Requirements Management and Traceability

For IIBA

By Leslie Munday

About Me

Requirements Analyst

- Data Flow Diagrams
- Mil-Std-2167A
- Schlaer/Mellor
- UML
- RUP
- Scrum/Agile

Contents

- A bit about requirements
- Some definitions
- Define requirements management and traceability
- Trace models
- Some examples
- Personal experiences
- Q & A

Requirements

- Why do we bother with requirements?
  - So we know what we are doing
  - So we know why we are doing it
  - So we know if we are doing it right
  - So we know if we are doing the right thing
  - So that we can assess the costs to do the work
  - Etc ..

What Is A Requirement?

- A requirement is NOT just a line of text.
- A requirement is some 'thing' that allows us to verify that the system implementation is going to be 'correct'.
  - A requirement is verifiable.
  - It may be described by text.
  - It may be described with images.
  - It has a number of attributes that allow it to be verified, such as boundary conditions, states under which the requirement is in effect, input or triggering data, expected output or results, timing.
  - Example requirements specification languages Z, VDM, how about OCL?

Some Definitions

- Business need – A type of requirement that is solicited from the business not structured and without a validation method (a wish)
- System Requirement – Business needs that has been specified in terms of impact to a computing system (includes validation information)
- Implementation – The design and coding that satisfies a system requirement
- Deployment – The system as seen by an end user
- Verification – Confirmation that the implementation will satisfy the requirement
- Validation – Confirmation that the deployed system satisfies it requirements

Requirements Management

- Requirements management is the procedures and guidelines for ensuring that requirements are:
  - Satisfied – Did we capture all the business needs
  - Feasible – Can we do this within budget and resource constraints
  - Justified – Do we need to be doing this
  - Consistent – Are we doing this right
  - Testable – Are we doing the right thing
  - Etc (For example, organized such that we can find requirements).

