

IRQA For Software Factories:

The Ultimate Requirements Engineering
Solution for Software Factories

Author: Jordi Borja, Director of Solutions Strategy

March, 2009



✓ Copyright Notice

Copyright© 2009 Visure Solutions, S.L.

All rights reserved.

This document may not be reproduced or transmitted in any way or by any electronic or mechanical means, including photocopying, recording, or any other means, without the express permission of VISURE SOLUTIONS, S.L.

All of the material contained in this document has been developed by VISURE SOLUTIONS, S.L. and is the property of VISURE SOLUTIONS, S.L.

IRQA® is a registered trademark of VISURE SOLUTIONS, S.L.

The other products mentioned are the property of their respective companies.

Visure Solutions, S.L.

C.I.F.: B-85080554

Headquarters:

Av. de los labradores, 1, 4th Floor

28760 – Tres Cantos (Madrid) Spain

Phone: +34 91 806 17 13

Fax: +34 91 804 39 50

✓ Contents

1. Requirements Engineering: A key process for Software Factories	3
2. Mature processes in Requirements Engineering	3
3. First challenge: My clients use different requirements tools	4
4. Second challenge: My clients use Word and/or Excel files for managing Requirements	6
5. Third challenge (or opportunity): My clients do not have a requirements management tool but they would not mind using the Factory's tool	8
5.1. And what about reuse?	10
6. Traceability: The best mechanism to carry out requirements analyses	11
7. This is all very well, but my factory cannot afford a requirements management tool	13
8. Conclusions	14
About Visure	14

1. Requirements Engineering: a key process for Software Factories

The ultimate objective of a Software Factory is to create systems that can address a set of needs specific to the end client¹. In Engineering, the client's functional and non-functional needs are called **Requirements**, and therefore Requirements Management and Definition processes are nothing more than **the capability to Extract, Analyze, Specify, Validate and Manage the client's needs**. Can there be anything more important for a Software Factory than truly understand the requirements the product they have to design has to meet? Although the answer to such a question should be immediate, studies show that the majority of projects developed by Software Factories fail due to incorrect Requirement Management and Definition implementation. In other words: they fail because the actual needs of the end clients have not been appropriately understood nor managed.

Regardless of your client's field of activity, there is strong pressure on providers to develop outsourcing projects with minimum risk and greatest expediency, minimum cost and the greatest possible quality. More and more companies trust Software Factories like yours because they are highly specialized in developing software and apply highly mature development processes to mitigate the risks inherent to software projects. Nevertheless, the question we have to ask ourselves, and you have to ask yourselves is: **Do we really have the skills and means for managing and understanding the needs of our clients?**

Several studies show the main cause of software projects failure (71% of all cases) is poor Requirements Development and Management. More than 40% of Software Factories have stated they experience problems related to Requirements Management, and more than half of these companies acknowledge they are not capable of correctly specifying and understanding the needs of their clients, or that they otherwise suffer from scope creep (the impossibility to clearly define the scope of a project). Do we need more information to sufficiently stress the relevance of Requirements Engineering and the attention it deserves? Several studies point out that 40% of project's time (and cost!) is spent in **"reworking"** processes (correcting already produced artifacts). More than 80% of that **"reworking"** effort is related to correcting errors associated to Requirements-related defects.

How are Software Factories addressing this challenge?

2. Mature processes in Requirements Engineering

In recent years Software Factories have invested a lot of **time and budget in actions intended to improve their software development processes**. Having the necessary maturity and experience in product development is the Factories' differentiating element. This expertise allows the end clients to allocate their resources to pursue their specific activities instead of diverting them into development processes in which they are not specialized. **Requirements Development and Management processes are key processes in Software Factories**, because they ensure the needs of the clients and their interdependence with other needs and other artifacts (tests, models,) are appropriately managed. These processes are also oriented to ensure that the final specification of a set of requirements reflects the client's needs faithfully, that validation procedures have been established for each of those needs, while also providing the option of managing the changes that will be incorporated to the developed product throughout its life cycle.

Yet, **processes alone are not enough** to ensure success in the arduous task that is Managing and Defining Requirements. Processes must be combined with **suitable process-user training and skills** and **Requirements Engineering Good Practices** (such as interview techniques, requirement writing techniques, ...), and must be supported with a **tool** that allows automating corporate processes utilization so the Factory, rather than having to depend on people's abilities can use the tool to ensure **homogenous process utilization for each of the projects involved**.

IRQA for Software Factories, allows automating corporate processes related to Requirements Engineering in a visual manner, to ensure that all projects use them correctly. *IRQA for Software Factories* will allow you to:

- **Visually define** types of requirements, change requests and associated client requests as well as validation processes (tests) and the services that specify requirement implementation.
- Create requirements, services or tests, based on previously developed functionalities by using **inherited relationships**.
- **Specify relationships between the different artifacts** defining what can be traced, who can create a trace and the reason why.

¹ Wikipedia defines Software Factories as follows: "In software engineering and enterprise software architecture, a software factory is an organizational structure that specializes in producing computer software applications or software components **according to specific, externally-defined end-user requirements through an assembly process.**"

- **Implement complex impact analyses**, such as traceability matrices established between any type of object, indirect traceability relationships or motive-related relationships by means of easily readable reports.
- **Define workflows** for client requests, requirements or tests, specifying who has clearance to execute a transition and to which status, and defining **change notifications** for your development team members or your end client.

