



HAL
open science

Tools collaboration for deployment of a requirements engineering process

Agusti Canals, Stéphane Lelièvre

► To cite this version:

Agusti Canals, Stéphane Lelièvre. Tools collaboration for deployment of a requirements engineering process. 2nd Embedded Real Time Software Congress (ERTS'04), 2004, Toulouse, France. hal-02271222

HAL Id: hal-02271222

<https://hal.science/hal-02271222>

Submitted on 26 Aug 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Session 6B: Development Process and its Improvement

Tools collaboration for deployment of a requirements engineering process

Agusti Canals, Stéphane Lelièvre
CS - Consultants

Agusti.canals@c-s.fr – stephane.lelievre@c-s.fr

Tel : +33 (0) 5 61 17 66 66 – Fax : +33 (0) 5 61 54 13 39

ZAC de la Grande Plaine - Rue Brindejonc des Moulinais - BP 5872 - 31506 TOULOUSE Cedex 5

Requirements Engineering vs Requirements Traceability

The first steps of the system and requirements engineering phases of any project are to define the management aspects (management plan, requirements engineering process, schedule management) and the project specifications, in the widest sense: collection, analysis and synthesis of the need expressions from all stakeholders, and definition of requirements to express this need. Then requirements will be allocated, changed, re-organized, and tracked during all the project lifecycle.

So the first priority is to have a process, even a simple one, then it's helpful to have this process supported by tools. A good process, even manual, is better than tools used without any process.

As there is often a confusion there, it seems important to distinguish 2 steps in a requirements management process :

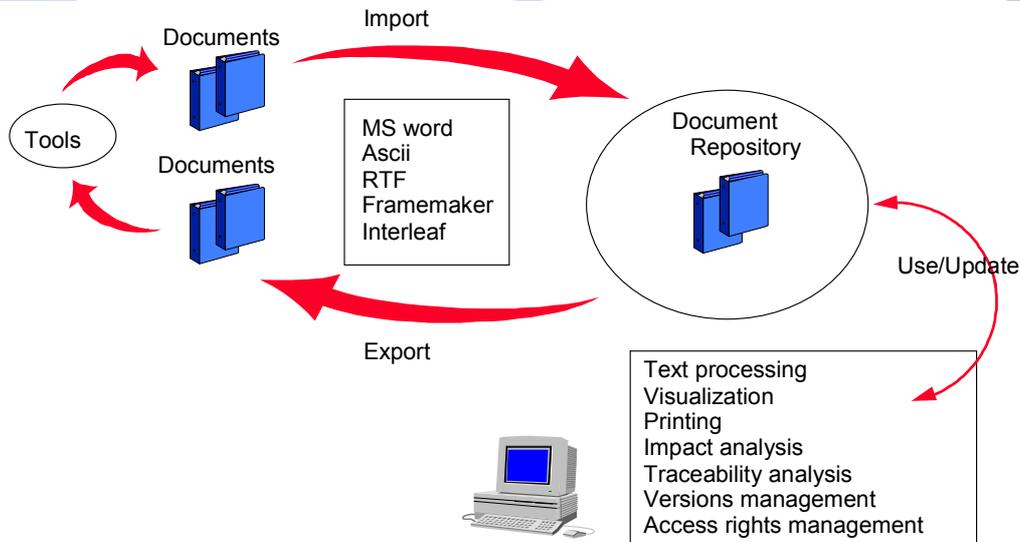
- Requirements elicitation, identification and allocation : For this step there are a lot of publications, feedbacks and studies. These requirements are usually formalized in 'Specifications' documents for the development teams or the subcontractors,
- Once formalized, the requirements have to be implemented and tested by the hardware and software development teams. This is the 'requirements traceability'.

Requirements and implementation artefacts often change, and the impact of each change must be analyzed.

