UNIVERSITY OF
NEWCASTLE

University of Newcastle upon Tyne

COMPUTING SCIENCE

Tailoring Traceability Information to Business Needs

P. Arkley, S. Riddle and T. Brookes.

TECHNICAL REPORT SERIES

No. CS-TR-945 Febuary, 2006

TECHNICAL REPORT SERIES

No. CS-TR-945 Febuary, 2006

Tailoring Traceability Information to Business Needs

Paul Arkley, Steve Riddle, Tom Brookes.

Abstract

Several surveys over the past 20 years have observed that traceability information is often poorly recorded. In previous work we have argued that this is a result of many requirements traceability systems failing to provide any direct benefit to the development process. In this paper we describe an application of traceability by a company, with a resulting increase in profitability as well as other benefits for the development engineer, project management and customer.

Printed and published by the University of Newcastle upon Tyne,
Computing Science, Claremont Tower, Claremont Road,
Newcastle upon Tyne, NE1 7RU, England.

Bibliographical details

ARKLEY, P., RIDDLE, S., BROOKES,T..

Tailoring Traceability Information to Business Needs
[By] P. Arkley, S. Riddle and T. Brookes.

Newcastle upon Tyne: University of Newcastle upon Tyne: Computing Science, 2006.

(University of Newcastle upon Tyne, Computing Science, Technical Report Series, No. CS-TR-945)

Added entries

UNIVERSITY OF NEWCASTLE UPON TYNE
Computing Science. Technical Report Series. CS-TR-945

Abstract

Several surveys over the past 20 years have observed that traceability information is often poorly recorded. In previous work we have argued that this is a result of many requirements traceability systems failing to provide any direct benefit to the development process. In this paper we describe an application of traceability by a company, with a resulting increase in profitability as well as other benefits for the development engineer, project management and customer.

About the author

Paul Arkley joined the CSR in November 2001 as a member of the BAE SYSTEMS Dependable Computing Systems Centre. He previously taught high school science and in 1986, he left teaching to work for British Aerospace Dynamics, Air Weapons Hatfield. Whilst working for BAe/SEMA on the Trident and Type 23 Frigate command and control projects, Paul obtained a Masters in Computing Science from Birbeck College University of London (1992). In 1992, he left BAe/SEMA for Eurocontrol (Brussels) to work on systems for flight plan coordination. He returned to the UK in 1994, to work as a freelance software consultant for the following companies, GPT, British Gas and Accenture. In 1999, he joined Sun Microsystems to work on the software support for fault tolerant and thin networks servers. The experiences gained on these projects enabled him to obtain Chartered Software Engineer status from the Engineering Council in 1998.

Dr Riddle worked as a Research Associate on the DCSC (1997 - 2001) and on the EU COPERNICUS project ISAT (Integration of Safety Analysis Techniques for Process Control Systems) from October 1995 - May 1997. He obtained his BSc in Computer Software Technology at the University of Bath in 1991, and went on to complete a PhD at Bath concerning the use of partial specifications and formal refinement theory to aid the process of explaining complex systems.

Suggested keywords

REQUIREMENTS
TRACEABILITY
DEVELOPMENT PROCESS

Tailoring Traceability Information to Business Needs

Tom Brookes
BAE SYSTEMS, Electronics and Integrated Solutions, Plymouth, UK
Tom.Brookes@baesystems.com
Paul Arkley, Steve Riddle
School of Computing Science, University of Newcastle upon Tyne, NE1 7RU, UK
{Paul.Arkley, Steve.Riddle}@ncl.ac.uk

Abstract

Several surveys over the past 20 years have observed that traceability information is often poorly recorded. In previous work we have argued that this is a result of many requirements traceability systems failing to provide any direct benefit to the development process. In this paper we describe an application of traceability by a company, with a resulting increase in profitability as well as other benefits for the development engineer, project management and customer.