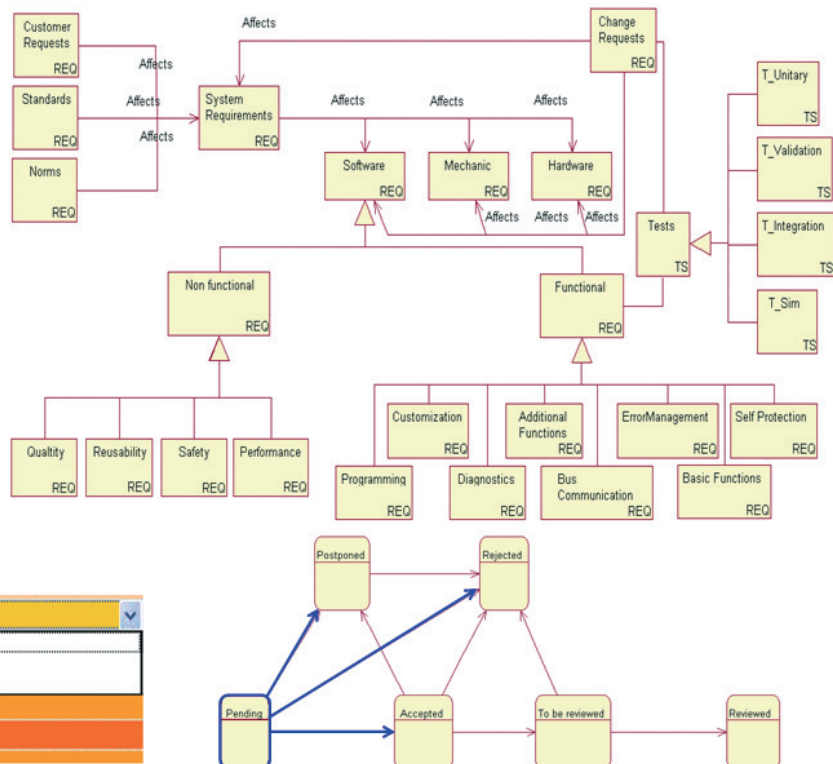
- Create **reusable project templates**, so new projects can be instantly created using your default Requirements Engineering process.
- Create **different processes for different types of projects** using existing processes as starting point.
- **Reuse requirements catalogues for different projects**, with their validation procedures and the associated services that specify how each requirement is to be implemented.

Software Factory support for CMMI processes

Define visually your corporate Requirements Engineering processes including:

- Types of requirements
- Inheritance relationships between requirements
- Traceability relationships between requirements (which requirements can be inter-traced and which user can create the trace)
- Indirect traceability paths
- Reasons for implementing trace paths
- Unidirectional or bidirectional traceability
- Define who can implement a status transition action
- Define status change notifications

Pending
Accepted
Rejected
Postponed
Accepted
Rejected
Accepted



3. First challenge: My clients use different requirements tools

One of the main problems faced by Software Factories, is that **working with vastly different clients each with diverse needs prevents them from using just one single Requirements Engineering management tool**. Each project has to face the challenge of working with the requirements management tool imposed by the client, which entails the following:

- Software Factories have to **acquire licenses for the various requirements tools**, depending on project specifications, therefore incurring a **new variable cost for each project**.
- Members of Software Factories have to **receive specific training for each of the Requirements Engineering tools**

used by their clients, which in turns **delays project start up time** and **increases variable costs**.

- **Impossibility of automating**, and therefore homogenizing, **corporate Requirements Engineering processes for each of the tools** used by the end clients, prevents the Factory from using their defined corporate Requirements Engineering processes, which are precisely the great differentiating value the *Software Factory* has to offer. These corporate processes are also the Factory's means to ensure that clients' needs are captured, specified and managed in a manner that guaranties a successful outcome of the project – thus avoiding the project from going to the failure statistics due to requirements-related defects-.

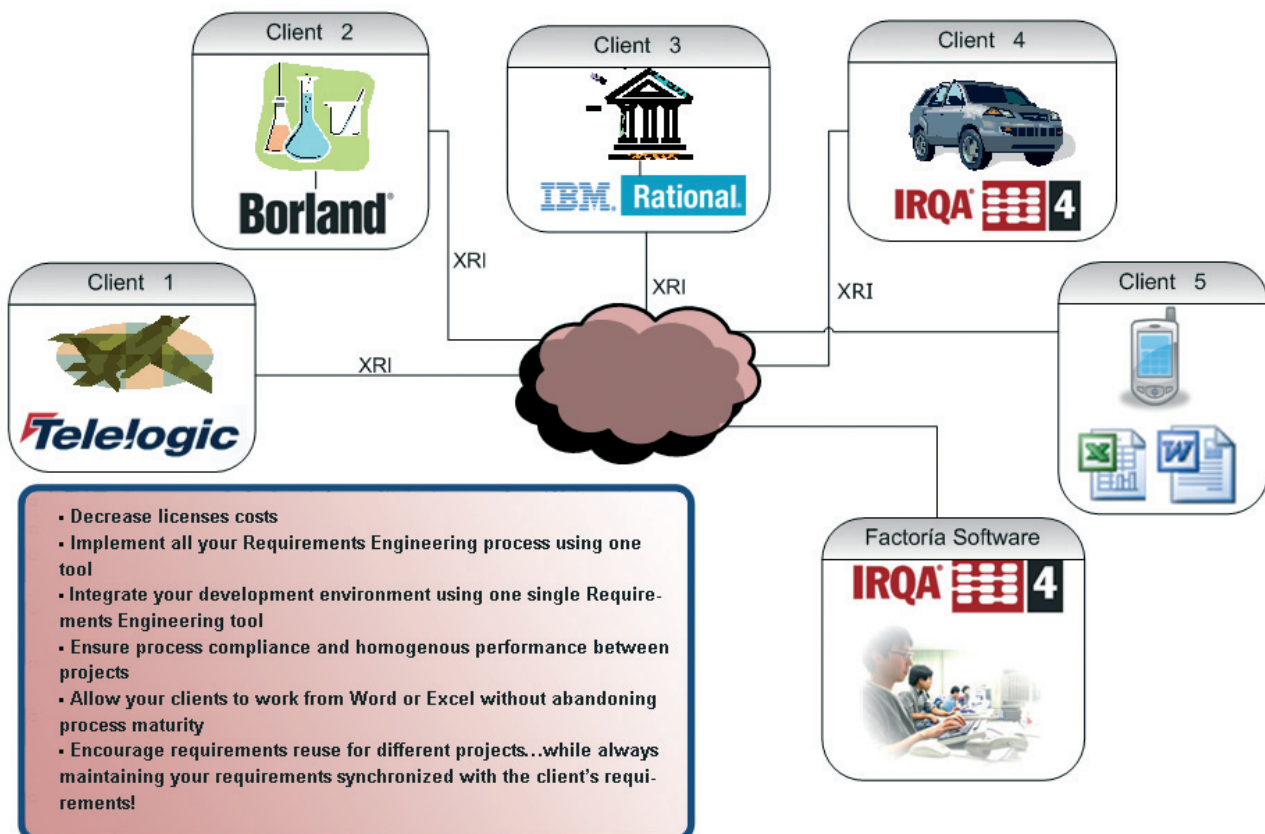
- **Difficulties when trying to integrate each of your client's requirements tools with development environments** (Modeling tools, Version and Configuration Control tools, Change Management tools, Test Management tools, ...).

