12-5-2006

Managing Information Security Complexity

Murray Brand
Edith Cowan University

DOI: 10.4225/75/57b6562e34766
4th Australian Information Security Management Conference, Edith Cowan University, Perth, Western Australia, 5th December, 2006

Managing Information Security Complexity

Murray Brand
School of Computer and Information Science, Edith Cowan University,
Bradford Street, Mt Lawley, Western Australia 6050
mbrand0@student.ecu.edu.au

Abstract

This paper examines using a requirements management tool as a common thread to managing the complexity of information security systems. Requirements management provides a mechanism to trace requirements through to design, implementation, operating, monitoring, reviewing, testing, and reporting by creating links to associated, critical artefacts. This is instrumental in managing complex and dynamic systems where change can impact other subsystems and associated documentation. It helps to identify the affected artefacts through many layers. Benefits to this approach would include better project planning and management, improved risk management, superior change management, ease of reuse, enhanced quality control and more effective acceptance testing. It would also improve the ability to audit, especially at a time when outsourcing of security functions is occurring throughout the world. ISO 27001:2006 provides a model for the implementation of an Information Security Management System (ISMS) that can be tailored by an organization. It is proposed that employment of a requirements management tool could manage the traceability aspects of an ISMS.

Keywords

Information Security Management System, Requirements Management, DOORS, Assurance, ISO 27001.

INTRODUCTION

There is no doubt that information security is a complex and dynamic discipline, and its scope and associated knowledge domain covers a very broad spectrum. When information security is outsourced, globalized or outnationed, traceability from the implementation and operation to user requirements becomes even more complex. The purpose of this paper is to outline how the principles of traceability, leveraged from requirements engineering, and extended to information security management through the use of a requirements management tool, can help to administer this complexity and assist in the functionality of an Information Security Management System (ISMS) as outlined by ISO 27001:2006. This paper proposes a way of navigating through the complexity by using a requirements management tool such as Telelogic's Dynamic Object Oriented Requirements System (DOORS) product. Benefits of requirements management discussed by Dick (2005, p.5) include:

- Greater confidence that objectives are being met.
- Ability to manage change.
- Improved customer / supplier relations.
- Ability to track progress / status.
- Ability to save costs through cost / benefit analysis.

It should hardly be surprising that business requirements change regularly. New business relationships are formed, internal structures are changed, business strategic focus shifts to new markets, technology advances, functionality and responsibilities are outsourced, and new regulations are introduced. As a result of business requirement changes, policies must be reviewed and updated through a formal and procedural testing process. Changes to policy cascade down through to risk assessments, security controls, designs, procedures, guidelines, and test cases. Without a management tool, it may not be clear what needs to be changed, what the artefacts of

the change are, who authorised the change and what the effective results of the change are. At the very least, it may be labour intensive to determine and an inefficient use of resources. If a business decision has been made to outsource, or outnation a security function, it would be desirable to be able to trace the business rule to the outside organisation's security documents, assessments, test results and reports. This would give a measure of confidence and assist in ensuring compliance with security policy and regulations. Policy for example, must be used as one of the inputs to the design of the network topology and the development of firewall rule sets. The design and implementation must be traceable and testable against the policy documents. Otherwise, the resultant firewall may not meet the requirements and specifications in policy. The rule sets must be under configuration control to ensure any changes are validated and are traceable back to policy. From the other point of view, if a subsystem or asset is found to be obsolete, or beyond repair, the initial requirements that initiated its introduction to the system may need to be re examined to find a suitable replacement. This change may also need to be reflected in updates to procedures, guidelines and network diagrams. In the event of a security incident, it will be beneficial to have quick access to relevant documentation to help to determine how the incident occurred and how to prevent it occurring again. Again, this will likely require changes to relevant documentation in the fastest time possible. It may also be required to look at the artefacts of the design process, the authorisations and the qualification process. Another particular problem is base lining. In the event of an incident, investigation or audit, it may be required to show the baseline of the security documentation at any particular time in the past. There may be a large number of security documents, all under version and configuration control. It could take an inordinate amount of time to recreate the baseline manually. It would be useful to be able to recreate the baseline for a particular point in time in the fastest way possible using a tool. The common thread to the discussion above is requirements engineering.

Requirements Engineering

Requirements are the common factor in all phases of the life cycle of any system, from the identification of a need, conceptual design, preliminary design, detailed design and development, implementation, operational use and through to disposal. They are the common language that all the stakeholders in the system use to communicate to each other. This communication assists in being able to facilitate the understanding of the need, fosters stakeholder agreement, and clearly defines business objectives. This methodology is well accepted within the realms of system analysts, system engineers and software engineers as best practice.

- Hull, Jackson and Dick (2002, p.2) list requirements as forming the basis for the following:
- Project Planning – Tasks can be broken down, delegated and tracked.
- Risk Management – Risks can be identified very early in the project and have a risk management strategy assigned.
- Acceptance Testing – Test cases can be developed as soon as the requirements are written and accepted.
- Trade Off – Tradeoffs that could be considered could include reliability versus maintainability, availability versus confidentiality and so on.
- Change Control – Requirements can be base lined and put into version control. Proposed changes can be vetted, and the impact of change determined through traceability mechanisms.