1 Introduction

Several surveys over the past 20 years have observed that traceability information is often poorly recorded [1] [2], yet if performed correctly it can be a valuable activity which is beneficial to the development engineer, manager and customer alike. We have previously argued that the problem of poor Requirements Traceability practice is not related to tools or their related technologies, rather it is dependent upon the use to which the data is put, and the stakeholders who benefit from that data. We have referred to this as the Traceability Benefit Problem [3].

As part of a Requirements Traceability practice survey we were fortunate to observe a project, conducted at BAE SYSTEMS E&IS (Plymouth, UK) that tackled the Traceability Benefit Problem. The E&IS (Electronics and Integrated Solutions) Operating Group designs, develops and manufactures a wide range of electronic systems and subsystems for both military and commercial applications.

These observations were enhanced by an opportunity, due to a tool vendor change, to collect requirements and testing metrics over the duration of that project. Such data can be difficult to obtain and is not often published in the public domain, so this was a welcome opportunity to capture this data and compare with predictions in the literature.

In this paper we describe how the project tackled the Traceability Benefit Problem by developing a Requirements Traceability system which is integral to their development process. A description of the product is not the focus of this paper: rather, we concentrate on demonstrating how this project exploited traceability information. We begin by describing the development processes, and introduce the traceability data model used by this project (Section 2). We then illustrate and describe the use of this model (Section 3) with data which was collected during a transfer to a new tool implementation which occurred in 2002-3. We then examine the specific benefits that were observed for this system (Section 4) and draw some conclusions, by comparison to other empirical results (Section 5).

2 Tailoring Traceability Information to Business Needs

The project produced variants of a common product which was developed and maintained by a small team (four to five) of specialist engineers. Each engineer had an assigned role, though they could be called upon to change their roles.

The approach taken by this project in developing a traceability system was to examine the uses to which traceability data was to be put, and for whom the data was being produced. This was achieved by examining the *business needs* required to support the engineers in their tasks. For example, the production of a new variant requires the engineer to identify the parts of the design that are required to be changed. The following broad business needs and associated traceability data and relationships were identified:

1. To show that all of the customer requirements have been proven. To achieve this, traceability relationships from the requirements to test procedures and related test results are recorded.
2. To identify which parts of the design are required to be changed to produce a product variant. To achieve this, traceability relationships are recorded from the requirements to design
3. To record the justification for design decisions. To achieve this, traceability relationships between requirements and the design decisions are recorded.

These needs drove the design of the product requirements traceability model (Figure 2-1) and the supporting engineering process. The transfer of this data model from RTM to a DOORS implementation afforded an opportunity to collate requirements and test metrics. The development process supported by the data model is described below.