- That is **impossible to apply standardized metrics, reports and dashboards** to the Requirements Engineering process because they need to be developed from scratch for each of your client's tools.

- **Difficulty using reused components for various projects while maintaining organization-wide components** (Requirements, validation procedures for requirements and for the services that specify requirements implementation as well as the different traces established between them), because said components reside in different tools.

Distributed environment

Connection to clients that use different Requirements Engineering tools from the Software Factory



IRQA for Software Factories solution makes possible for Software Factories to use **one single Requirements Engineering management tool, while keeping all requirements synchronized with the specific tools used by each of your clients.**

Using an advanced information exchange technology - designed jointly by Visure and the main companies in the automotive industry - that has been successfully tried and tested for years in this sector, IRAQ for Software Factories allows the requirements to reside, in synchrony, in the client's repository (which uses its own requirements management tool), and in the Software Factory's own requirements repository (IRQA).

Given IRAQ's advanced capability to provide support for the complex development processes previously described, **Software Factories will be able to automate their processes into one single tool.** This will facilitate corporate use of development processes for all projects while maintaining all requirements residing in the client's repository synchronized, even if the client's repository is not an IRAQ repository.

Also, the Software Factory **will be capable of integrate processes into its development environment by using any of the multiple integrations available in IRAQ**, or by using the *Integration Framework*, which easily allows creating new integrations with development tools that are not specifically supported by the system.

IRAQ for Software Factories allows integration into the organization's existing development environment. Visure understands Requirements Management and Definition as a key and complex process that is the de facto transition from Business to Technology, and therefore Requirements Engineering requires highly specialized tools such as IRAQ, rather than making do using parts of integrated suites lacking such high degree of specialization. This is why Visure has engaged in a great effort to integrate a tool specifically designed for Requirements Engineering - such as IRAQ - with the main development suites that consider Requirements Management the initial stage of development rather than a stage crucial for project success.

In addition, the **Software Factory will be able to standardize its reports, metrics and Dashboards for all projects**, working homogeneously with all of them and acquiring the capability to compare metrics between projects, regardless of the requirements management tool used by the end client.

And let's not forget the possibility of **reusing requirements, validation procedures and requirements specification services between projects**, taking advantage of the

advanced functionality of reusable components available in IRAQ.

With IRAQ for Software Factories, working with clients having different requirements tools will no longer be a problem. You can guaranty your clients that you will use your mature development process to ensure project success and mitigate risks associated to incorrect Requirements Management and Definition processes that arise from the Software Factory not being able to use their own processes. Your processes and experience are the Software Factories' great value. Your clients will thank you for using them!

4. Second Challenge: My clients use Word and/or Excel for managing requirements

Despite the numerous studies that indicate the relevance of Requirements Engineering in the development process, many end clients continue to use Word, Excel or even Outlook to implement Requirements Management and Definition processes.

This is not a problem for them because they use these applications for specific needs and do not have to manage them or apply to said Requirements analysis, specification, validation or management techniques.

Those responsibilities are the province of Software Factories: You.

Oftentimes it is difficult to convince a client to use a Requirements Management and Definition tool when its business focus is not software development. When this happens Software Factories have to accept they will receive client's requirements in a Word document or an Excel spreadsheet or even through email.

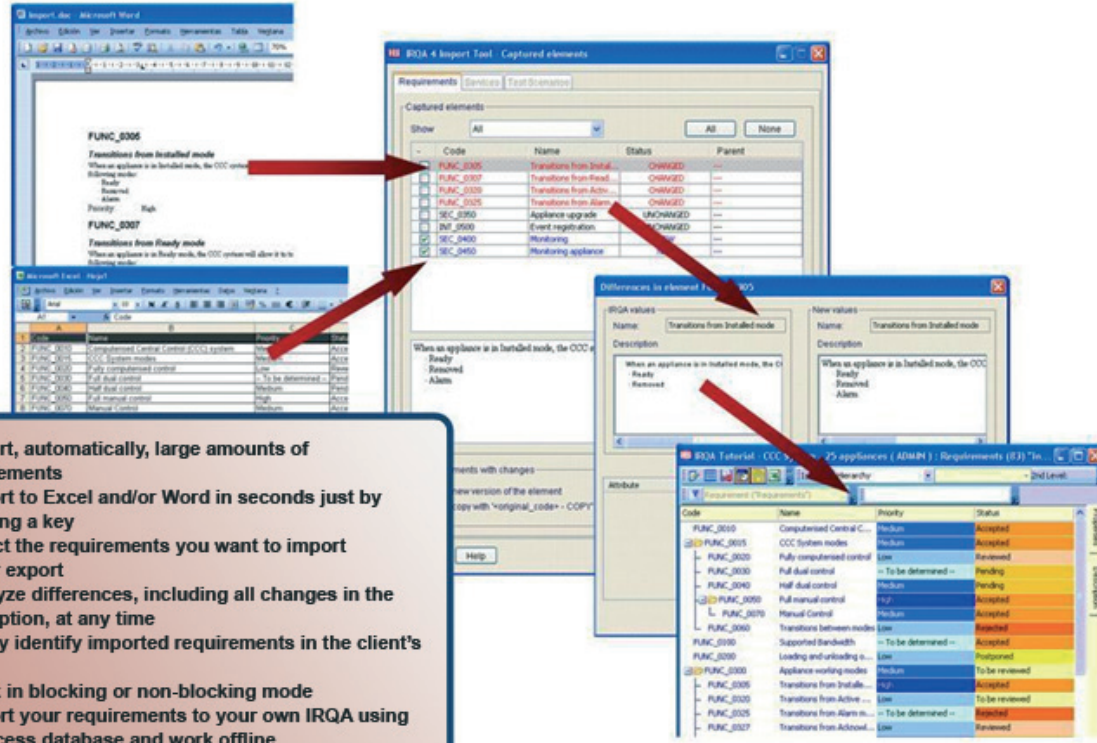
Accepting these as working formats entails giving up mature corporate Requirements Engineering processes, and therefore assuming a risk that has been time and again the habitual cause for project failure: incorrect Requirements Management and Definition processes.