Traceability Definition

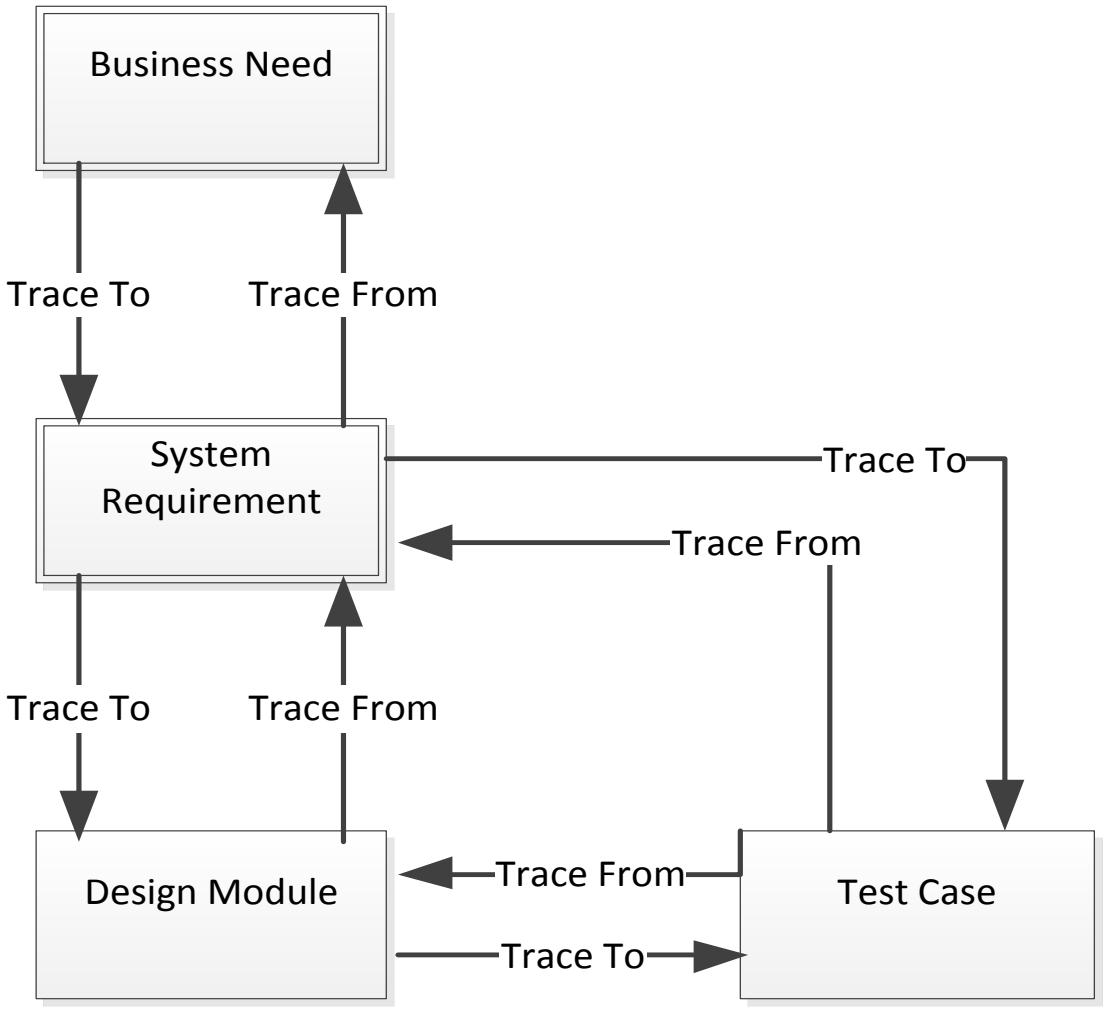
BABOK v2

- The ability to identify and document the lineage of each requirement, including its derivation (backward traceability), its allocation (forward traceability), and its relationship to other requirements.

What is Traceability?

- Traceability is the act of connecting a requirement to(↓), from(↑) or across(→), any other artifact produced during development, including:
  - Business need – Connects to system requirements
  - System Requirement – Connects to business needs, to design modules and test cases
  - Design module – Connects to system requirements and to test cases
  - Test case – Connects to system requirements and to design modules.

Simple Traceability Tree



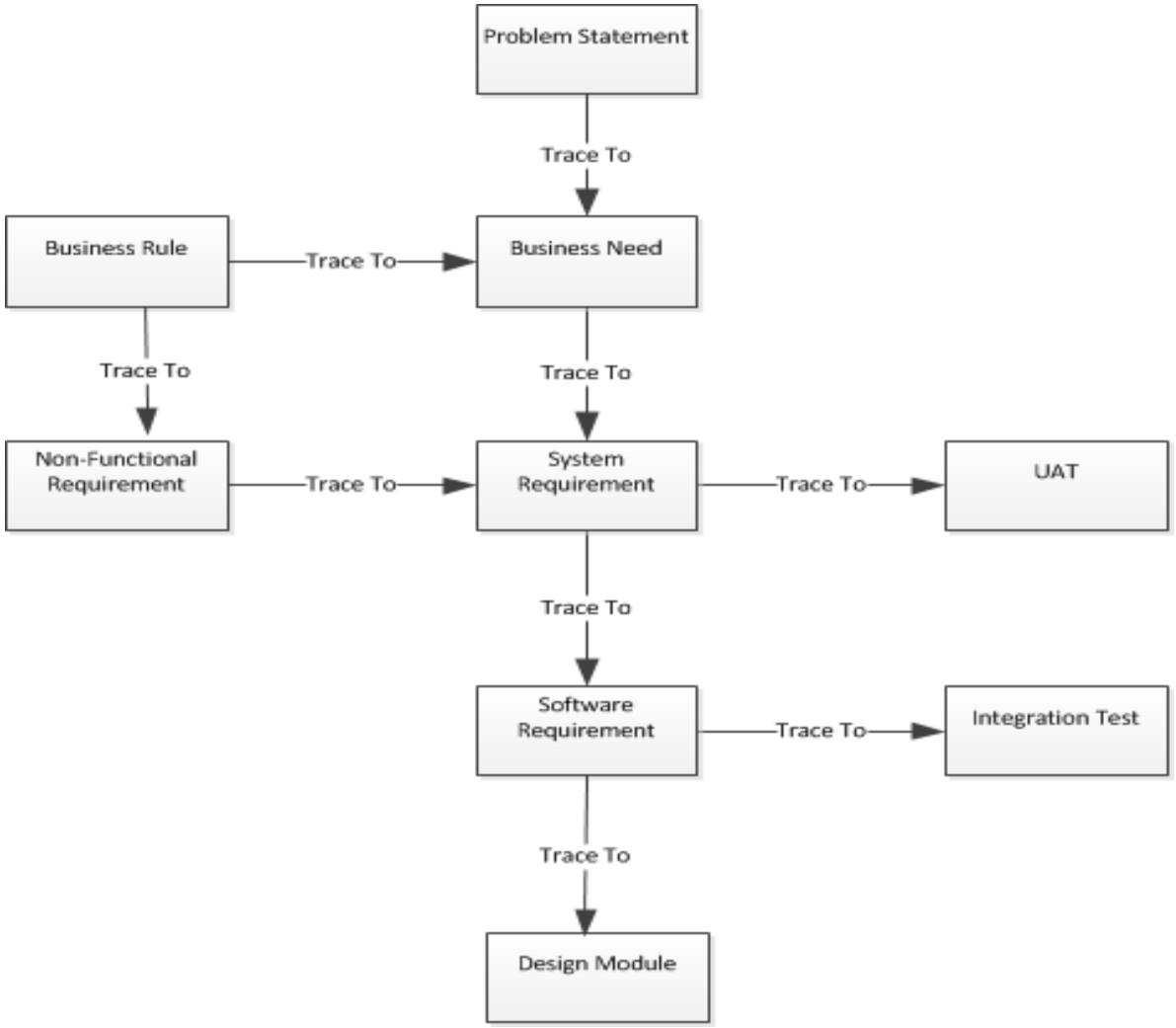
What Do We Get From Simple Traceability?