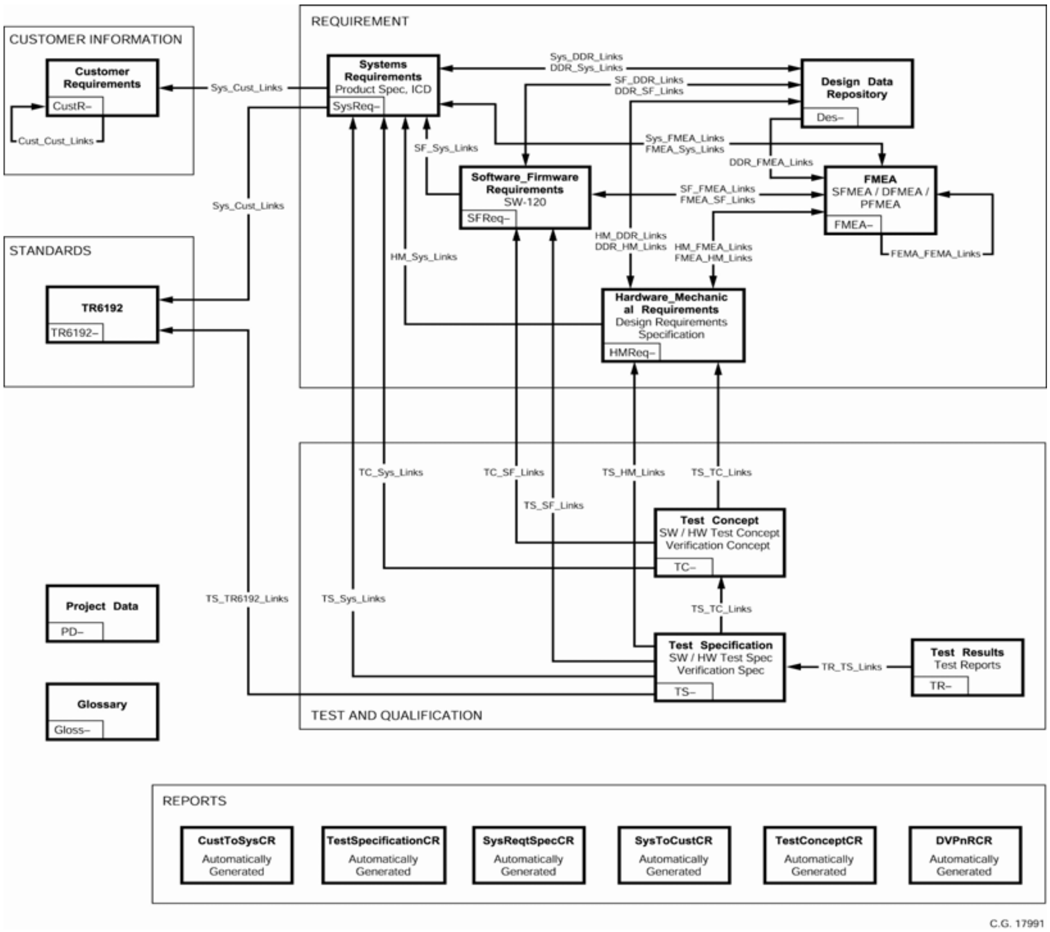

Figure 2-1 Traceability Data Model

2.1 Development Process

The process consists of the following phases.

Prepare Inputs for a Proposal

This development phase is concerned with capturing and reviewing the customer's requirements, and is divided into a number of tasks. The first task is to identify and capture requirements from the customer's documentation. Any queries on the requirements are recorded and raised with customer. The requirements are reviewed with respect to compliance with existing products, and from this information a compliance matrix (requirements vs. current product) is generated. This matrix provides a basis for selecting the most suitable product to be a basis for the customer's new variant, and gives an indication of the extra work required to produce the new variant.

Manage, Analyse, Develop System Requirements

This phase starts once a contract has been signed and continues throughout the entire lifecycle of the product. The model and supporting processes are employed to control the introduction of new requirements and the modification of existing requirements to prevent "Requirements Creep". The main task of this phase is the development of the customer's new requirements. For each of these requirements, a development risk grading is assigned and a verification method is identified and recorded. The compliance matrix is regenerated to ensure that the new product complies with the customer's requirements and that there is a verification method for each requirement.

Design

The design phase enables the recording of design decisions, the member of staff who made the decision and how it is related to the requirements. A Failure Modes and Effects Analysis also allows the recording of failure mode estimations and how they relate to design decisions. The accepted level for the total failure mode estimation is specified by the customer. This recorded design information traceability enables the assessment of the impact of any change in requirements.

Prepare Test and Qualification Procedures

The data model enables the recording of relationship between requirements and test descriptions. A test description is further decomposed into number of test cases, which are in turn decomposed into a number of test steps. The test steps information describes in detail the nature of the test and the expected result. From the information in the traceability system it is possible to generate a document which describes the test description required to qualify a product against the customer's requirements.

During the qualification phase of the program, the defined tests are run and the results recorded. Having traceability of all the information from customer requirements through to the test results in the data model allows the verification information to be generated quickly and accurately. Any missing verification evidence is clearly identifiable. This allows rapid sell off of the completed product. In addition, as verification evidence is accrued during the project lifecycle, the degree of compliance to the original requirements can be closely monitored by management to ensure that the project remains on track for success.