Solutions currently used by CS

For several years, the process used by CS to manage requirements was the following one :





Each development tool included in the project development cycle is completed by a set of macros or scripts allowing input data traceability. The documentation produced by each development tool is stored in a repository (in our case the database of DOORS™, the very well known tool provided by Telelogic). The *import/export* functions allow to extract traceability information from the input documents, store and manage it in a database, perform impact analysis and build traceability matrices.

This CS instrumented process was used by the project development teams, but also by the teams answering to invitations to tender. So this process was supposed to be used during the 2 majors steps defined above : requirements capture and definition, then traceability and impact analysis. We used only DOORS™ to support this approach.

Widely involved in the Airbus A380 program, CS also used REQTIFY™, from TNI-Valiosys, for the A380 software projects. CS also selected REQTIFY™ for other Aerospace and Automotive projects. REQTIFY™ is a simple, easy to handle and easy to deploy tool that needs formalized requirements as inputs, then provides requirements traceability, bi-directional impact analysis and documentation generation.

So, as several large groups today, we have 2 tools involved in our requirements management process : DOORS™ from Telelogic and REQTIFY™ from TNI-Valiosys.

Tools positioning analysis

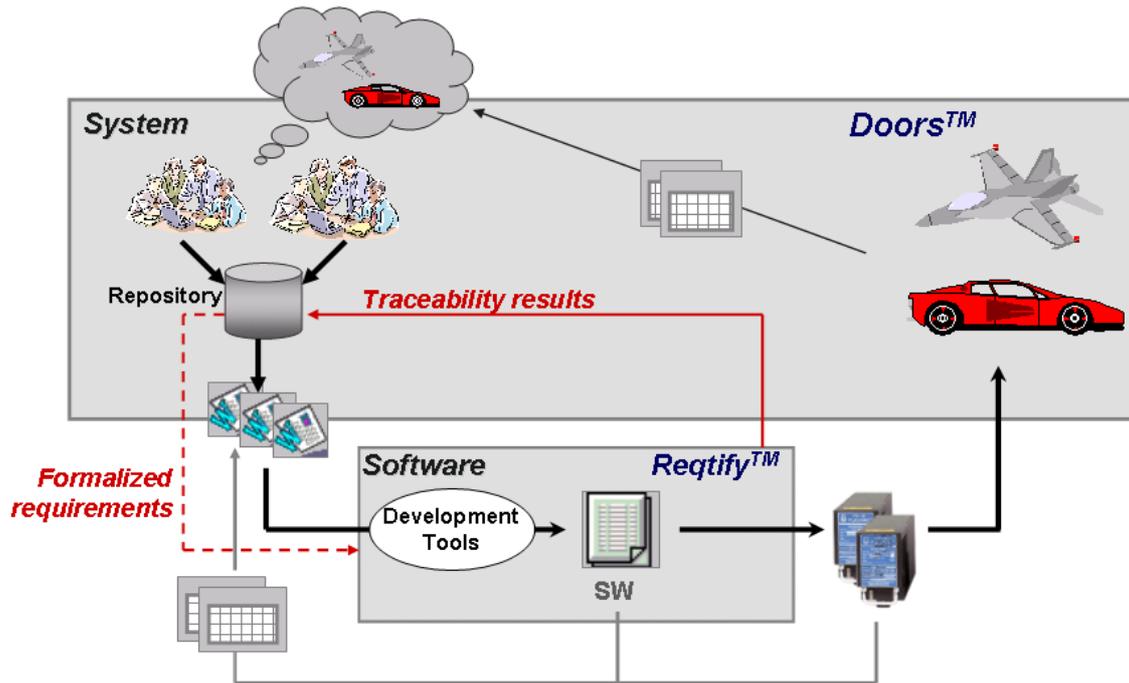
Instead of defining a competition for the selection of only one tool for the group, we decided to analyze their real positioning in order to get the maximal added value from each tool and then to see what was the best approach :