- Managing the lifecycle of a set of requirements.
- Allows us to identify the origin of each requirement.
- Tracks the reason for every change made to the requirement.
- Allows **justification** for the implementation and deployment of a requirement.
- Ensures that implementation and deployment **satisfies** the requirements.

Types Of Requirement

- Requirements come in many flavors:
  - Functional requirements – Describe a feature that can be observed over a period of time.
  - Non-Functional requirement – Impose restrictions upon functional requirements in terms of:
    - timing, security, business rules, hardware restrictions, technology constraints, design constraints, and many others.
  - Business Rule – Restriction on a business need.

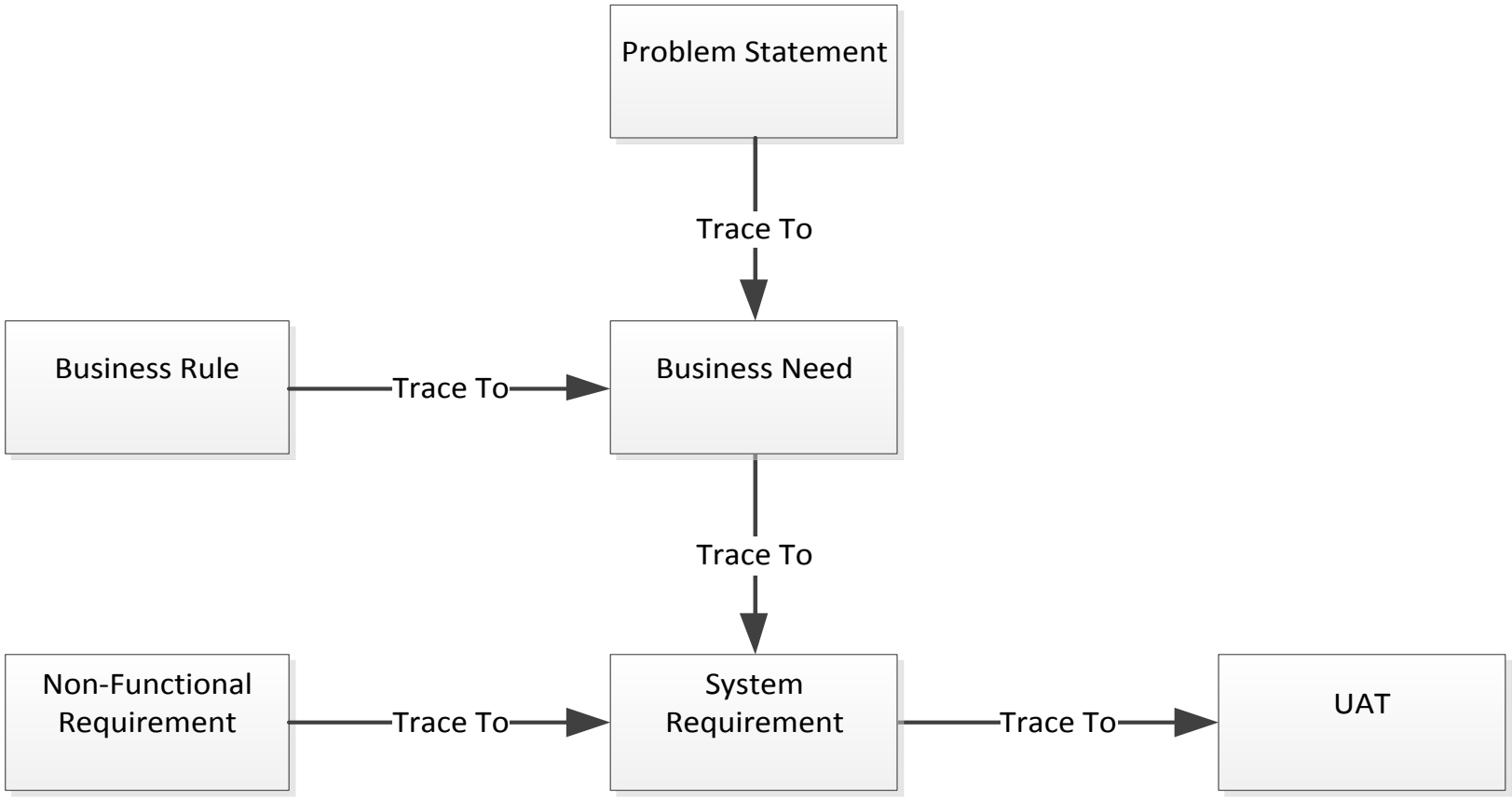
More Complicated Traceability Tree



What Can We Get From More Complicated Traceability?

- Using traceability to prove requirements are:
  - Justified – Trace the system requirements from the business needs
  - Satisfied – Trace the business needs to the system requirements
  - Feasible – Trace non-functional requirements and functional requirements
  - Testable – Trace the system requirements to the user acceptance test cases (user stories or use cases).

Realistic Traceability Tree



Tools

- Document (MS Word for example) – Commonly used to create the BFRD (Big Functional Requirements Document). Analysts start with a template and fill out each section with the relevant information. Traceability is performed by making reference to text in another section of the document.
  - Suitable for small projects with no more than ~20 system requirements, otherwise traceability becomes unmanageable and is abandoned before the document is complete.
- Spreadsheet (MS Excel for example) – Is an improvement on a text document in that each requirement can be managed as a separate entity and traced to its location in the spreadsheet. Clutter is removed. Easier to manage requirements and reference them as they a moved.
  - Typically where the business needs are located. May be suitable for projects up to ~100 system requirements, but will become unmanageable since links are managed manually and there is no mechanism for automatically tracking broken links.
- Requirements Management Tool (ReqPro or DOORS for example) - Using RM tools adds automation to spreadsheets and provides reporting capabilities for tracking broken or suspect traceability links.
  - Good for medium sized projects up to ~1000 or so system requirements, after which the manual maintenance becomes to great and traceability is gradually abandoned.
- Modeling Tools (Rational, Enterprise Architect for example) - Allows graphical representation of traceability between requirements elements and design elements. Also test cases can be represented and linked via an interface to the testing tool.
  - Excellent way to represent traceability if the requirements are going to be modeled and the model is going to be maintained throughout the life of the project.
- Online Shared Repository (SharePoint or Rally for example) – Much more flexible than Excel, but tool setup configuration and some maintenance is required.
  - Used for agile projects where the requirements are textual based and a simple traceability model is used.

