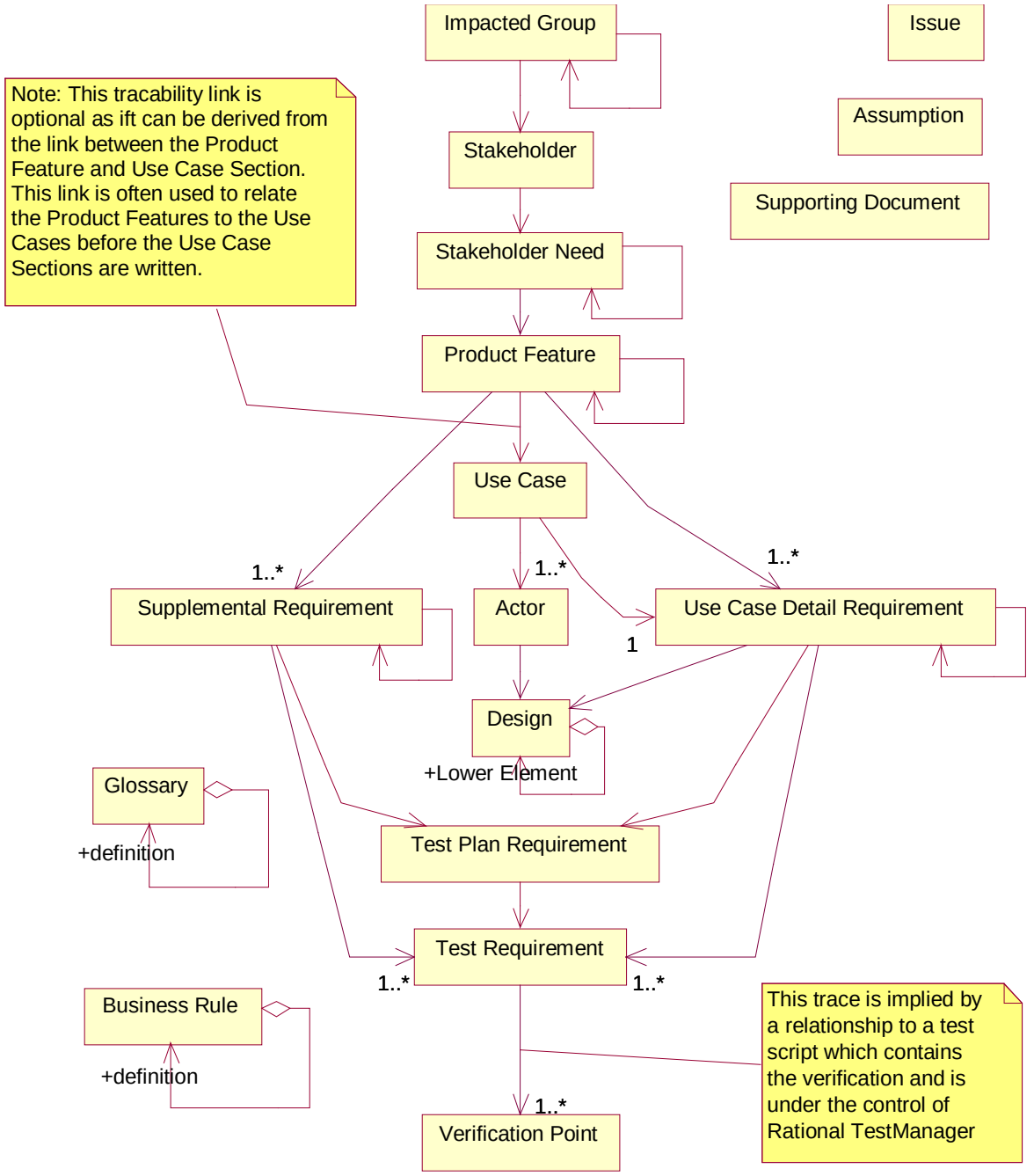

Figure 4-1 Requirements Tracability Diagram

4.1 Criteria for Impacted Group

The Impacted Group requirements are defined as database-only requirements in RequisitePro and will be traced to Stakeholder requirements when the Stakeholder requirement names an individual who is a member of or represents the Impacted Group. An Impacted Group requirement may trace to zero or more Stakeholder requirements.

An Impacted Group requirement may trace to zero or more Impacted Group requirements where the traced-to requirement represents a more detailed definition of the traced-from requirement (e.g. Store Associates traces to both Kitchen Cabinet Specialist and Check-out Associate).

4.2 Criteria for Stakeholder

The Stakeholder requirements are defined as database-only requirements in RequisitePro and will be traced to Stakeholder Need requirements when the Stakeholder requirement names an individual responsible and accountable for the problem or need represented by the Stakeholder Need requirement. A Stakeholder requirement may trace to zero or more Stakeholder Need requirements.

4.3 Criteria for Stakeholder Need Requirements

The Stakeholder Need Requirements defined in the *Vision Document*will be traced to Product Feature Requirements expressed in the same document. Each Stakeholder Need Requirement traces to zero or more Product Feature Requirements.