Requirements are typically broken down into lower levels at each phase of the life cycle and have associated test components. Typically, this is communicated in the "V" model as shown below in figure 1. It shows that requirements in each layer link to statements in the layer above and below, and that the requirements have associated test phases at the same level.

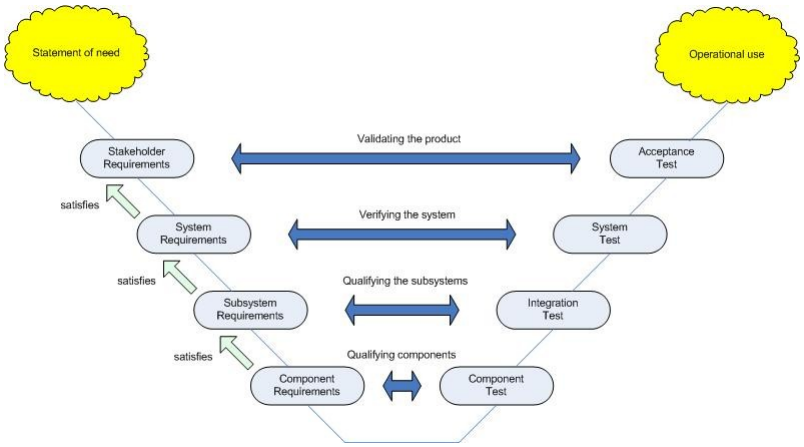

Figure 1. "V" Model. Adapted from Hull et al. (2002, p.9)

Traceability is evident with these links. Requirements at the stakeholder level will typically be broken down, and more detailed at the system level. These in turn link and break down into more detail at the subsystem level, and again down to the component requirements. If a requirement is not traceable to a higher of lower level, then it is likely not required and can be dismissed. Similarly, each requirement must be written to be testable, and have an associated test case. A requirements management tool is employed to assist in traceability and to perform analyses of the links. Hull et al. (2002, p.13) list some important forms of traceability analysis techniques that are listed and summarized below:

- Impact Analysis – This determines the impact that would occur to lower level requirements if a selected requirement were to be changed. This helps to change control. If a requirement changes, the impact on lower level requirements can be assessed.
- Derivation Analysis – This works in the opposite direction to impact analysis, and is used to trace to the higher level requirement that gave rise to it. It helps when performing cost benefit analysis. If the cost of implementing some requirement is too high, its benefit can be determined and a decision to keep it or lose it can be made.
- Coverage Analysis – Determines if all requirements trace not only to lower layers, but also across to tests as well. This is a good tool for progress measurement in terms of test development and requirement implementation.

The quality associated with a system can be considered as being fit for purpose and as meeting the requirements. This is determined by tests at each level in the "V" model. This also gives a measure of assurance. "Assurance is defined as the measure of confidence that the security features and architecture of an information system accurately mediate and enforce an organization's information system security policy" (Cole, Krutz and Conley 2005, p.591).

Figure 2 below shows an extension to the classical "V" model by showing the relationships with design and test artefacts. The benefit of this is to show how at each level, design artefacts can be linked to requirements, requirements link to tests, and tests link to test artefacts. This allows very quick and easy determination of the history of any particular design implementation to requirements and test documents. Benefits include:

- Traceability.
- Assurance.
- Ease of reuse.
- Ease of auditing.
- Ease of risk assessment.

- Easier to manage and implement change.
- Easier to assign work functions, responsibilities or outsourcing.
- Mechanism for outsource reporting using existing documentation as templates.
- Easier for scheduling of work including upgrades, obsolescence identification.
- Ease of change management.
- Easier to develop contractual documentation for outsourcing.

SSE-CMM

The Security Systems Engineering Capability Maturity Model (SSE-CMM) identifies a framework to measure and improve the performance of security systems engineering practices (SSE-CMM Project, 2003). Its scope covers the life cycle of security systems. The phases of the life cycle include concept definition, requirements analysis, design, development, integration, installation, operations, maintenance and decommissioning. Each of these phases have associated mile stones, and are usually marked by the end of test activities and/or reviews. By following processes and practices in a formal fashion, confidence in repeatable results should be attained. The SSE-CMM can be used to rate the maturity of an organization's security engineering practices.

Developing defined processes provides many benefits. It allows knowledge gained in previous efforts to be used in the future, resulting in the ability to accurately predict how much effort in time and manpower is required to perform similar functions. It ensures that results are repeatable and measurable. It enhances efficiencies, and provides confidence that security needs are being met. Process development and improvement is so much easier if links from requirements to artefacts are readily available for assessment. Process improvement is clearly articulated as a requirement in standards such as ISO/IEC 27001 (2006, p.10).

