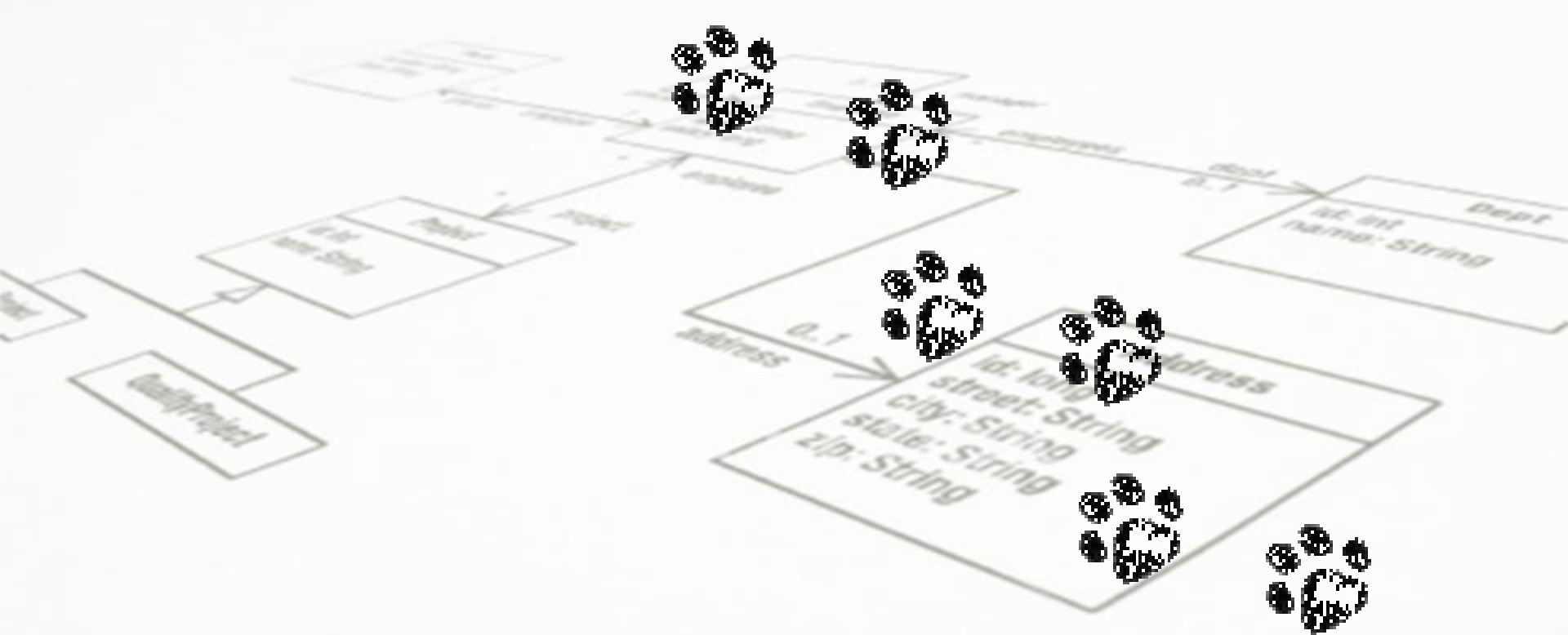




Assignment I:

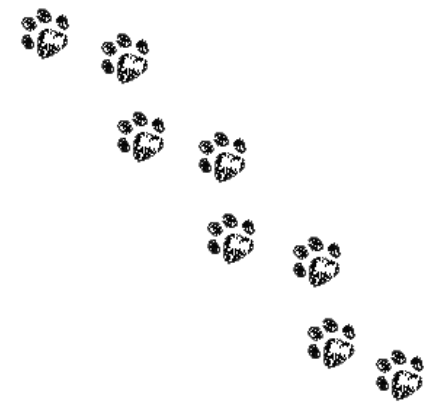
Requirements Traceability Tools

Requirements Engineering II

November 3rd, 2008

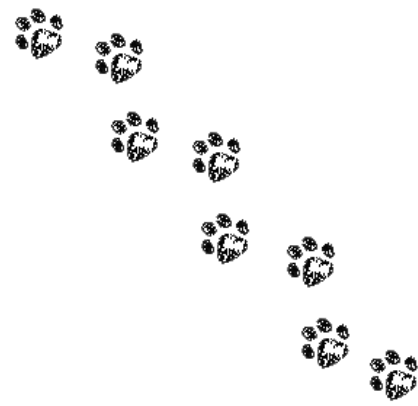
Group C: Michael Sommerhalder, Krzysztof Dabkowski, Stefan Schurgast

Agenda



1. You are here
2. The role of requirements traceability tools (RTT)
3. Requirements of RTT
4. Market overview
5. Are the tools meeting RT requirements?
6. Conclusion about RTT on market

You are here.



Traceability helps you ...

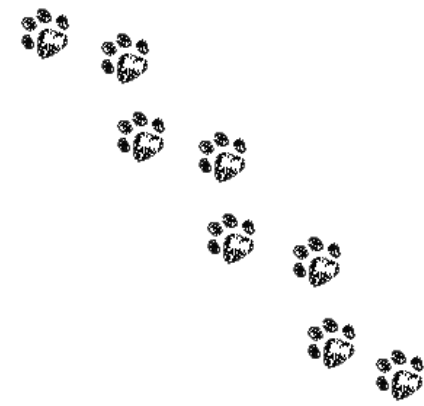
... find a way through requirements,

... retrieve relevant information and

... see implication of change

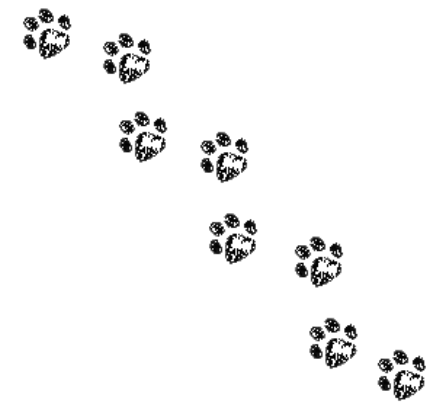


The role of requirements traceability tools



- Store and manage requirements
 - Storage and group management
 - Versioning
 - Branching
- Enable requirements tracing
 - Evaluation for completeness checks
(do all system level requirements go down to equipment level with or without modification)
 - Assessment of requirements deviations over all levels
 - Requirements Analysis
- Make sure that requirements tracing is done (because documentation isn't usually a programmers hobby)

Requirements of RTT as seen in literature



- Strong preRS support [2]
- Capture rationale of different abstraction levels [1, 2, 4]
- Capture requirement's history [2]
- Automated RT-matrix [3]
- Automated matching from design documentation representation to requirements documentation [3]
- Link rationale and sources to requirements [5]
- Be able to trace between performance/management/modeling domain [4]

Requirements of RTT by INCOSE

- Traceability analysis
- Identify inconsistencies
- Visibility of existing links
- Verification of requirements

Source: INCOSE, <http://www.paper-review.com/tools/rms/read.php>



Merging literature and pracicers

- Traceability analysis
 - Automated RT-matrix
 - Matching design doc. to requirement doc.
- Identify inconsistencies
- Visibiliy of existing links
 - Link rationale and sources to requirements
 - Trace between performance/management/modelling domain
 - ~ Capture requirement's history
 - ~ Focus on preRS
- Verification of requirements
- Requirements performance verification (roll up of actuals)

plus

- Rationale at different abstraction levels



Market overview



IRQA 4 - [IRQA Tutorial - CCC System - 25 appliances (ADMIN) : Requirements (89) "Initial"]

Project Edit Engineering process Tools Reports View Window Help

Current

1st Level: Hierarchy 2nd Level:

Code	Name	Priority	Status
FUNC_0015	CCC System modes	Medium	Postponed
FUNC_0020	Fully computerised control	Low	To be reviewed
FUNC_0040	Half dual control	-- To be determined --	Pending
FUNC_0050	Full manual control	Low	Rejected
FUNC_0070	Manual Control	Medium	To be reviewed
FUNC_0060	Transitions between modes	High	Postponed
FUNC_0100	Supported Bandwidth	Low	Pending
FUNC_0200	Loading and unloading of applia...	-- To be determined --	Rejected
FUNC_0300	Appliance working modes	Low	To be reviewed
FUNC_0305	Transitions from Installed mode	Medium	Postponed
FUNC_0307	Transitions from Ready mode	High	Pending
FUNC_0310	Transitions from Loading mode	Medium	Postponed
FUNC_0320	Transitions from Active mode	Low	Rejected
FUNC_0325	Transitions from Alarm mode	-- To be determined --	Pending
FUNC_0327	Transitions from Acknowledged ...	Low	Postponed
FUNC_0330	Transition from Stopped mode	Medium	Pending
FUNC_0340	Event generation	High	To be reviewed
FUNC_0350	Appliance upgrade	Low	Postponed
FUNC_0400	Monitoring	-- To be determined --	Pending
FUNC_0410	Working mode monitoring	Low	Rejected
FUNC_0430	Pre-programming	Medium	To be reviewed
FUNC_0440	Pre-programmed states	High	Pending
FUNC_0500	Event registration	Medium	Postponed
FUNC_0610	Routing security	Medium	Postponed
FUNC_0620	Firewall	Low	Rejected
FUNC_0630	Remote activation of the CCC C...	-- To be determined --	Postponed
FUNC_0670	Self diagnostic test of computer	Low	Pending
FUNC_0680	Self diagnostic test of micro cont...	-- To be determined --	Postponed
FUNC_0690	Self diagnostic test of appliance ...	Low	Pending
FUNC_0700	Self diagnostic test of all critical	Medium	Rejected

Ready

Description

The system will enable control of all electric appliances in the average home; for example, kitchen appliances (ovens, microwave cookers, and refrigerators), air conditioners, central heating, clocks, entertainment devices (TV, radio, DVD players/recorders and MP3 players), and alarm systems.

MAY NUM DESP

OSRMT - DEMO@demo

File Edit Tools Admin System Help

OSRMT

- Feature
 - System Data Entry
 - Manual Data Entry
 - Maintain Full Artifact Text
 - Binary File Attachments
 - Import Requirements
 - Spellcheck
 - Externally Linked Documents
 - Uniquely Identify Artifacts
 - Define Artifact Hierarchy
 - Artifact Detail List
 - System Navigation
 - Traceability
 - Identify Source and Origin
 - Trace External Artifacts
 - Trace Artifacts
 - Identify Untraced Requirement
 - View Related Artifacts
 - Configuration Management
 - System Output
 - Customization
 - Security

Feature #	Name
1	System Data Entry
2	Manual Data Entry
3	Maintain Full Artifact Text
4	Binary File Attachments
5	Import Requirements
6	Custom Database Fields
7	Spellcheck
8	Externally Linked Documents
9	Uniquely Identify Artifacts
10	Define Artifact Hierarchy
11	User Defined Fields
12	System Navigation
13	Group and Sort Artifacts
14	Filter List of Artifacts
15	Ad hoc Queries
16	Traceability
17	Identify Source and Origin
18	Trace External Artifacts
19	Trace Artifacts
20	Identify Untraced Requirement
21	Configuration Management
22	Track Requirement History