- DOORS™ is very efficient to manage the requirements in the high-level system design phases, where several teams are discussing the customer need, what are the requirements,... At this level it is necessary to centralize the discussions in a database, manage change proposals about requirements definition (with the DOORS™ CPS – Change Proposal System). The center of interest is the requirements definition, the center of the tool is the database, it's so consistent. The outputs of this approach are some formalized "Requirements documents", used as inputs for the next development steps.
- For the development and tests phases, the use of DOORS™ was possible only thanks to more or less accurate internally developed macros and/or interfaces with the tools used during development. But we think that it's not the most effective way to use DOORS™, for a very simple reason : the center of interest is no more the requirements definition, but their implementation and tests, which is disseminated across various development and test tools. It may be complex and expensive to try to centralize these artefacts in a database. The risk is to make the tool rejected by the development teams, but if the real problem would not be the tool but the way to use it.
- REQTIFY™ is very efficient for development and tests teams receiving formalized specifications documents as inputs, outputs of a manual work for small projects or of a work supported by DOORS™ for larger projects. REQTIFY™ is

provided including interfaces with various tools. But REQTIFY™ itself can't manage collaborative definition of high-level requirements or system requirements. Its added value starts once the requirements have been defined, not before.

An original 2 tools based approach to maximize the added value

Following the positioning analysis, we concluded that the maximal added value is in the collaboration between these 2 tools instead of in a competition. This 2 tools based approach is described below :



- At system level the multi-team need analysis is centralized in a database, requirements are discussed, optimised with rigorous change decision process, upto their integration in one or several specifications documents. Clearly, this is not the positioning of REQTIFY™, and this need is perfectly addressed by DOORS™,
- REQTIFY™ is used as soon as the specifications are produced, to capture requirements in documents and manage them across the whole development lifecycle for software.

Important note : Requirements can be captured directly in the database, but as teams usually work through document exchanges, such a direct import between tools creates a second channel and this duplication must be managed. If not, it must be preferred to keep the same reference as project teams, meaning specifications documents themselves,

- REQTIFY™ then manages requirements implementation and tests during all the development process, and provides coverage analysis, impact analysis of a change at any level and various reports. Results can be produced in a documentation format (Word, HTML,...) or directly inserted in the DOORS™ database for analysis by the system engineers.

A very important point is that the insertion of REQTIFY™ in our process was possible without any modification : neither of our requirements management approach nor of the work performed at high level with DOORS™ :

Such an approach has several major benefits :

- DOORS™ is used for its maximum added value, widely recognized and presented : requirements definition in a widely collaborative and iterative way,



- We provide development teams with REQTIFY™ as a light and efficient solution, more external than a database-based one, keeping the focus on the implementation, for traceability and impact analysis. As it has been designed for these tasks, REQTIFY™ offers a very pleasant user-interface to navigate in the traceability graph,
- By solving our compatibility and interface problems (versions of various used tools), REQTIFY™ allows us to replace more and more macros and scripts by only one tool, supported by its editor. A simple ‘Save’ action leads REQTIFY™ to update the traceability analysis results, there’s no need to synchronise the files or models with a database,
- Each tool is a well-accepted solution at the appropriate level. This acceptance allows to widely deploy the requirements engineering process : This process, initially supported and defined with DOORS™, is not modified by REQTIFY™ as this tool needs such a definition as input,
- DOORS™ requirements can be easily captured by REQTIFY™ for development teams, and REQTIFY™ analysis results can be provided to DOORS™ in order to give a good view of the system implementation to the system engineers, using DOORS™ ,
- The global cost is much lower : the overall purchase cost of REQTIFY™ is lower, the training is very short and its administration is very simple as there’s no database,
- **Most of all, this 2 tools based approach is an effective and economical solution to make the development teams play the game of the requirements management process.**

As already said, this CS instrumented process is used by the project development teams, but also by the teams answering invitations to tender, or request for proposals. We also plan to combine the capabilities of each tool in order to choose the most efficient approach, but the general way should be :

- To use REQTIFY™ to generate the coverage matrix showing the compliance with the customer requirements, as it takes only a few minutes to customize the tool according to the requirements syntax. Of course DOORS™ will still be used for some proposals where it is the most adapted.
- Once the business is secured, to use the DOORS™ and REQTIFY™ collaboration, or only one of these tools depending on the project context.