3 An Illustration of the Traceability System

Having described the traceability data model and development process, this section illustrates the benefits obtained from traceability by describing the development history for a product variant. Figure 3-1 outlines the development history for a product variant which was undertaken in 2002 - 2003.

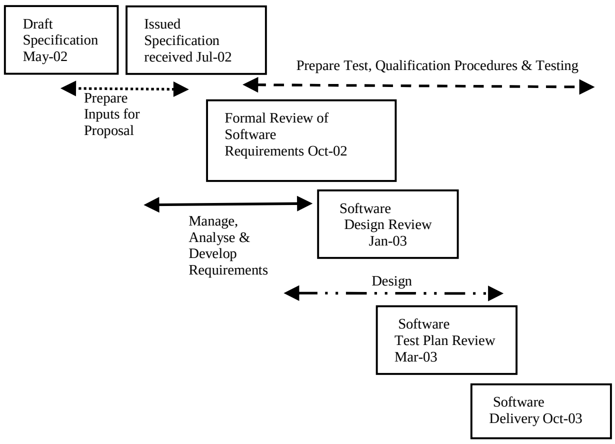

Figure 3-1 Software Development History: Milestones and Phases

3.1 Managing Requirements and the Customer

The development of the new product variant was initiated in May 2002, when the customer's initial draft contractual specification was received. This specification was compared against the existing specifications and related software requirements which were recorded the traceability system. Within a month of receiving the specification, the traceability system enabled the development engineers to answer such questions as *"what is the same, what has changed and what is new?"* with respect to the product software requirements. This analysis resulted in an initial breakdown of 40% new requirements, 40% unchanged and 20% requiring minor modification (shown in Figure 3-2). This information provided the engineers and their management with an indication of the potential work required to produce this new variant. During this period the engineers employed the traceability system to record any issues related to the specification.

In July 2002 the customer issued the final version of contractual specification. The traceability system allowed the engineers to determine which system requirements were affected by the final version, which resulted in a substantial increase in new functionality (shown in Figure 3-2). Again, this information was employed in the revision of the estimated cost of producing the new product.

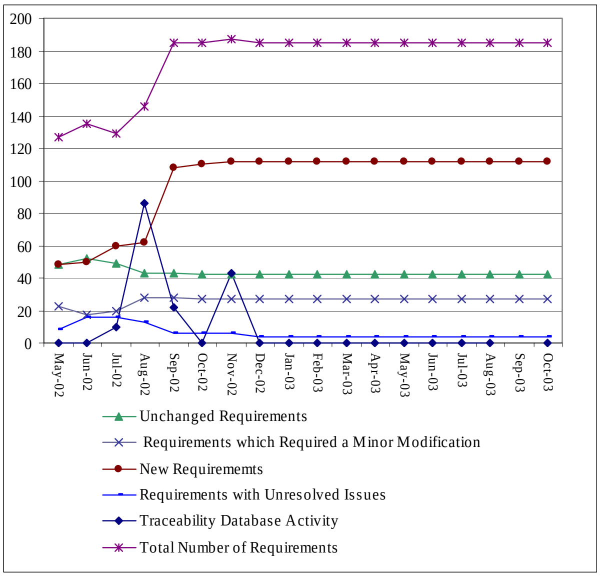

Figure 3-2 Requirements Activity during Project Lifetime

In October 2002, a formal review of the requirements was undertaken with the customer. The traceability system was employed to generate a report which demonstrated how each item in the specification was satisfied by the requirements and how it would be proved by high level qualification tests. A peer group review was undertaken of the documents by the customer's representatives and the development engineers. Once the changes resulting from the peer group review were agreed, the specification, requirements and qualification test were frozen for that release of the software.

3.2 Quantitative Management

During requirements analysis phase the traceability system allowed the rate of change for each requirement to be calculated. This information allowed the engineers and their management to determine a time when the requirements where stable enough to progress to the requirements peer review and commence the design phase.