With IRAQ for Software Factories, your clients will be able to continue working with their preferred Office tool, but you will not lose control of your Requirements Engineering processes.

Import initial client requirements using an advanced import tool from Word and Excel that enables you to manage the import process exceptionally, and save your data import configuration for next iterations.

Does your client want to work in Word or in Excel? No problem!

File Import/Export from or to Word and Excel with change identification and offline work functions



Import, automatically, large amounts of requirements
Export to Excel and/or Word in seconds just by pressing a key
Select the requirements you want to import and/or export
Analyze differences, including all changes in the description, at any time
Easily identify imported requirements in the client's view
Work in blocking or non-blocking mode
Export your requirements to your own IRAQ using an Access database and work offline

Work with imported requirements in IRAQ, and export them back to Word or Excel by simply pressing a key.

Your client will receive the new requirements for review, input modifications or additions and then you can import again the reviewed information to the repository using the import configuration saved for that particular client.

When importing a set of client-modified requirements IRAQ will tell you which requirements have been modified or added.

You will be able to analyze the differences between the requirement in the repository and the new version sent by the client. Once imported, you will be able to identify in the repository which requirements have been modified by the client for you to review.

In addition, with IRAQ for Software Factories, the client can implement modifications or create new requirements in **Outlook**, facilitating the essential interaction process between client and Factory.

This advanced Import/Export technology also allows Software Factories to work in either **blocking or non-blocking mode**.

If you do not want to have to go through complex processes for consolidating your different lines of work choose blocking mode. If you want to continue working and evolving requirements as they are reviewed by your clients, choose the non-blocking mode.

Using this technology gives Software Factories the additional advantage of allowing their staff to **work offline in the absence of repository connection access** using Office tools. They may even export their requirements, tests and services (or part of them as needed) to a **local IRAQ facility** using Microsoft Access as repository, and thus be able to work without online access using the advance IRAQ work environment and later synchronize all changes with the central repository.

5. Third Challenge (or opportunity): My clients do not have a requirements management tool, but would not mind to use the factory's tool

Offering your clients the possibility of working in a synchronized manner on your IRQA requirements repository hosted in your Factory is undoubtedly a **competitive advantage**. If you can convey the importance of Requirements Engineering to your client and your commitment to clearly define and manage their needs to provide a Software that has less risks and more quality you will have shown your ability to provide a competitive advantage.

The problem is that clients may be convinced, enthusiastic even, about using your requirements repository but do not want to purchase licenses for products that are not related to their main business focus. Maybe they do not want to go through the process of learning how to use a tool that may seem complex to them. With IRQA for Software Factories this

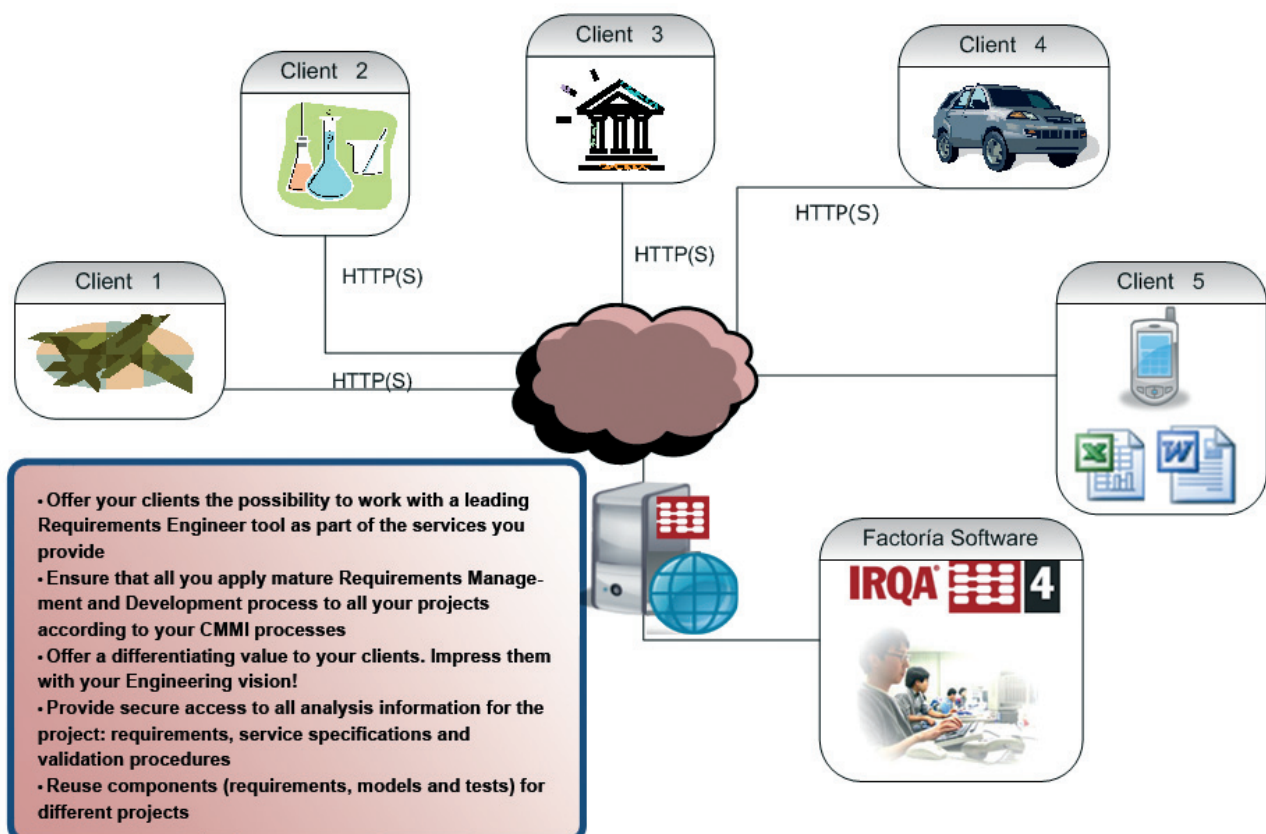
is no longer a problem for Software Factories but a business opportunity.