We also think that REQTIFY™ can be used even when subcontracting with a customer using DOORS™ , but we still have to confirm this view in real-size projects.

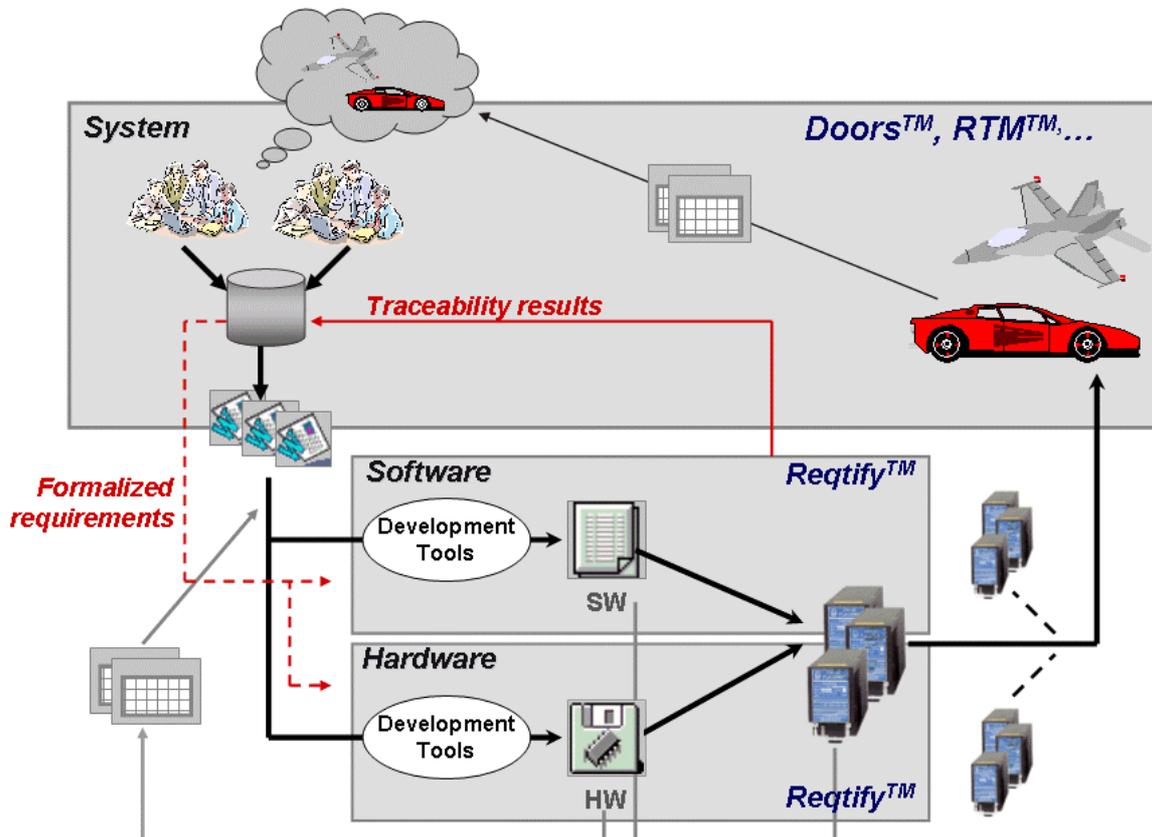


Beyond the CS experience

This approach based on the collaboration between a database based tool (DOORS™ but also RTM™ from Integrated Chipware, or CORE™,...) and a file-centric tool (REQTIFY™) can be extended in a very general way to a wide range of developments, including both Software and Hardware (FPGA, Electronic Boards) developments. So it can be a good answer to a wide range of problems not solved by the use of only one tool for all the process steps. Let's just take 2 examples among a lot of others :