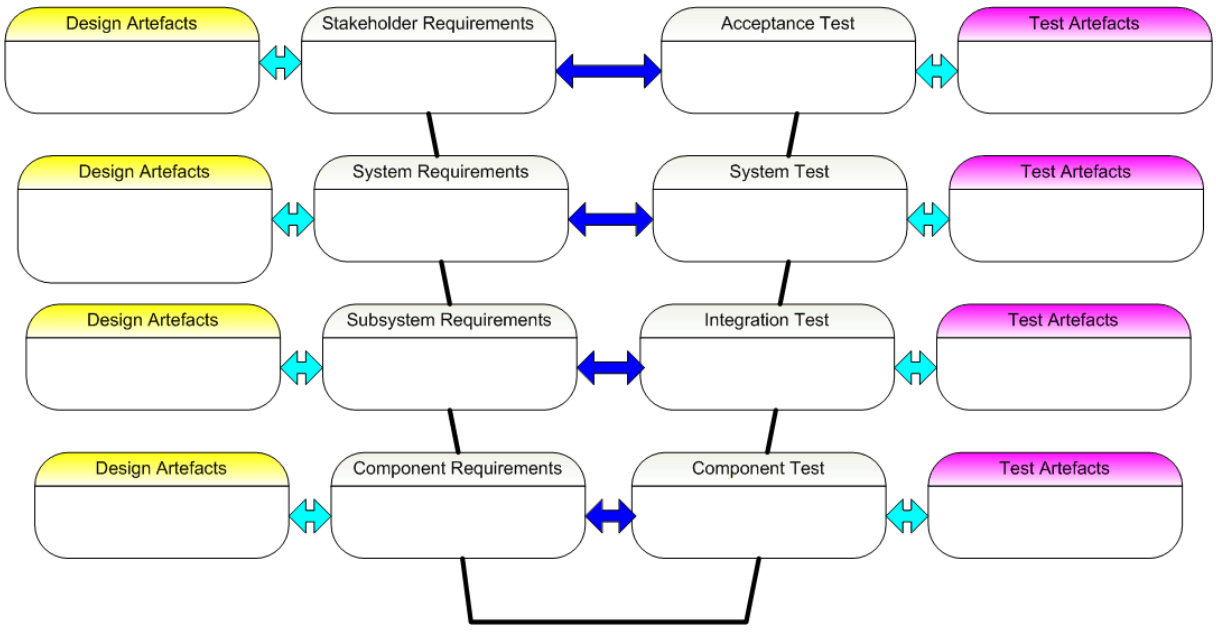

Figure 2. Extended "V" Model to show design and test artefacts

Consider the introduction of a firewall to the discussion. Business requirements, legislation, business partnerships and business rules are the predominant drivers for the development of firewall policy. They are the constraints that limit what is passed in and out of the organisation through the internet. This in turn provides impetus for design changes, risk assessments and subsequent test activities of the firewall including ingress and

egress testing. As the firewall enters service, it must be monitored and audited. Changes in business rules, threats and the development of new technologies will most likely impact policy, risk assessments, firewall rules, test procedures, monitoring and auditing. It shows maturity in process if the impact of a change can be determined quickly, and associated changes are made to relevant practices and procedures. A requirements management tool would greatly assist an organization to track its processes, identify process improvements, and assist in auditing.

Test Plan Development

Testing is a common thread throughout the lifecycle of a system. Assurance is more likely to be achieved if a comprehensive test plan is followed that checks the status of each component. A Test and Evaluation Master Plan (TEMP) functions as a blue print for the test activities, and is comprised of the following activities discussed by Cole et al. (2005, p.55).

- A detailed test plan for complete test coverage of the system under test.
- Communicates the extent and nature of the tests.
- Schedule of events.
- Specification of equipment and organizational requirements.
- Definition of the test methodology.
- Construction of a deliverables list.
- Determination of the expected outputs.
- Instructions on how to carry out the tests.
- Record of the test inputs and results.

Development of the TEMP is instrumental to the effective and comprehensive testing of an ISMS, and should be initiated during the initial design phase of the system. It can be updated as time evolves and should also form the foundation for the artefacts of the test process. It also indicates a maturity in security systems engineering of the organization. A requirements management tool would greatly assist this process from a traceability, tracking and coverage analysis perspective.

Modeling

The model in figure 3 shows how traceability and accountability can be tracked throughout a very generic model of an information security system. Business requirements are driven by and accountable to legislation, regulations and contractual obligations. Each line item is traceable to a number of technical requirements (or controls) which should be driven by standards and have associated risk assessment documents. They also provide data for project management and contract management. In turn these requirements should be traceable to design artefacts which should be in version control and under configuration management. Each design artefact should be linked to test cases which are driven by a TEMP, and produce artefacts such as reports. Linkage from a business requirement through to testing and reporting helps to provide assurance. In the case of a security function being outsourced, it helps to be able to collect all associated artefacts to write the contract. It also assists in reuse and process improvement initiatives for SSE-CMM.