A Stakeholder Need requirement may also trace to other Stakeholder Need requirements where the traced-to requirement represents a contributing problem or need to the origin requirement.

4.4 Criteria for Product Feature Requirements

The Product Feature Requirements defined in the Vision Document will be traced to the corresponding Use-Case Requirements, Use-Case Detail Requirements and/or Supplementary Requirements in the Use-Case Specifications and the Supplementary Specification Documents.

Early in a project, tracing to Use-Case Requirements acts as a planning placeholder prior to the detailed tracing to Use-Case Specification Requirement. The significant trace from Product Feature Requirements is to Use-Case Detail Requirements.

Each Product Feature Requirement with a Status of Incorporated and a Target Release identified must trace to one or more Use-Case Requirement, Use-Case Detail Requirement and/or Supplementary Requirement.

A Product Feature requirement may also trace to other Product Feature requirements where the traced-to requirement represents a feature to which the origin requirement is reliant.

4.5 Criteria for Use-Case Requirements

The Use-Case Requirements defined in a RequisitePro database will be traced to the associated Actor Requirements defined in a RequisitePro® database and to Use-Case Detail Requirements defined in Use-Case Specification Documents..

Each Use-Case Requirement should trace to one Use-Case Detail Requirement with a Location attribute value of Name.

4.6 Criteria for Actor Requirements

The Actor Requirements defined in a RequisitePro database will be traced to the associated Design Elements defined in a a RequisitePro® database. Typically this trace points to an interface.

4.7 Criteria for Use-Case Detail Requirements

The Use-Case Detail Requirements defined in Use-Case Specification Documents will be traced to Test Plan Requirements defined in the Test Plan, Test Requirements defined in Test Case Documents, and Design Requirements defined in a RequisitePro database.

A Use-Case Detail requirement may also trace to other Use-Case Detail requirements where the traced-to requirement represents a system behavior expressed by that requirement to which the origin requirement is reliant.

Every Use-Case Detail Requirement must trace to one or more Test Requirements.

4.8 Criteria for Supplemental Requirements

The Supplemental Requirements defined in a Supplemental Specification Document will be traced to Test Plan Requirements defined in the Test Plan and Test Requirements defined in Test Case Documents.

A Supplemental Requirement may also trace to other Supplemental Requirements where the traced-to requirement represents a requirement to which the origin requirement is reliant.

Every Supplemental Requirement must trace to one or more Test Requirements.

4.9 Criteria for Design Element Requirements

A Design Element requirement may trace to other Design Element requirements where the traced-to requirement represents a lower level design element.

4.10 Criteria for Test Plan Requirements

The Test Plan Requirement defined in the Test Plan Document will be traced to Test Requirements documented in Test Case Documents.

4.11 Criteria for Test Requirements

The Test Requirement defined in Test Case Documents will be traced, implicitly to Verification Points, via Automated Test Scripts, Manual Test Scripts in TestManager.

4.12 Criteria for Issue Requirements

This Traceability Type allows you to add traceability items representing issues you want to track within RequisitePro. These issues can then be associated with whichever traceability items that they impact. An example of using the Issue traceability type would be to track issues associated with Glossary Items. If a definition is uncertain, or in dispute, issues could be raised and included in RequisitePro. This will ensure that the issue is not forgotten and allows a view to be built reporting on all Glossary Items with outstanding issues. Another good use of this traceability type is to track issues raised when reviewing the Use-Cases and other development artifacts.[11]

4.13 Criteria for Glossary Requirements

This is a trace between Glossary Terms and their definitions. You may choose to trace to the Glossary Requirement from any document, however, usually the trace is established between a Glossary Term within another Glossary Definition to the source of the Glossary term.

4.14 Criteria for Assumption Requirements

This Traceability Type allows you to track the Assumptions that you have made. The Assumptions can then be associated with whichever traceability items they affect.

4.15 Criteria for Business Rule Requirements

This is a trace between Business Rules and their definitions. The Business Reules can then be associated with whichever traceability items they affect.

4.16 Criteria for Supporting Document Requirements

This Traceability Type allows you to add any documents that you like into the traceability hierarchy. This is particularly useful for including pre-existing examples or documentation that clarifies the meaning or purpose of another traceability item. The flexible traceability mechanisms of RequisitePro allow you to associate supporting documentation with any traceability item of any type. An example of using the Supporting Document type is to include the detailed EDI message specifications as supporting information for the Glossary, or as appendices to the Use-Cases that will use the messages.[11]