Example With Word

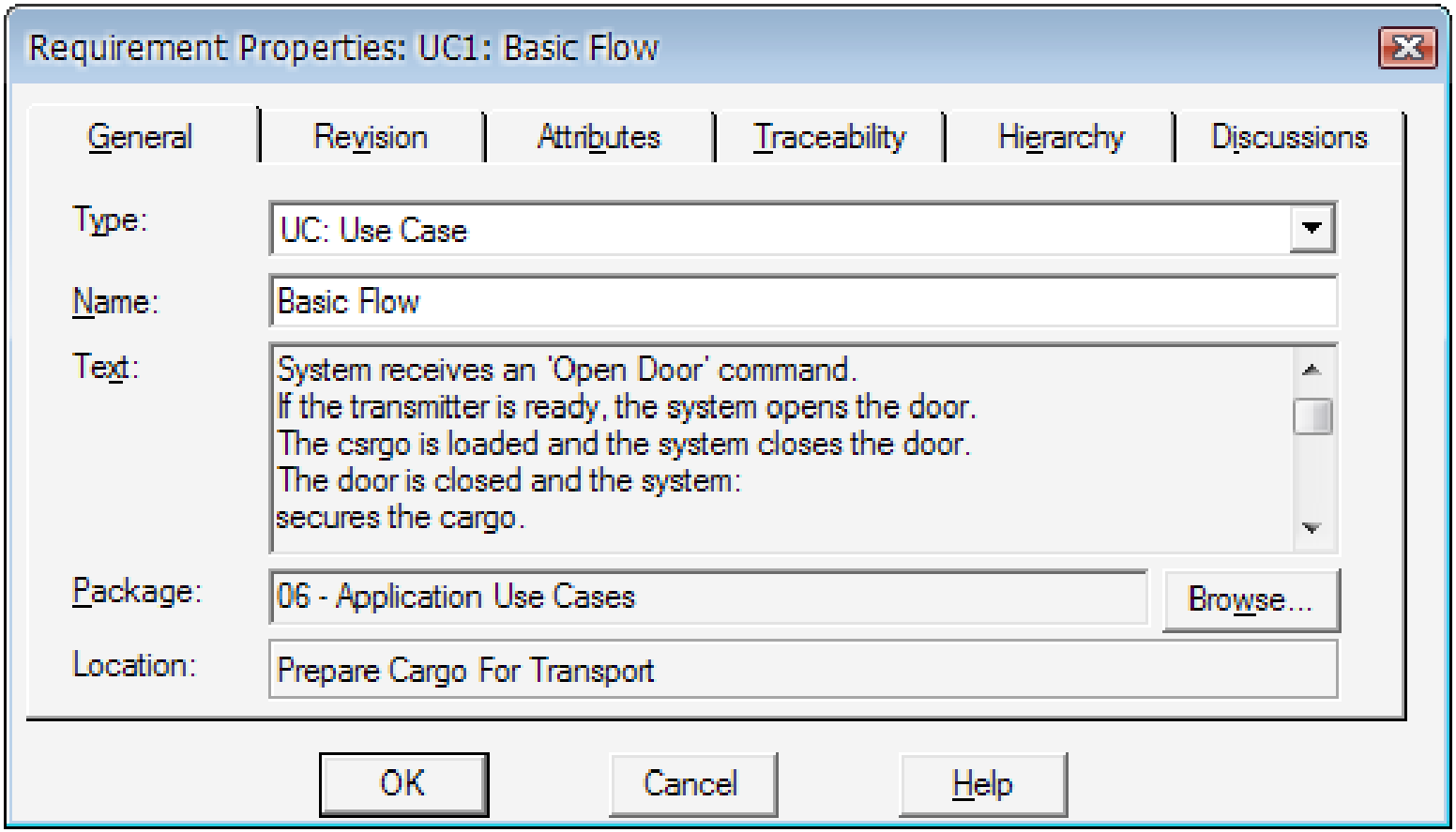
1) Business Needs
    1. BN#1: *business need text*
    2. Etc..
2) System requirements
    1. SR#01: system *requirement text* [BN#1]
3) Non-Functional Requirements
    1. Performance
    2. Security
        a. NFR#01: *non-functional requirement text* [SR#01]