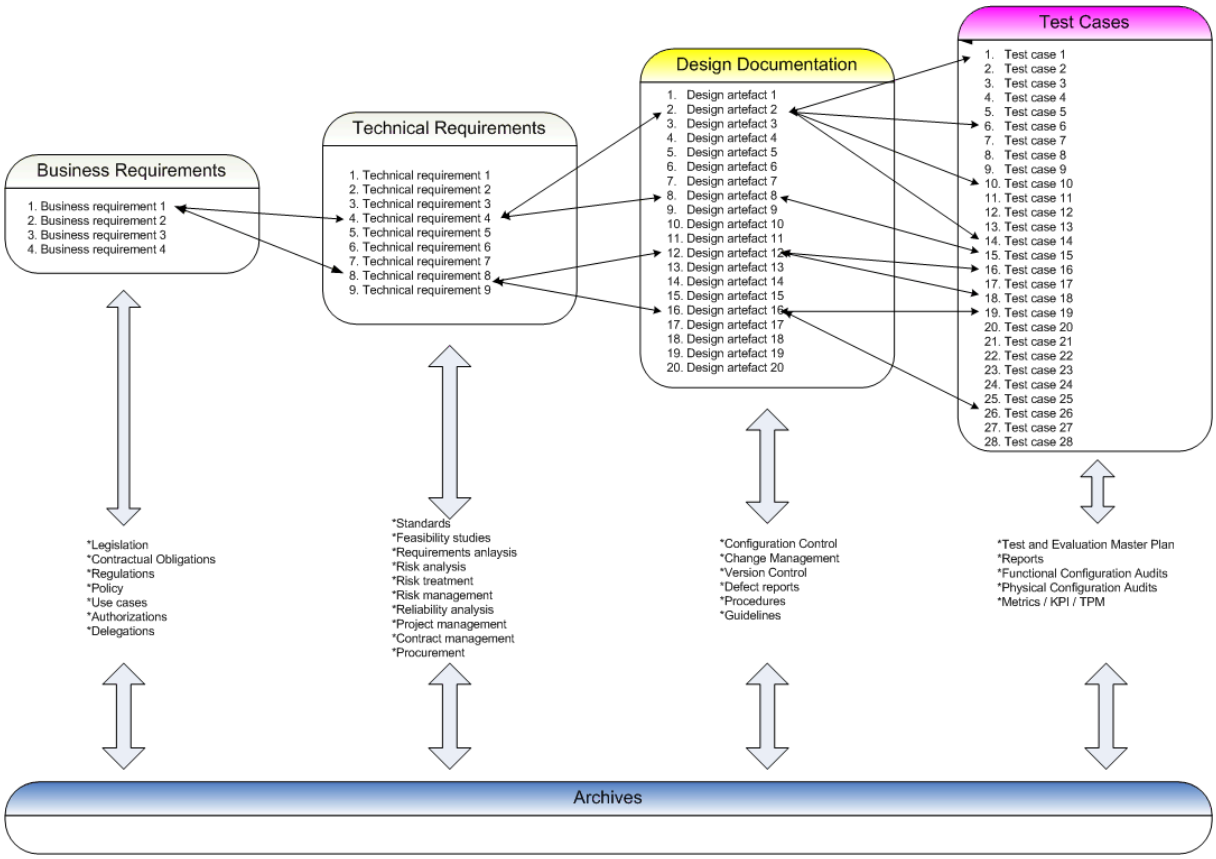

Figure 3. Generic Traceability Model

DOORS

Features

The architecture of DOORS is centred on a database that contains folders, projects and modules in order to manage information and assist in navigation. Project data can include all the business requirements, use cases, risk assessments, legislative constraints, controls, design artefacts, records of meetings, schedules, budgetary information, network diagrams, implementation details, test cases and reports. Anything of value is easily added and can be constructed and modified by the information security architecture team. Navigating around is fairly intuitive and similar to a windows explorer type interface. Access controls can be set to limit user permissions to view, edit and contribute to the artefacts.

Various views can be used to display the data in the most useful way to the user. Data objects can consist of Word documents, spreadsheets, project files, images such as photos, sketches, diagrams to only mention a few. Any number of attributes can be set for any object such as the creator, the owner, review state, approval state, version control number, create time, modification time, priorities and so on, which can be displayed in customised views. History and version control is maintained to ensure that object information is retained including the author of changes, dates of changes, authorisations and is essential to being able to trace changes. Baselines reflect critical times in any project and these can be created and managed with DOORS. These contain all histories and cannot be edited once a baseline has been created, thus providing a repository for the life time of the documentation for the system. Traceability is provided by being able to create links between statements in modules, documents or to diagrams and other images. Reports can be generated similar to any customisable report generated from databases such as Microsoft Access. Tables, documents, spreadsheets, HTML, Microsoft Project, Microsoft Access files and images can be imported and exported into and out of DOORS.

Extensibility and customisation is available through its Doors Extension Language (DXL) programming language to create specialised programs. DOORS also integrates with:

- Mercury TestDirector – For organizing and managing the testing process as well as adding traceability to the testing process.
- Rational Rose – Adds traceability to software design processes. Allows assessment of how requirements impact software design, finds design elements not justified by requirements and finds requirements associated with a specific element of the design.
- PVCS Version Manager – Version control to assist in traceability to specific files, folders or file revisions. Create baselines in DOORS to match PVCS Version Manager version labels. Generate reports on PVCS and DOORS configuration status.
- Rational ClearQuest – Allows the integration of defect and change management and requirements management.
- Rational ClearCase – Enables lifecycle traceability from requirements through to configuration management.
- Telelogic SYNERGY™/Change – Change management solution for change request tracking and reporting to improve accountability by tracking change requests. Increases quality by reducing the risk of unauthorized changes being introduced.
- Telelogic SYSTEM ARCHITECT® - A modelling solution to develop enterprise solutions, and for business process improvement.
- Telelogic DASHBOARD™ - Assists in identifying project risks, status and trends by provision of automated tools for collection, analysis and reporting of measurement data from Telelogic DOORS and Telelogic SYNERGY.