Offer your clients the possibility to connect to their own repository hosted by the Factory using their preferred browser through a very simple and attractive interface as part of the service package. Doing this you will make possible for **requirements, change requests, client requests, etc.... to be stored in your requirements repository**. You will also make them become part of your corporate Requirements Management and Definition process.

Using a sole repository with the client allows for **greater client-factory interaction**. It also eliminates the risk associated to working with Word or Excel documents that may be difficult for the client to maintain and manage.

Requirements Engineering as Service

Clients without Requirements Tools connected to the Requirements Repository hosted by the Software Factory



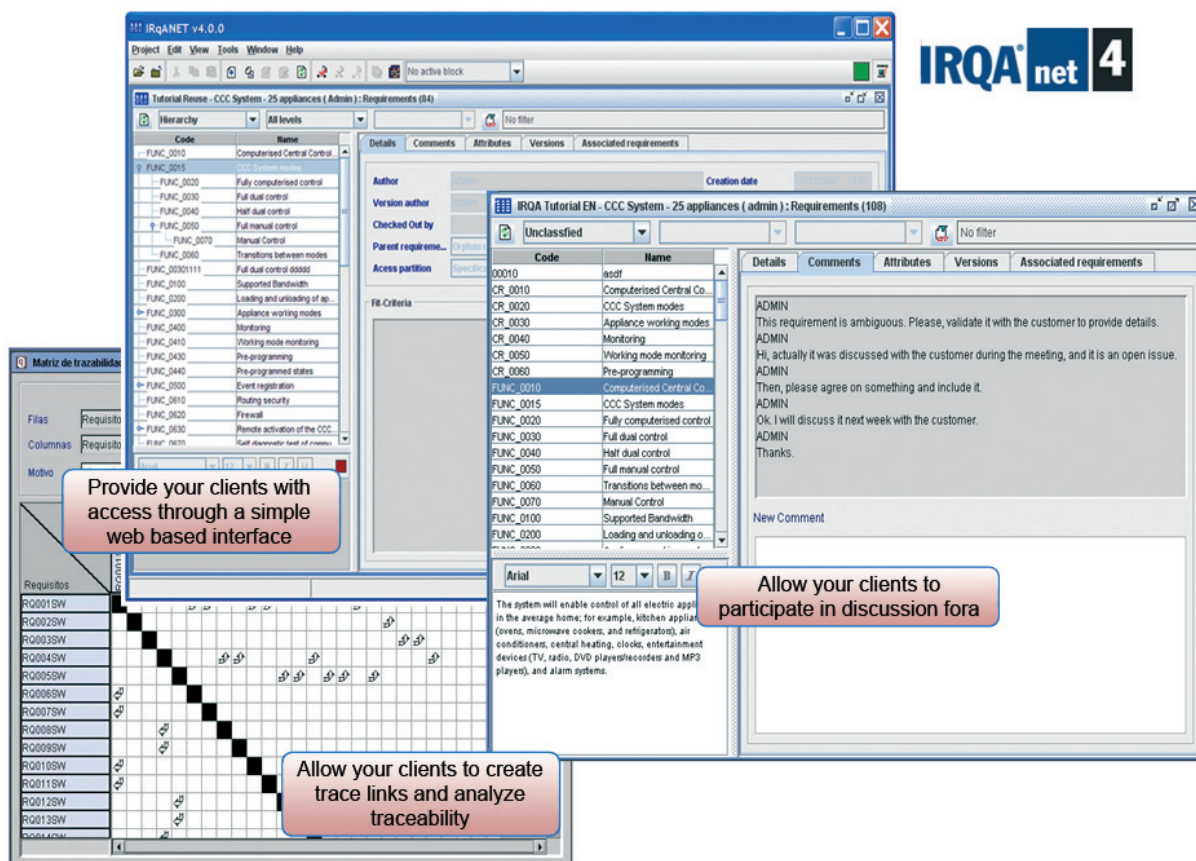
Establish trace links between client change requests and requirements (or let them do it), show them online the impact caused by a change and provide them with fully updated information on the requirements status.

IRQA for Software Factories allows clients to access the Factory's requirements repository using a browser, **without having to open up any firewall port**, through http(s), that is, through a secure connection. With IRQA for Software Factories you will be able to **control access to information** in a completely flexible manner. You will also be able to guaranty that each client can only

access their particular project, and within that project only the requirements or even the requirements' attributes that you want to grant access to, both in reading or reading/editing modes. It will also, **allow you to hide information** specific to the Factory that you do not wish to share with your clients. You can also allow your clients to participate in requirements **discussion fora** using a fully user-friendly simple environment for which prior training is not necessary.

All this while naturally meeting the restrictions specified in your Requirements Engineering process!

IRQA NET Client: Allow remote access to their Requirements



The screenshot displays the IRQA NET Client v4.0.0 interface. The main window shows a 'Hierarchy' tree on the left with a list of requirements (e.g., FUNC_0010, FUNC_0020, etc.) and their names. The right pane shows details for a selected requirement, including a 'Comments' section with a discussion forum. A 'Matriz de trazabilidad' (Traceability Matrix) is visible in the bottom left corner, showing a grid of links between requirements. The IRQA net 4 logo is in the top right corner.