Example Using Excel

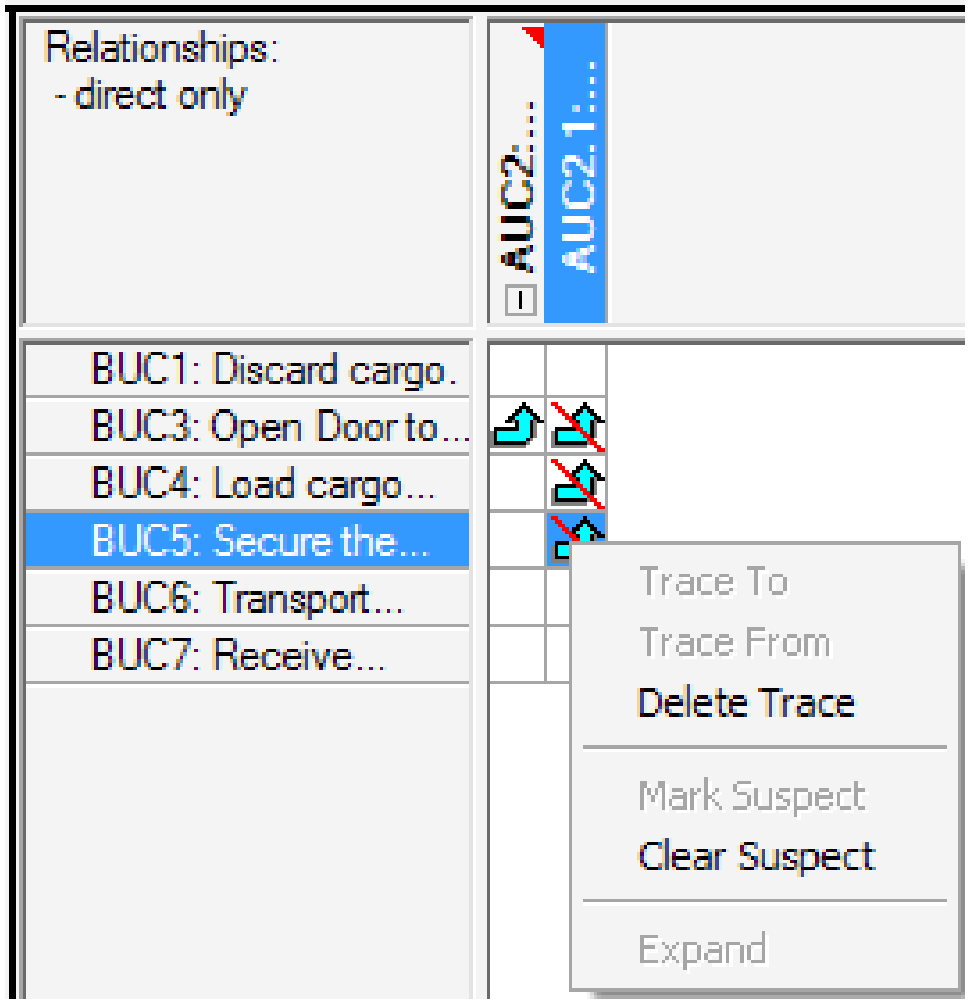
Requirement Id	*Requirement*	Trace From
BR:01	*Business requirement text*	PS:1
SR:001	*System Requirement Text*	BR:01
NFR:001	*Non-functional requirement text*	SR:001