Requirement

Requirement \ Details \ Background \ Use Case \ Relations \ History \

Goal: Create, retrieve, update and delete hierarchy of artifacts

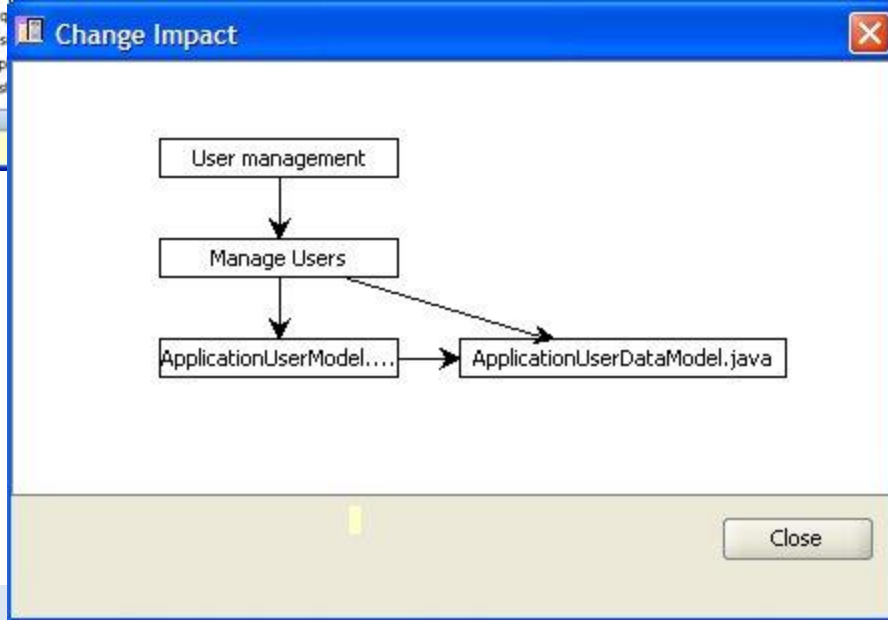
Context: Artifacts: Features, Requirements, Design, Implementation, Test Cases etc

Precondition: New product created

Main Flow:

Step		
1	User selects a product	Move Up
2	User selects an artifact of the desired type	Move Down
3	User creates a new child artifact	Add Row
4	System displays a data entry form for the selected type	Remove Row
5	User enters and saves the new data elements	Apply
6	System creates a hierarchical relationship between the artifacts	

OK Cancel



RTT compared



	DOORS	IRqA	RequisitePro	OSRM
Strong preRS support (2)	Ability to capture requirements from file or self creating; identification with by key words, structure, unique identifiers, etc	Strong importing capabilities; several ways of parsing the requirements	Document parser can automatically identify requirements based on key-words. Requirements are automatically entered into the requirements repository	Import from XML, usage of self defined macros. Capabilities of ordering requirements.
Capture rationale of different abstraction levels (1, 2, 4)	Also critical, is the ability to attach rationale, assignments, criticality, test/validation and many other issues to the requirement, allocation, and the system element to which a requirement is linked.	Rationale of each change may be saved. Not well developed feature.	Allocation, rationale, accountability, test/validation, criticality, and issues can be captured via text, list (single or multi-value), integer, real, date, time, or URL link	Rationale, and numerous fields are available.
Capture requirement's history (2)	automatically records who, what (down to the actual attribute changed), when and how automatically.	Requirement date, time, user, change reason, among others, are recorded in addition to name and description changes in requirement history.	includes date, time, author, before and after contents, and annotated change rationale by the author. Automatically updated.	In case of change saving a whole copy of a requirement