Provide your clients with access through a simple web based interface

Allow your clients to participate in discussion fora

Allow your clients to create trace links and analyze traceability

5.1. And what about reuse?

As noted before, **one of the main values a Software Factory can offer is high specialization and expertise in project development.**

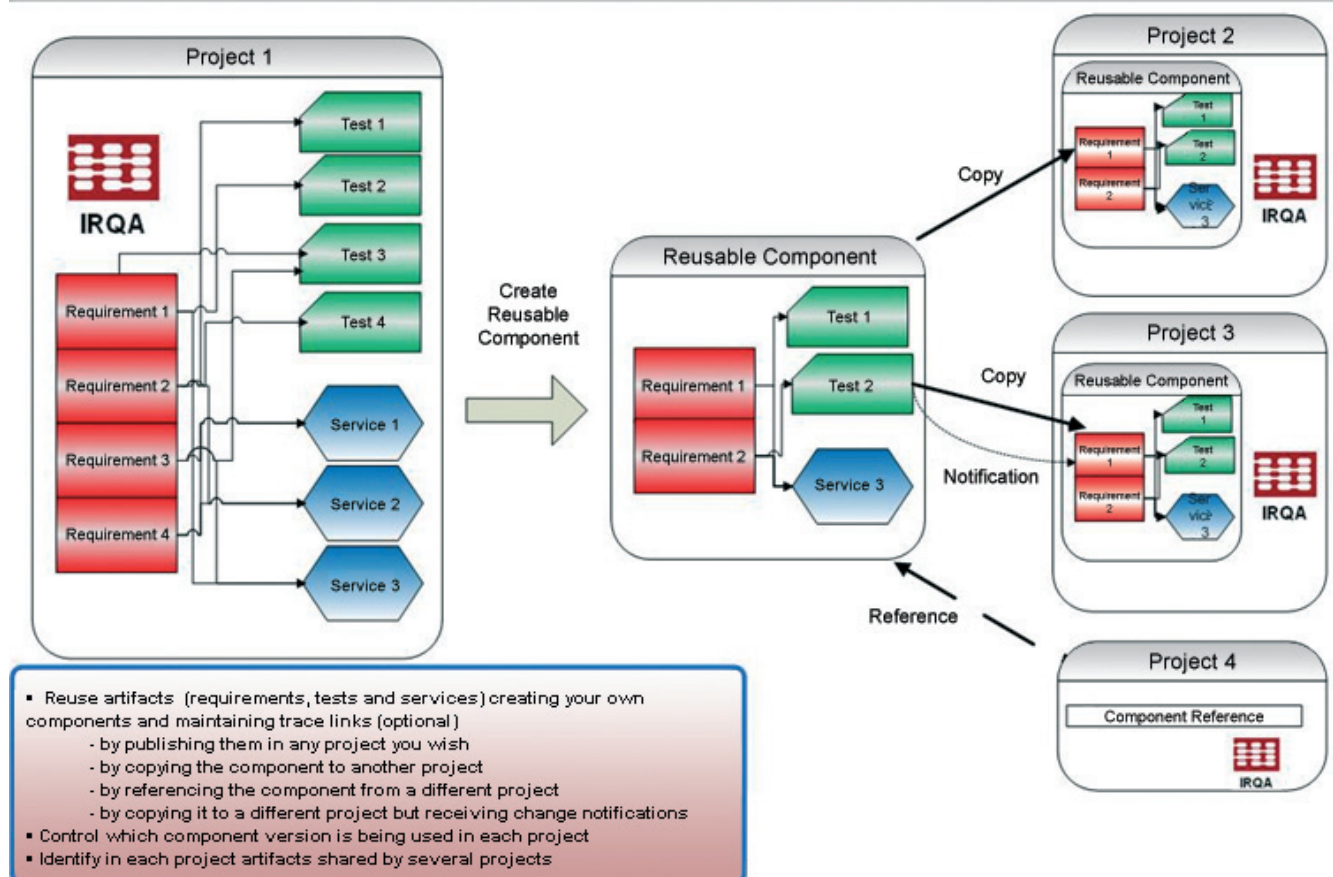
Besides the experience and training of the team members, something that characterizes a Software Factory is that they should be able to reuse components developed during prior projects, in order to not have to start every project from scratch. In other words, a Software Factory should be able to REUSE parts of prior development projects and identify which parts of their current developments can be reused in the future.

While some factories have shown a remarkable capacity for reusing technological components, even by using component repositories that can be queried for similarity, **very few factories are capable of reusing components from the actual requirements.**

One of the most important challenges for reusing components is to be able to identify which components of prior developments can be reused to cover the specific needs of a new project. So we go back to needs and to Requirements.

IRQA for Software Factories allows product development teams to **create requirement-containing reusable components, tests** (validation procedures for a set of requirements), and **services** (textual models or diagrams containing the specifications for a set of requirements). These components are then **published so they can be used in new or existing projects completely or partially.** In addition, components may optionally include traces established between requirements, or between requirements and the corresponding validation procedures and services that specify how to implement them. They may even have external references to technical components stored under Version Control in a Technological Component repository.

Reusing Components



Using this approach, **projects can search for existing and published components** and import them (total or partially) for their use. When these elements are imported projects will not only be able to reuse a set of specifications that coincides with a client need, but will also be able to use available information on the validation process for that particular component (which tests have to be run) and the services that specify its functionality presented as use cases or UML or DFD diagrams. And if you want to reuse the existing implementation, partially or totally, you can even implement this component by following the reference to the technological component under version control. **The requirement specification process will be part of the new project, but you can also reuse the tests needed to validate the new system and the details of services provided by this component.**

When a component is published it can be made accessible in three different manners, whether they have been reused is partial or total:

- **Copy reuse:** the project in which the reused component is used pastes a copy of the last available version of the given component and works with the new copy, disengaging from the reused component. In this manner any modification to the original component is not reflected in the project that is now reusing said component.
- **Reference reuse:** the project reusing the component works with a **read-only** copy synchronized to the original component. Any change to the original component is notified to all projects referencing said component so the members of the new team are aware that a change has occurred and can decide whether they want to update it or not.
- **Copy reuse and notification:** in this reuse model the projects reusing a component get a copy of said component but have to be registered to be notified of any change to the original component. Whenever such change occurs all the registered projects are alerted that a new component version is available. At this point a decision has to be made to update the version, use only parts of the new component or continue working with the older version.