- Hardware design : today a recent and very demanding standard has to be applied for Hardware in Aeronautics : the DO254 standard, equivalent of what is the DO178B for Software. Even for mature requirements engineering process, applying it across the Hardware design should be more easily accepted and easier to deploy with a file-centric tool than with a database based tool (does it make sense to import hundreds of VHDL files and test benches in a database ?) .
- As a matter of fact, engineers live in a document-oriented world : Today's projects need to be very reactive and leave no place for the implementation of a bad requirement, or for a change realized without impact analysis. The document-based work generates a high risk of redundancy and/or non-synchronisation with a database. A 'file-centric' tool is a good solution for managing what is effectively in the documents.

The global process could so be presented as below :



Conclusion : Perspectives and reuse of CS experience

For requirements management, the process is the priority. Tools are only there to support this process as effectively as possible. Today, we consider that this 2 tools based approach is the most effective one for large groups, from both technical and cost points of view. It can be applied to software and hardware designs. It should also be the unique solution to support the deployment of recent demanding standards as DO254 for Hardware in Aeronautics, and is a good way to avoid redundancy in some document-oriented projects.



CS used DOORS™ and REQTIFY™ individually in various contexts, and is now deploying a 2 tools based approach described above. We consider there is a higher value in a collaboration rather than in the selection of only one tool after a competition.

As a consulting company, CS considers that this experience can help major groups looking for an effective solution to support their requirements engineering process, in Aerospace, Automotive, Industrial and other markets.

References :

[Air&Cosmos] ‘On the leading-edge of mission critical software’ - Air & Cosmos article published (in French) in n°1908S dated 17/10/03

[A380-SW] ‘Traceability of software requirements, why ? how ? – A380 airborne avionics software’.
Patrick Farail – Airbus France – Update of the article presented at DASIA 2000

[Daimler1] ‘Requirements Engineering in Automotive Development’ – Mathias Weber and Joachin Weisbrod – Daimler Chrysler RIC

[AFIS1] ‘L’expérience THALES avec DOORS’ – Jean-Claude Fromenteau - AFIS 2nd Conference

DOORS™ is provided by Telelogic : www.telelogic.com

REQTIFY™ is provided by TNI-Valiosys : www.tni-valiosys.com

About CS

CS is a major player in the market for integrating and running industrial and scientific applications and secure infrastructure services.

CS is firmly positioned at the top of its industry, ranking second in France for industrial and scientific applications and third in computing infrastructure services. CS assists its customers over the long term by providing solutions that are ideally suited to their needs and resources.

Its expertise and its corporate culture based on commitment and innovation have been rewarded with contracts to manage large, complex projects in a number of fields. For example, CS has been awarded substantial contracts by the Délégation Générale pour l’Armement (DGA), France’s arms agency. DGA contracts have included the €158 million CLA 2000 project to renovate the air force’s local air traffic control system and the €146 million SRSA project to develop an air-to-ground communication system, also for the French air force.

CS is also a technology partner for Airbus and its embedded software suppliers, Société Générale’s preferred supplier for distributed systems administration, the service operator for Renater 3, the French research community’s national telecommunications network, and the project manager for the future banking services network in France.

Today CS is regarded as an established service provider by its major customers, which have remained loyal for 20 years because of its employees’ expertise, commitment and customer service focus.

About the Authors

Agusti Canals is a software engineer (Université Paul SABATIER, Toulouse) and has been working at CS since 1981. Now project manager and senior software engineering consultant, he has already presented papers on HOOD, Ada, UML and object business patterns. He also currently teaches software engineering in different training structures, like Ecole Centrale Paris, Paris Dauphine or Université Paul Sabatier (Toulouse).





Stephane Lelievre is a software engineer (EERIE – Nîmes) and has been working at CS since 1998. Now project manager and software engineering consultant, he has deployed new software development process based upon generated code from models and traceability with REQTIFY™.