Test Case Id	Description	Trace From
Test Case #001	Descriptive text	SR:001

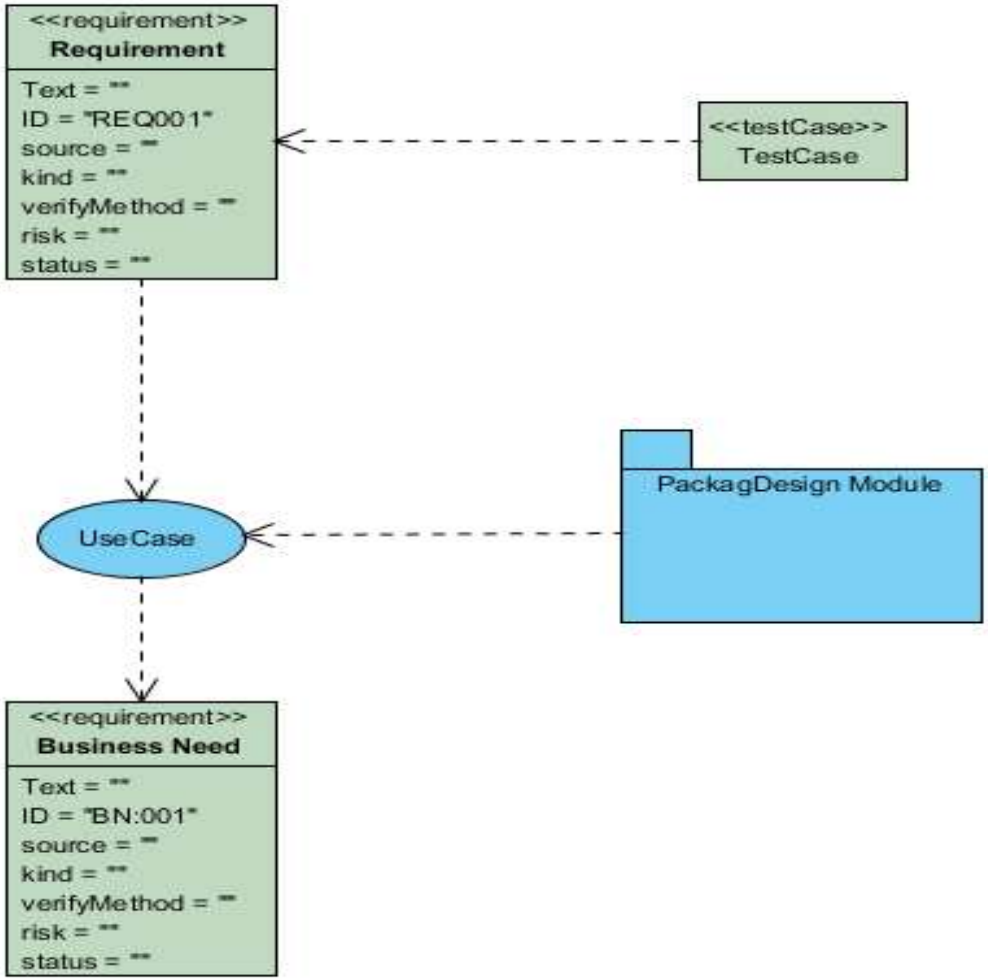
Requirements Management Tool (ReqPro)



ReqPro Traceability Matrix



Diagraming Traceability



SharePoint Example

Why Is Traceability So Hard?

- Requirements do not trace 1 to many as we go down the tree. 1 requirement may trace to many parent requirements. (Rally.)
- Requirements change, traceability breaks.
- Manually intensive, even with the most complex of requirements management tools.
- Traceability is a guessing game, it is never perfect.
- Not understanding why we are performing traceability, why are we tracing this requirement?
- More important project tasks. Traceability gets left behind and never catches up.

Summary

- Traceability is about linking one item to another to show a relationship (specifically one of those items is a requirement)
- Traceability shows requirements that have no parent business need (not justified)
- Traceability shows business needs that have no requirements (not satisfied)
- Traceability allows a supplementary (non-functional) requirement to be linked to several functional system requirements, hence removing duplication.
- Traceability links from requirements to test cases ensure correct validation of the requirements
- Traceability links ensure that when a requirement changes, that all impacted items are identified. (Justify the cost to the business of making a change to the requirements.)

Recommendations

- Create a traceability tree model and enforce it. (Do not allow tracing between requirement types not specified in the model.)
- Identify a level of detail to which it is necessary to trace requirements. (Epics vs user stories, Scrum example. Or use cases vs actions, RUP example.) Packaging requirements is a very simple way to maintain traceability.
- Understand the benefit of each trace type in the model.
- Understand the cost of maintaining the traceability model.
- Is it necessary to maintain the traceability model throughout the whole lifecycle of the project, or is the a point at which diminishing returns on benefit make traceability no longer cost-effective?
- Is it necessary to trace every requirement, or just the important ones?
- Manual traceability is laborious and prone to mistakes. Automate the process as much as possible.

References

- Analysis Through Picture (Chapter 19)
- Introduction to RequisitePro
- Rally traceability links
- Rational DOORS Next Generation

Discussion

- Space Station
  - Used document references and spreadsheets.
- AT&T
  - ReqPro used to capture textual requirements. Integration links to Test Director.
- Rally at T-Mobile
  - Rally captures user stories and business needs. Links to test cases.
- MS Word at Regence
  - Moved from BFRD to use case documents and ReqPro.
- SharePoint ..
  - In the absence of any requirements management tools, SharePoint is easily configured for simple requirements traceability.