

Managing Requirements & Improving Quality

TraceCloud

By

Shambhavi Roy

Product Manager, TraceCloud

sami@tracecloud.com

www.tracecloud.com

Table of Contents

Synopsis	3
What are Requirements?	3
Why manage Requirements?	4
A Requirements Management Plan (RMP)	4
Requirement Types	4
Traceability (Relationship) between Requirement Types	5
Folders	6
Roles	7
Permissions	7
Project Lifecycle Actions	8
Project Structure	9
Define Requirements	10
Set Requirement Attributes	10
Collaborate & Review Requirements	11
Approve Requirements	12
Trace (Relate) Requirements	12
Manage Change	14
Baselining	14
Change in Bulk	15
Track Progress	16
Report on Requirements	17
List Reports	17
Trace Tree Reports	18
MS Word Reports	18
Scalability, Usability, Performance, TCO, Open Formats	19
About Us	19

Synopsis

Failure to define, communicate, and validate business requirements is the root cause of many project delays, overruns and failures. There is a growing realization of the importance of requirements in organizations. The goal of this paper is to show how TraceCloud can be used to collaborate, define, approve, control and track requirements to have a high quality end product.

What are requirements?

Requirement is the industry term used to describe end user needs or the level of service expected from an application. Requirements tend to vary in granularity and meaning and can be categorized in different ways.

- **Business requirements.** They describe the overall objectives of the system, or how it'll affect the business.
- **User requirements.** They describe user needs and problems.
- **Functional requirements.** They describe specific behaviors of the system and the internal workings of the software.
- **Quality of service requirements.** They describe localizability, accessibility, performance and scalability.

In a typical project, different organizations own different types of requirements. For example:

- Product Marketing, Business Analysts, Program Managers may define business requirements. They, along with the project and release managers, may group these requirements in a release.
- Development managers may define functional requirements and associate them with business requirements
- Test managers may define test cases and assign them to functional requirements. Typically, they'll also own the execution of test cases and the tracking of test results to completion.

While the above example is typical in most software development environments, it may differ based on the organizational needs. Hence, at the beginning of a project, it's important to understand and define a requirements management plan (RMP).

Why manage requirements?

Any time you analyze the root cause of a failed project, failure to define, track and manage requirements and constantly changing requirements are the prime culprits. If you investigate some of the failed projects, you may recognize the following symptoms:

- Lack of clearly defined requirements
- Lack of a clear relationship between different types of requirements. That problem resulted in
 - Changes to business requirements that were not communicated to developers
 - Changes to functional requirements that QA was not aware of and did not test
- Unapproved/unauthorized changes to requirements
- Lack of visibility of overall release status
- Lack of ownership of requirements
- Inability to report on requirements and their status

A requirements management tool and a requirements management plan can go a long way towards solving the basic problems. In the next section, we'll see how a requirements management plan (RMP) can get different stakeholders on the same page.

A requirements management plan (RMP)

A properly defined requirements management plan is a good way to document the process and get the acceptance of all the participants. An RMP includes the following:

- Type of requirements you plan track
- The attributes that are relevant to the requirement types
- The folder structure for storing requirements
- The roles of people who'll work on the requirements
- Who owns which requirement types and what actions can they perform
- Actions to be performed at different points in the project life cycle

Here is a typical RMP for a software development project. You may modify this to suit your needs.

Requirement types

- Release :
 - A high level requirement that tracks a collection of features being delivered. It may be called a Release, a Sprint or an Iteration, and it usually goes through the typical SDLC milestones.
 - Attributes
 - Concept commit date
 - Execute commit date