These three reuse mechanisms allow each project – in addition to provide them with the capacity to partially reuse a component – to choose the **best reuse strategy** depending on the specific needs of the project and the type of component in question.

Also, **components may be published so they are available for use only in certain reuse modes.** For instance, a set of requirements related to the Official Data Protection Act could only be shared in reference mode to ensure that all projects use updated legislation whenever the law is amended. In the same manner a functional printing component of an online document can be reused by several projects, in this case by sharing printing requirements, the validation process (those tests intended to verify the given component, in this case the printing component), and the use cases showing how this component should work functionally-wise. All these options are provided including all the trace links.

Reusing requirements is a key practice to **create product families**, that is, similar projects featured in different versions which only variation is a small set of requirements.

IRQA for Software Factories, also **lets you know which component and which version are being reused and in which projects.** This functionality is in place so you can better estimate the impact of any change on the given component.

6. Traceability: The best mechanism to carry out requirements analyses

Traceability is, without a doubt, one of the challenges faced by any software development process, and more so for Software Factories. Traceability is not an end in itself. Rather, it is the **mechanism used in Requirements Engineering to carry out dependency analyses between requirements and between requirements and other development artifacts.** This dependency analysis allows us to solve problems through, for instance, **change impact analysis**, finding out which change requests have been transformed in the requirements, which change requests have already been developed and used, which parts of the development are not associated to any validation tests...

Most requirements solutions provide mechanisms for tracing Requirements with Requirements, and Requirements with other development elements, such as tests or services. However, **many requirements solutions do not provide useful and simple tools and views to carry out the dependency analyses** just mentioned, or they do it partially, focusing on the mechanism (creating trace links) rather than on the objective (be able to carry out a trace analyses).

For Software Factories what is especially relevant is to be able to do dependency analyses. Knowing the impact a new client request is going to have in terms of time, effort and cost; knowing which requests are already in the development stage; knowing which change requests have been successfully tried; which requirements do not have associated tests and therefore no associated validation processes yet exist....

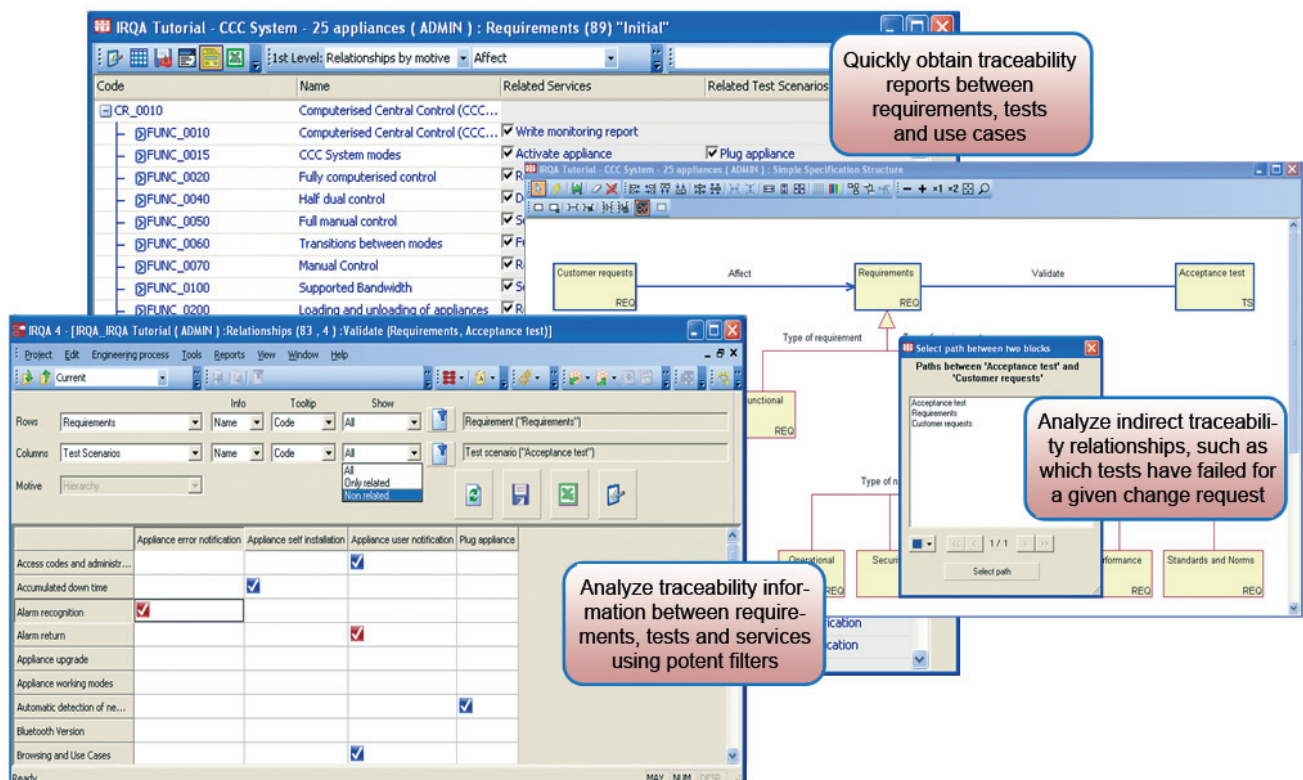
First, **not all elements can be inter-traced**. Limiting traceability between elements to what really makes sense according to the Requirements Engineering corporate procedure is crucial. For instance, it does not make sense to establish a direct trace between a change request with a test if no related requirement and service exist to specify how to execute a change request. Likewise, it probably makes no sense for your organization to trace a performance requirement with a Unitary test. **IRQA for Software Factories allows you to specify which trace links can be implemented and which profiles can implement them.**

Secondly, **traceability matrices usually show direct relationships** between two types of elements. If a change request is traced back to a System Requirement and this Requirement is in turn traced back to a set of tests and services, a Software Factory will have to analyze which client's change requests are associated to which tests in order to learn which requests have been already traced and implemented. This is what is known in Requirements Engineering lingo as **indirect** traceability.