During the design phase, the recording of relationships between design items and requirements enabled management to estimate the progress. Similar quantitative progress metrics were obtained in the testing phases by determining the rate at which trace relationships were recorded between tests, test case and test steps and the requirements.

The traceability system provided the most important progress metric, that of determining when the development of the software was complete. This was determined by demonstrating that all the requirements had been tested by following the trace relationships from the requirements to tests, test cases and finally to each test step and the associated validated test result. Only when this could be demonstrated would the customer accept the product.

3.3 Component Reuse

As previously mentioned, the traceability systems enabled the engineers to determine which development components (requirements, design elements and tests) could be reused. The process of comparing the customer's specification with previous specifications enabled the engineers to determine which requirements could be reused. The identification of these requirements resulted in the identification of related design elements, validation test, test cases and test steps which could also be reused. For example, it was found that 65% of the existing high-level tests could be reused with at most a minor change (Figure 3-3). This level of reuse remained constant, indicating that the engineers had accurately identified the tests that could be reused at the beginning of the project.

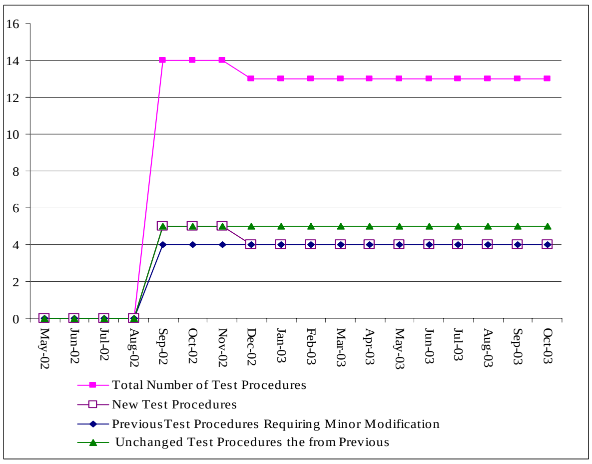

Figure 3-3 Test Procedures

In a similar fashion, the traceability system enabled the engineers to determine which test cases could be reused (Figure 3-4). It was found that 54% of test cases could be reused with minor changes or unchanged. Again this level of reuse remained constant indicating that the correct test case had been identified at the beginning of the project.

The accurate identification of which components can be reused by the traceability system improves the efficiency of development process and therefore reduces development costs. This in turn reduces the overall business risk. This is seen as a major benefit by the development engineers and their management.

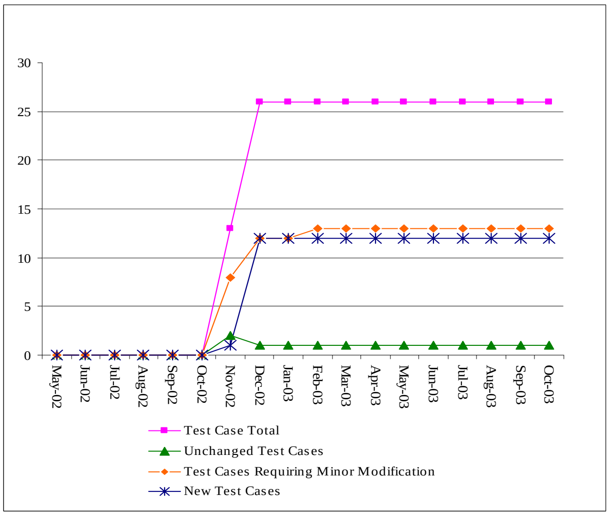

Figure 3-4 Test Cases

4 Why is this System Successful?

The development engineers and their management at BAE SYSTEMS E&IS did not consider their system to be a hindrance to the development process. In fact they considered the system to be at the heart of their development process. We examine the reasons for this by considering the system from three viewpoints.

4.1 The Development Engineer's View Point

Though this traceability system required manual effort to transcribe and enter data, these exertions were outweighed by the benefits the system afforded to the engineers.