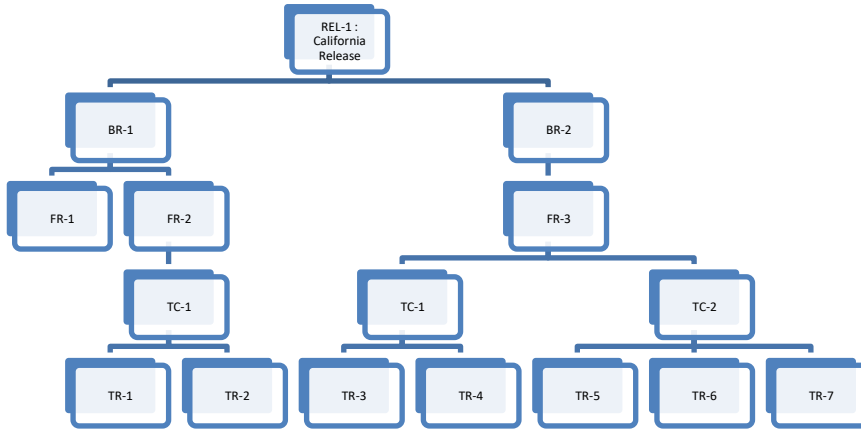
- Delivery commit date
 - QA hand off date
 - Go live date
 - Release manager
- Owner
 - Release manager
- Business requirements:
 - This requirement type defines features to be implemented. Usually Business Analysts or Product Managers define business requirements. You may have also heard of this being referred to as Use Cases.
 - Attributes
 - Impact to Customer: Drop Down : yes, no
 - Requesting Customer: Text Box
 - Customer Commit Date: Date
 - Priority : Drop Down: 1,2,3,4,5,6
 - Owner
 - Product Marketing / Product Managers / Business Analysts
- Functional Requirements :
 - They are defined by the development team and have technical/implementation details. Every business requirement can be broken down into multiple functional requirements
 - Attributes
 - Effort: Number: Number of person days
 - Percent complete : Number
 - Developer : Email Id
 - Owner
 - Development Engineer
- Test Cases:
 - Test Cases are written by the QA team to test functional requirements.
 - Attributes
 - Server configuration: Text
 - HP quality center Id: Number
 - Percent complete : Number
 - Owner
 - QA Engineer
- Test Result :
 - They track the results of test iterations
 - Attributes
 - Pass: Drop down: yes/no
 - Percent complete: Number
 - Owner
 - QA Engineer

Traceability between requirement types

- A release includes one or more business requirements
- A business requirement includes one or more functional requirements
- A function requirement includes one or more test cases

- A test case includes one or more test results

A Trace Tree looks like this:



Folders

We have a high level folder for every requirement type, and that folder and its sub-folders contain requirements of that type. A typical folder structure looks like this:



Roles

In most organizations, we have multiple groups of stakeholders.

- Release managers
- Product managers (Business analysts, Product marketing)
- Development managers
- Developers
- Quality Assurance

Depending on the project's complexity, you may have fewer or more roles. In a small project with fewer than 500 requirements, an administrator and one or two other user roles may suffice. However, if you are dealing with a project that has thousands of requirements and a large number of users, it helps to define roles and assign these roles to individuals. Each group owns and is interested in some specific aspects of the SDLC.

Permissions

Once we have requirement types, folders and roles, we can assign permissions to roles at the folder level. The table below documents actions that a role can perform in particular folders. According to the first row, a release manager can create, edit, delete, trace and approve requirements in the 'California Release' folder.

Folder Name	Create	Edit	Delete	Trace	Approve
California Release	Release Manager	Release Manager	Release Manager	Release Manager	Release Manager
Oregon Release	Release Manager	Release Manager	Release Manager	Release Manager	Release Manager
California Release BRs	Product Manager	Product Manager	Product Manager	Product Manager	Product Manager
Oregon Release BRs	Product Manager	Product Manager	Product Manager	Product Manager	Product Manager

California Release FRs	Development Manager	Developers	Development Manager	Developers	Development Manager
Oregon Release FRs	Development Manager	Developers	Development Manager	Developers	Development Manager
California Release TCs	QA	QA	QA	QA	QA
Oregon Release TCs	QA	QA	QA	QA	QA
California Release TRs	QA	QA	QA	QA	QA
Oregon Release TRs	QA	QA	QA	QA	QA

Project lifecycle actions

To ensure high project quality and avoid missed/misunderstood requirements, every stakeholder performs certain actions at various stages in a project’s lifecycle.