5. Rational RequisitePro® Views

5.1 Attribute Matrix views for each Requirement Type limiting non-user attributes

5.2 Scoping View, see Target Release in 3.4

6. Rational RequisitePro® Extensions

6.1 Calculate Feature dependency count

6.2 Calculate requirement level counter

6.3 Calculate Technology Risk

6.4 Trace Check the requirements

6.5 Soda document for UCMS

Appendix 1: Technology Risk Assessment

Risk Factor		Weight
1. Which hardware, needed for the feature, is new to the comapany?		X5
None		0
CPU	High	3
Peripheral and/or additional storage	High	3
Terminals	High	3
Mini/Micro/CU	High	3
2. Is the system software (non-operating system) new to the IT project team?		X5
No		0
Programming Language	High	3
Database	High	3
Data communications	High	3
Other	High	3
3. How knowledgeable is the primary Stakeholder(s) in the proposed application area?		X5
Limited	High	3
Understands concept but has no experience	Medium	2
Has been involved in prior implementation efforts	Low	1
4. How knowledgeable is IT team in proposed application area?		X5
Limited	High	3
Understands concept but has no experience	Medium	2
Has been involved in prior implementation efforts	Low	1
Total		10-150

Table A-1 Technology Risk Assessment. [5]

Answer the questions for each feature, multiply the weight by the weight factor, in table A-1 the weight factor for all questions is five. Then total the weighted answers for the Technical Risk. Range 10-150. The Solution Center may want to revise this assessment with experience.

Appendix 2: Tracability Diagramming Notation

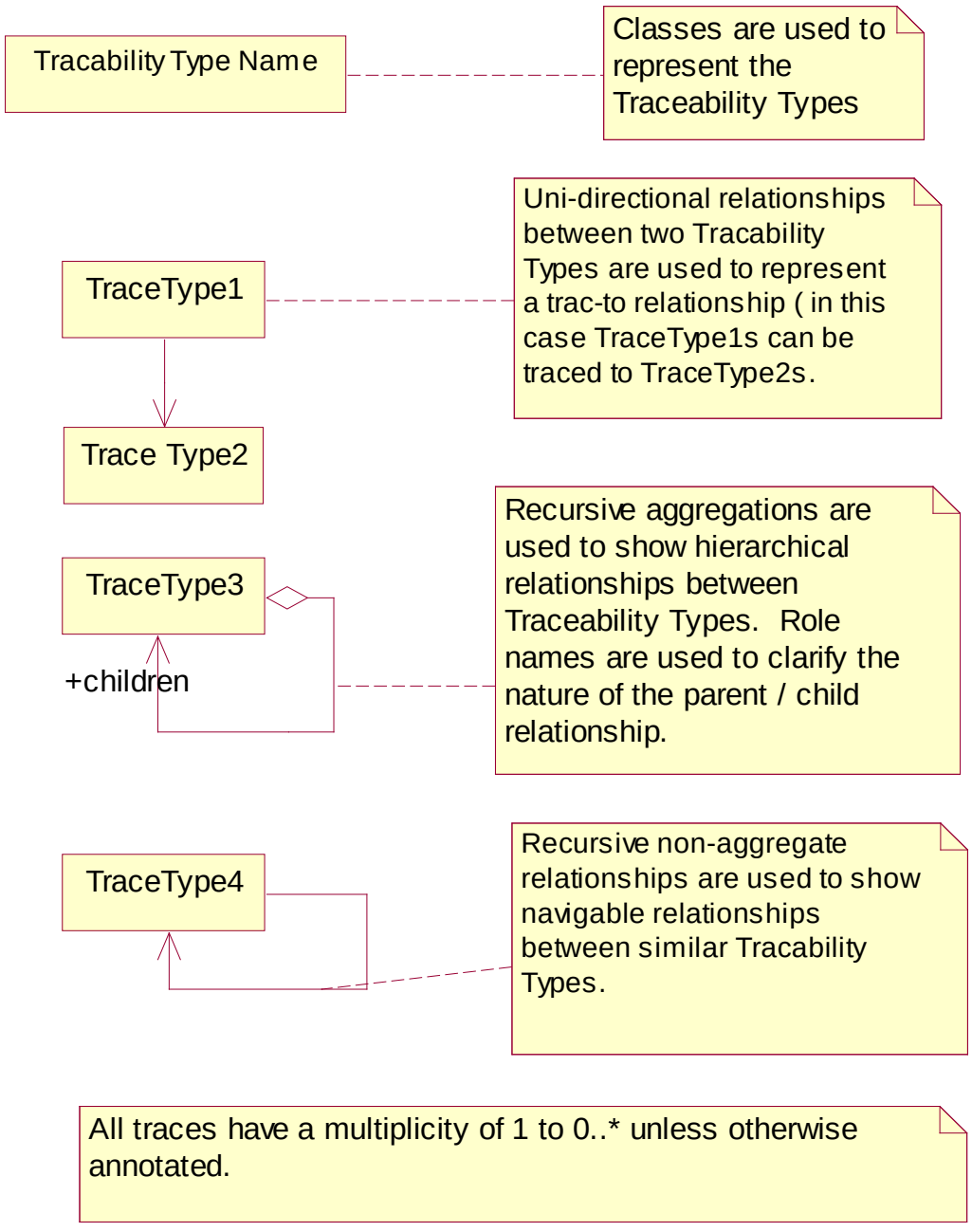

Figure A2-0-1 Tracability Diagramming Notations [6]