Many tools available in the market would require obtaining a traceability matrix between change requests and requirements, annotate all requirements associated to a change request, and later generate a new traceability matrix between requirements and tests to annotate, one by one, all the traced tests with each of the requirements already traced back to the change request. **With IRQA for Software Factories, indirect dependencies analyses are done automatically, choosing the indirect traceability path between the two elements under analysis.**

Traceability and Impact Analysis

Trace change requests, requirements, tests and services and analyze change impact



Thirdly, traceability matrices are bi-dimensional. They show relationships between two sets of elements. However, many analyses require more than one dimension. For instance, being able to see which requirements are associated to a change request, while also being able to see the validation tests associated to said requirement and the specified services. In these cases typical traceability diagrams are useful when the number of requirements, tests and associated services is small, but in real projects where the number of traced artifacts is large, they quickly become unmanageable. IRQA for Software Factories provides a significant number of views that allow multiple dependency

analysis while consolidating in one view all the required information that can be easily browsed using a pull down-tree menu structure.

The great traceability power provided by IRQA resides in its ability to incorporate service concepts (models, use cases) and tests in the tool itself. Whether the services or the tests are defined in the tool itself or they are defined in specialized Modeling or Tests tools, IRQA keeps both services and tests synchronized in its repository, therefore providing a traceability analysis power unmatched in the existing market.

One single tool for Requirements, Tests and Services

Define services that specify requirement implementations through textual use cases or diagrams. Or integrate into your corporate modeling tool

Code	Name	Test Status
TST_010	Connect appliance	Pending
TST_020	Correct appliance connection	Passed
TST_030	Incorrect appliance connection	Failed
TST_040	Self diagnostic test	Pending
TST_050	Plug appliance	Passed
TST_060	Appliance self installation	Failed
TST_070	Appliance user notification	Passed
TST_080	Appliance error notification	Pending

Define your Validation procedures in IRQA, or integrate information into your corporate Test Management tool

Id Actor Event Direction

- 1 User The user presses the 'Connect appliance' option in the screen.
- 2 User The system provides the possibility to select the initial working mode of the appliance.
- 3 User The user selects the working mode for the appliance being connected.

7. This is all very well, but my factory cannot afford a requirements management tool

Visure is aware that one of the main objectives and competitive value a Software Factory can offer is price. Yet price considerations cannot undermine the competitive ability of getting things right the first time. Incorporating a requirements management tool in a Software Factory works generates advantages that far outweigh its cost. Your prestige and your client satisfaction are at stake. Statistics show that doing nothing about it is not enough.

Still, the costs are a real concern for Software Factories. IRQA for Software Factories is a custom made solution designed by Visure for Software Factories that takes into account the cost savings principle so important for Factories, allowing those Factories to extract maximum benefits from using a tool such as IRQA, and achieving the desired effect of garnering maximum client satisfaction.

It is Visure's goal to enable Software Factories to benefit from this custom-designed solution. In order to do this, we want to offer the greatest flexibility in economic terms to factor in the individual needs and possibilities of each factory. Scenarios such as **per project license leases, usage based payment, combining temporary and permanent license agreements...** are all possible with IRQA for Software Factories. Our objective is that price not be the problem if IRQA for Software Factories is actually the solution Factory needs.

8. Conclusions

Besides allowing your company significant savings when acquiring multiple tools used by your clients and reducing personnel training needs to a minimum, IRQA for Software Factories incorporates in one single tool all the options for defining and managing tests, and the possibility to define services, including UML and DFD diagrams. IRQA for Software Factories is not intended as a modeling or a test management tool, just as it is not our objective to compete in these markets. In fact, IRQA integrates its tests with the main tools specialized in tests, as well as with the services offered by the main UML modeling tools. Yet, many Software Factories do not use Modeling tools or tests in which case IRQA for Software Factories can be an appropriate tool to introduce practices related to those Process Areas in your organization as first steps in a process in which more specialized tools will be gradually introduced in the future.

IRQA for Software Factories also provides endless additional functionalities, such as baseline management, requirement-concurrent access control, documentary views for those that prefer working in a Word-like environment without compromising the Requirements Engineering process, pre-defined processes and a powerful tool for creating new reports and metrics, an advanced security control system, change and version control, filters and customizable and reusable views between projects, RFP and provider management support, commercial database support, tool and support software available in Castilian Spanish, ...

Visure Solutions, "The Requirements Company", also offers highly specialized training courses in Requirements Engineering to enable your Software Factory personnel to become proficient in Requirements Capture Techniques, Human Skills for Requirements Engineering, How to write requirements in correct Spanish language,

About Visure

Visure Solutions, "The Requirements Company", is a company 100% financed with Spanish capital. It is specialized in Requirements Management and Engineering. Our objective is to become the technological leader in the industry. The quality of our IRQA product, vouchsafed by leading organizations worldwide and our team's ample experience in developing, marketing and implementing our product are our main company assets.

Visure Solutions goal is to further innovation by incorporating the last Requirements Engineering techniques to our IRQA solution so our clients can ensure product and system quality from the inception stage. Client orientation will continue to be our fundamental guiding value, and therefore our focus will be concentrated on providing those services our clients need to guaranty suitable personnel training, improved Requirements Management processes and implementing our products.

Our headquarters offices are located in Tres Cantos, Madrid. We have distributors deployed all over the world in countries like Germany, Brazil, Italy, India, Switzerland, the SA, etc.

Some of our more relevant clients are AUDI, Deutsche Post, BOSCH, ST Microelectronics, Knorr Bremse, Metro de Madrid, PharmaMar, Repsol YPF, Sistemas Técnicos de Loterías (STL), Telefónica I+D, amongst others.
I+D, etc.

For more information about the solution

Jordi Borja Sanz.
Director of Solution Strategy
jborja@visuresolutions.com

C/ Labradores, 1; 4th floor
28760 Tres Cantos (Madrid)
Telephone: +34 91 806 17 13
Mobile: +34 679 198 168
Fax: +34 91 804 39 50



Avda. de los Labradores, 1, 4th Floor
28760 Tres Cantos, Madrid (Spain)
Tel: +34 91 806 17 13 Fax: +34 91 804 39 50
info@visuresolutions.com
www.visuresolutions.com