Implementation

Figure 4 below shows a sample implementation of the model from figure 3, shown in a single module. It is not complete and shown for example purposes only. It shows the module explorer on the left and the module data on the right. The bottom left panel shows the user name of Eric McCall which is the DOORS evaluation version username. An evaluation CD, valid for 30 days can be requested through Telelogic. The arrows to the right of the module data shows that there are links attached to the data. Outgoing links show the arrow pointing to the right. Incoming links show the arrow pointing to the left. Only the links as shown in figure 3 were created for the purposes of demonstration. Text, tables, spreadsheets and images can be added to each line of module data.

The implementation used is purposely very generic for demonstration purposes. Artefacts could be anything from the results of a risk assessments, authorisations, firewall rule sets, design documents, results of audits, identification of threats and vulnerabilities, risk strategies, design of controls, management commitment documents, and so on. It is straightforward to implement many layers of requirements, artefacts, test cases and any other class of documentation that could be useful, and create appropriate links. Traceability, impact analysis, derivation analysis and coverage analysis reports can then be created very easily.

Figure 5 shows the result of a traceability report. It shows the links between business requirement 1, its technical requirements, design artefacts and the test cases. Supporting text, images, spreadsheets and tables have not been added, in order to show the traceability perspective.

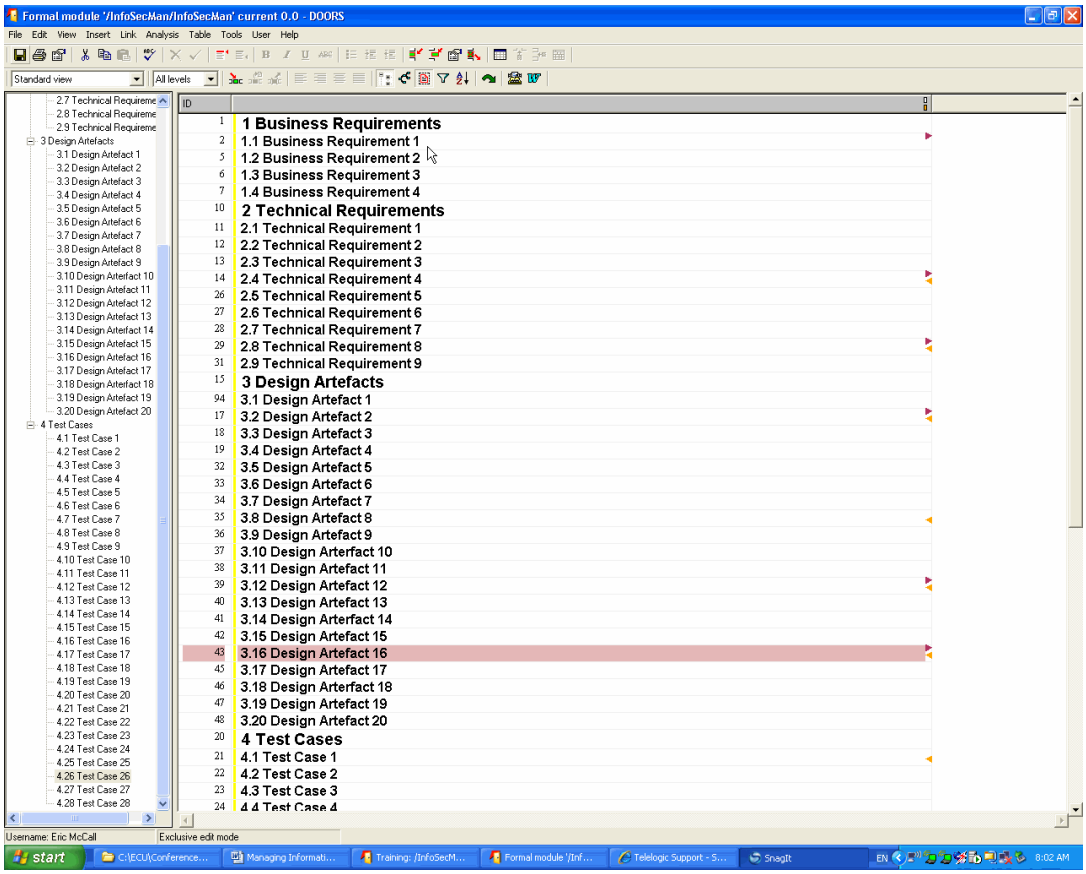

Figure 4. DOORS formal module display

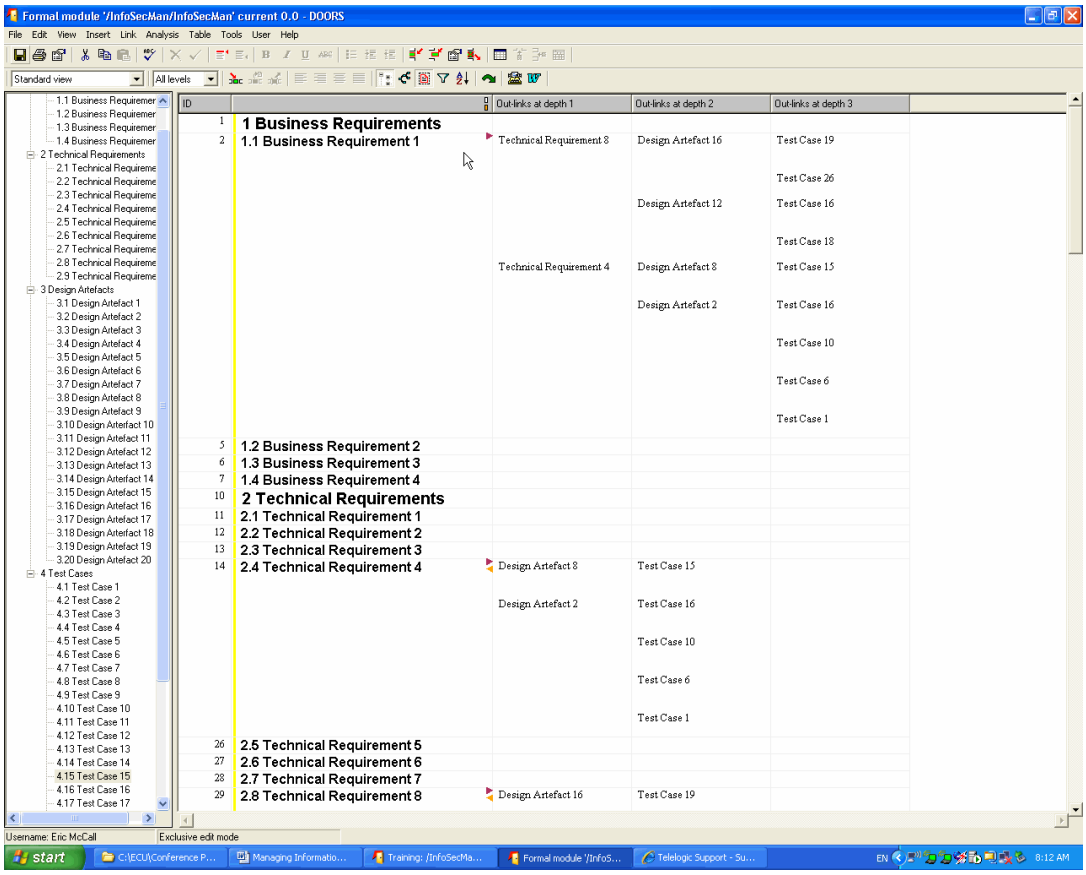

Figure 5. Traceability report

Templates can be created and reused to conform with various standards. Some of the templates that are available from the evaluation version are shown below in figure 6, with the tree from the IEEE software standards expanded.

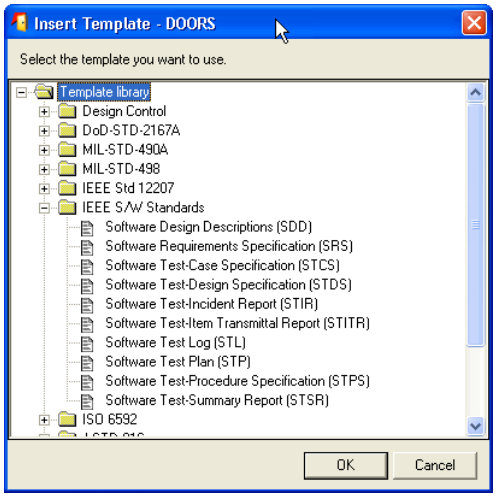

Figure 6. Available templates in evaluation version.