The traceability systems allowed the engineers to coordinate and control changes to their requirements. The trace relationship between the customer's specifications and the requirements enable the development engineers to determine and negotiate the impact of any changes to specification change. These trace relationships were employed to generate documentation which formed part of the procurement contract. These documents bound the engineers and customer together, resulting in the reduction of *requirements creep* (introduction of new requirements) and the elimination of *"over the wall"* (customer imposed) changes.

The ability of the traceability system to allow the engineers to identify which development components (requirements, design items and tests) could be reused was also seen by the engineers as a way of improving their efficiency.

4.2 The Manager's View Point

Though establishment and maintenance of the traceability system required project budget, these cost were justified by the project management. The ability to identify development components which could be reused resulted in a perceived reduction in project risk. The identification of these components also enabled the project management to make better estimations of the production costs. The traceability system allowed management to estimate progress by rate of creation of traceability links between development artifacts.

4.3 The Customer's View Point

The traceability system does not directly increase the cost to the customer, but the presence of the system is beneficial to him. The main benefit to the customer is a demonstration of how the requirements related to his specification and how the product will be tested to demonstrate compliance with the specification.

To summarise, the system provides development engineers with a stable, efficient environment, management with reduced risk and improved cost estimates, and customers with a clear link from their specification to compliance tests.

5 Conclusions and Related Work

We have presented an overview of a successful traceability system, using empirical data describing the changes to requirements over the project lifetime. It is unusual to have the opportunity to collect empirical requirements data, though in related work studying requirements evolution Felici collated data for software releases of an Avionics System [4]. Our work is at a finer level of granularity, as Felici was working at the level of software releases rather than individual requirements. Nevertheless it is interesting to apply his metrics to our data.

For example, Felici shows the application of a Requirements Maturity Index (RMI) to his data to demonstrate the maturity of each software release. Looking at the RMI for our data, it is possible to observe the growth in maturity of the requirements as the project progress (Figure 5-1).

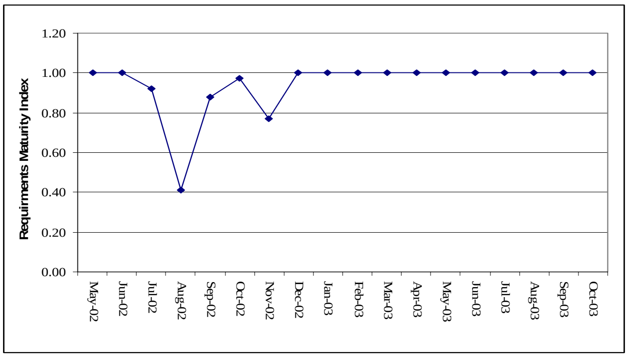

Figure 5-1 Requirements Maturity Index

In Figure 5-1 we see two dips. The first occurs in August 2002 and the second, smaller dip occurs in November 2002. This can be compared with the requirements activity shown in Figure 3-2, and with the project history; in July 2002 the specification was issued and in October 2002 the requirements were reviewed. Both events resulted in an increase the reworking of the requirements and hence a temporary fall in the RMI.

This traceability system is tailored to the local development process. A challenge for future work is to generalise the characteristics which make this development process/traceability systems successful in order that they can be applied to other development processes and products.

6 References

[1] B. Ramesh, C. Stubbs, T. Powers, and M. Edwards, "Implementing requirements traceability: a case study", in *Proceedings 2nd IEEE International Symposium on Requirements Engineering*, York, England, March 27-29, 1995, pp 89-100.

[2] O. Gotel and A. Finkelstein, "An Analysis of the Requirements Traceability Problem", in *Proceedings First International Conference on Requirements Engineering*, Colorado Springs, Co, 1994. pp 94-101.

[4] P. Arkley and S. Riddle, "Overcoming the Traceability Benefit Problem" in *Proceedings 13th IEEE International Requirements Engineering Conference*, Paris, France 2005, pp 385-389.

[5] F. Massimo, PhD thesis, *Observational Models of Requirements Evolution.* School of Informatics, University of Edinburgh 2004.