RTT compared cont'd

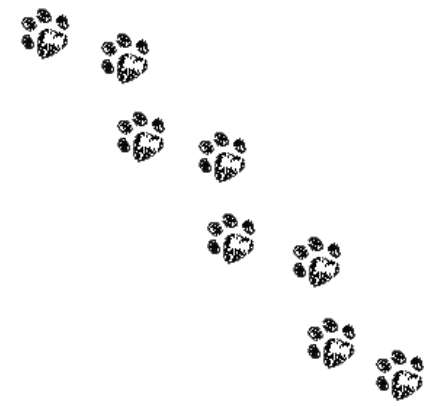


	DOORS	IRqA	RequisitePro	OSRM
Automated RT-matrix (3)	entire chain of links performed on line or printed in the form of a matrix style report	Supports wide capabilities connected to RT-matrix like system elements reqs, test scenarios relationships and others	Derived documents can be traced to the source requirement using a right click in a traceability matrix.	Traceability matrix as well as traceability tree fully supports links to the specific reqs.
Link rationale and sources to requirements (5)	Done by using assigned attributes or other associated objects in the database.	Performed using traceability matrix and block diagrams.	Link created on different levels and stages of the specification process	Linked, but comparison not possible
Be able to trace between performance/management/modelling domain (4)	supports this using attributes associated with the requirements or independent data linked to the requirements	Quantitative values may be associated to requirements (non-functional in the case mentioned) or services through user-defined dynamic attributes.	Using requirement types, users can create performance requirements, system requirements, marketing requirements, test requirements, etc.	Performance requirements may be entered with priority, cost, risks etc. should be able to do this using the customized reference fields.

Source: INCOSE, <http://www.paper-review.com/tools/rms/read.php>

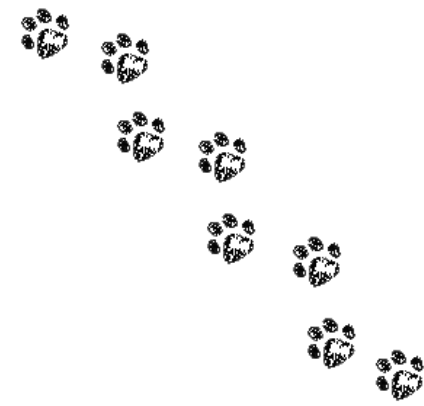


RT Matrix in tools



- IBM Rational RequisitePro
 - A RT Matrix is automatically created by importing reqs
 - derived reqs can be traced from the matrix
- Visure IRQA 4
 - RT Matrix links systems reqs with system elements
 - link to test scenarios
 - enables filtering for finding a specific set of reqs
 - capability of exporting the matrix to MS Excel

RT Matrix in tools cont'd



- Telelogic DoorsRep
 - RT Matrix available only as a form of report
 - shows all links between reqs and system elements
- OSRMT
 - RT Matrix available but poor functionality supported (lack of information)

Conclusion of RTT comparison

- All of the tools satisfy the main aspects
- Differences in functionalities and usability
- Choice between the products should be determined on how the specification process looks like

Sources



- [1] Dick, J. (2005). Design traceability. IEEE Software 22, 6 (Nov./Dec. 2005). 14-16.
- [2] Gotel, O., A. Finkelstein (1994) An Analysis of the Requirements Traceability Problem, Proceedings of the First International Conference on Requirements Engineering, Colorado Springs. 94-101.
- [3] Huffman Hayes, J. A. Dekhtyar, S. K. Sundaram (2005). Improving After-the-Fact Tracing and Mapping: Supporting Software Quality Predictions. IEEE Software 22, 6 (Nov./Dec. 2005). 30-37.
- [4] Jarke, M (1998). Requirements Traceability. Communications of the ACM 41, 12 (Dec. 1998). 32- 36.
- [5] Ramesh, B., M. Jarke (2001). Toward Reference Models for Requirements Traceability. IEEE Transactions on Software Engineering 27, 1 (Jan 2001). 58-92.
- [6] Cleland-Huang, J., R. Settini, O. BenKhadra, E. Berezanskaya, S. Christina (2005). Goal-Centric Traceability for Managing Non-Functional Requirements. Proceedings of the 27th International Conference on Software Engineering, St. Louis, USA. 362-371.
- [7] Huffman Hayes, J., A. Dekhtyar, J. Osborne (2003). Improving Requirements Tracing via Information Retrieval. Proceedings of the 11th IEEE International Requirements Engineering Conference, Monterey Bay. 138-147.
- [8] INCOSE Requirements Management Tools Survey, <http://www.paper-review.com/tools/rms/read.php>, accessed 28.10 22:50