ISO/IEC 27001:2006

International Standard ISO/IEC 27001:2006 specifies generic, top level requirements for "establishing, implementing, operating, monitoring, reviewing, maintaining and improving a documented ISMS within the context of the organization's overall business risks" (Standards Australia, 2006, p.1). These requirements are very much in line with the implementation guidance provided by International Standard ISO/IEC 17799:2006. With respect to conducting an internal ISMS audit, ISO/IEC 27001 (2006, p10) declares:

> *The organization shall conduct internal ISMS audits at planned intervals to determine whether the control objectives, controls, processes and procedures of its ISMS:*
> *a)conform to the requirements of this International Standard and relevant legislation or regulations;*
> *b)conform to the identified information security requirements;*
> *c)are effectively implemented and maintained; and*
> *d)perform as expected.*

The common thread to the artefacts of the information security system that will be audited, are the requirements. Requirements engineering principles and practices are proposed as a tool for managing the complexity of the traceability aspects of the ISMS.

ISO/IEC 27001 (2006, p.7) states:

> *Documentation shall include records of management decisions, ensure that actions are traceable to management decisions and policies, and ensure that the recorded results are reproducible. It is important to demonstrate the relationship from the selected controls back to the results of the risk assessment and risk treatment process, and subsequently back to the ISMS policy and objectives.*

The essential documentation requirements listed in ISO/IEC 27001 (2006, p.7) is summarized to include:

- Statements of the ISMS policy.
- Scope of the ISMS.
- Procedures and controls in support of the ISMS.
- Description of the risk assessment methodology.
- Risk treatment report.
- Risk treatment plan.

- Documented procedures to support security processes of planning, operation, control and measurement of the effectiveness of controls.
- Records required by ISO 27001.
- Statement of applicability.

A requirements management tool such as DOORS, can clearly help assist in attaining these objectives. Projects can contain multiple modules, and links can be created between them. A simplified traceability model for ISO 27001 is shown below in figure 7.

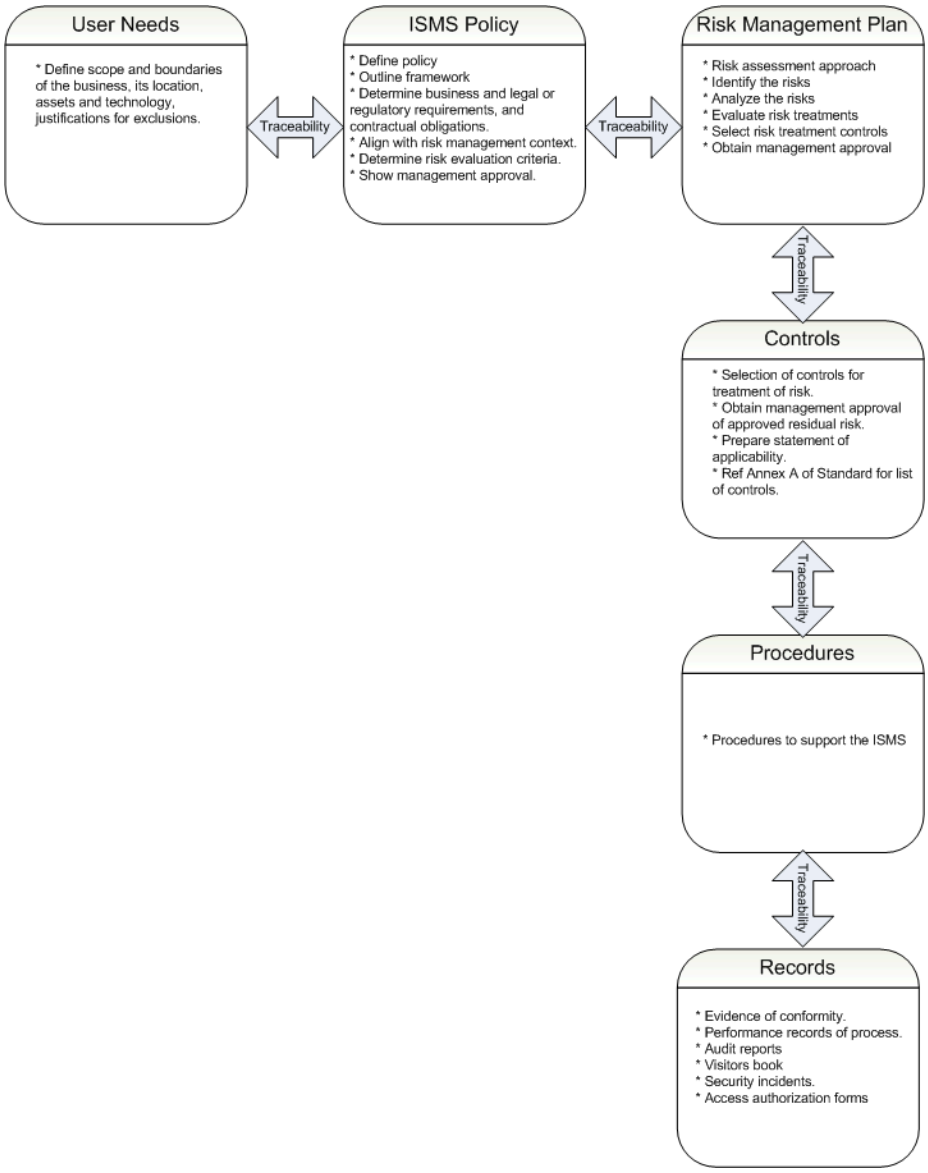

Figure 7. Simple model view of traceability for ISO/IEC 27001.

Figure 8 below shows the beginning of a template for an ISO 27001 controls module that can be easily linked to other modules such as a user needs module, an ISMS policy module, a risk management module, a controls module, a procedures module and a records module. Once created, it would then be easy to create traceability reports, impact, derivation and coverage analysis reports, create baselines, import and export supporting documents to provide a tool to manage the complexity of an information security management system.

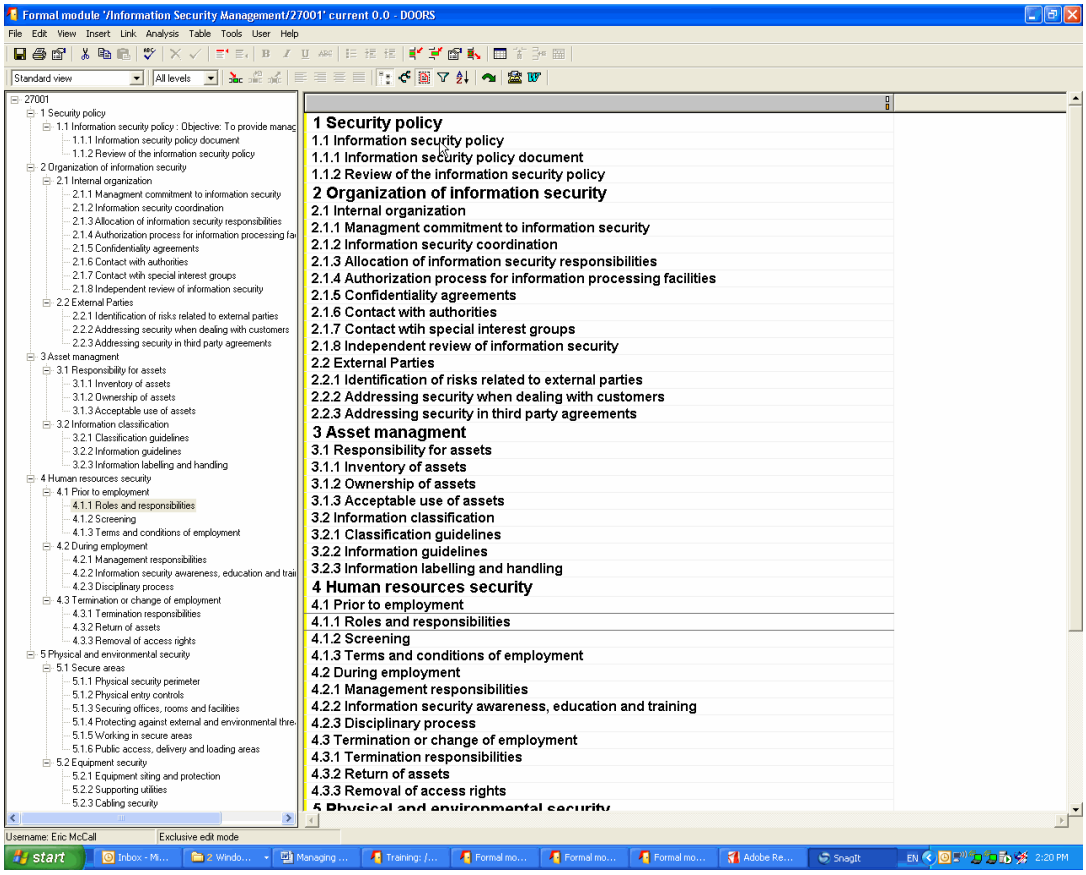

Figure 8. ISO/IEC 27001 Module under construction.

CONCLUSION

This paper has presented a tool based approach to managing the complexity of information security where traceability between business requirements and associated artefacts such as policies, guidelines, procedures and other documentation is desirable. This is especially true when the functions of information security may be outsourced or outnationed and assurance may need to be determined in the least amount of time. Traceability to business requirements also helps to optimise many functions including risk assessment, scheduling, budgeting, change management and the development of test cases. This also makes it easier to assign or outsource functionality and responsibility and to identify documentation that will be required to support business requirements. This also makes it easier to develop contractual documentation, perform risk assessments and conduct auditing activities.

Artefacts of the security management function exist, such as policy documents, procedures, guidelines, firewall rule sets, configuration files, group policies, procurement assessments, access control lists, network diagrams, cabling diagrams, patch panel configurations and router configuration scripts, to name only a few. All of these documents are linked back to business requirements. If they don't, then they probably are not required, and may just be increasing the attack surface and running costs whilst reducing performance. Uses of a requirements management tool could include performing as the repository for an ISMS, a quality management system, and a change control system simultaneously. It is only the view of the information that needs to be changed, which is available through a requirements management tool such as DOORS.

Telelogic's DOORS product is well suited for the task of managing requirements. The architecture of the repository for this information can be tailored and extended to customise it as a management tool. If a business requirement is changed, all associated documents that are linked to the requirement are instantly identified,

providing an impact analysis. If a guideline, or procedure, needs to be modified because the function is being outsourced, a derivation analysis can be used to determine the business requirement that gave rise to it. Assurance is assisted by being able to perform a coverage analysis to ensure that any particular business requirement has associated implementation artefacts and test cases. Change proposal investigations are assisted by easily being able to track and trace any implementation to related artefacts, and back to the original business requirement. If an outsourced function needs to be modified, it can be tracked back to the originating business requirement and assessed in the least amount of time before being approved. This also assists in base lining and being able to reconstruct the state of the artefacts, including test reports, for any particular point in time together with histories, approvals and authorisations. Future work includes continual development of an ISMS as outlined in ISO/IEC 27001:2006 at a more practical level.

REFERENCES

Cole, E., Krutz, R., Conley, J.W. (2005). *Network Security Bible*. Wiley Publishing, Inc., Indianapolis.

Dick, J. (2004). *What is Requirements Management*. Retrieved October 1, 2006 From www.neueda.com/automationtools/pdfs/What is Requirements Management.pdf

Hull, E., Jackson, J., Dick, J. (2002), *Requirements Engineering*. Springer, London.

SSE-CMM Project, (June 15 2003), *Systems Security Engineering Capability Maturity Model, SSE-CMM, Model Description Document, Version 3.0*. Retrieved October 6, 2006 From http://www.sse-cmm.org/docs/ssecmmv3final.pdf

Standards Australia. (June 23 2006). *AS/NZS ISO/IEC 27001:2006 Information technology – Security techniques – Information security management systems – Requirements*. Standards Australia, Sydney

COPYRIGHT