Mile Stone	Expected Action
Release kick off	A release is created
Concept commit	<ul style="list-style-type: none"> Product Manager creates the business requirements and works with the Release Manager to trace these BRs to the release object. Product Managers and Release Managers ensure that there are no suspect traces between releases and business requirements.

Execute commit

- Developers study the business requirements and build functional requirements to address the BRs.
- The functional requirements are vetted and approved and traced up to the business requirements.
- At execute commit, both Product Managers and Development Managers ensure that
 - there are no dangling business requirements in the release (i.e. missed requirements)
 - there are no orphan functional requirements (i.e. functional requirements that don't address any business needs)
 - there are no suspect traces between business requirements and functional requirements (i.e. all change have been accepted by both sides)

Development Complete

- Before the development is complete, test cases trace to functional requirements or business requirements
- QA, development managers and product managers ensure that
 - there are no dangling business and functional requirements (i.e. those not covered by a test case)
 - there are no orphan test cases (i.e. test cases that don't necessarily test any functionality)
 - there are no suspect traces between test cases, BRs and FRs

Go live

- Before go live, QA team has tested all the test cases and created test results
- All test results trace to test cases
- QA, Development Managers, Product Managers and Release Managers ensure that
 - there are no dangling business and functional requirements and test cases
 - all test results have passed
 - all test cases , FRs, BRs are 100% complete
 - there are no suspect traces between test results, test cases, FRs, BRs and the release.

Project Structure

Once we have a requirements management plan, we go ahead and create a project

Define requirements

Once the project structure is defined, we can go ahead and create requirements. There are three ways to create requirements.

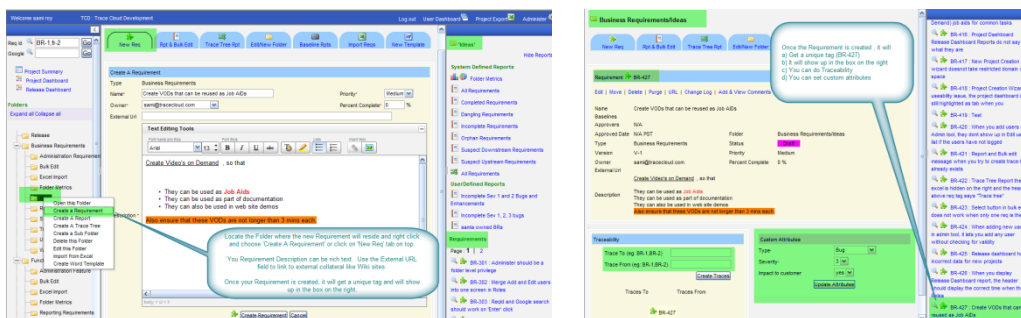
1. create requirements one at a time
2. import from excel
3. import from a word document (a business requirement document or a product requirement document or a system functional specification document). The imported text is parsed to create requirements.

Once the requirements are defined, we set the attributes

1. attributes can be set one at a time for each project
2. imported from excel
3. created using the bulk edit feature of the tool

During the definition process, we can request user feedback and discuss requirements. Once the requirements reach a certain level of maturity, we can submit them for approval.

While there is no set rule as to when the requirement gathering for each requirement type starts, it typically follows this hierarchy: release requirements, business requirements, functional requirements, test cases and test results. [Demo Video](#)



Creating requirements in bulk using excel. [Demo Video](#)

Set requirement attributes

This step may be optional depending on your project needs.

Requirements never exist in isolation. While defining requirements is useful, to get the most bang for our buck we also want to define attributes that are relevant to requirements. For example, for business requirements, it may be important to track the requesting customer, the promised date and/or other attributes. Similarly for functional specifications, we may want to track dependencies, estimates, delivery dates etc.

Which attributes are relevant to your requirement types is something you define as part of your RMP (requirements management plan). As part of this effort you may define

- which attribute should have a list of values
- which attributes are mandatory
- which attributes get default values

At the minimum a good requirements management tool should let you define your custom attribute types, set default values and set attributes to mandatory/optional. As a lot of folks use MS Excel, it would help requirements management tools should support the use of Excel to manage and update attributes in bulk.

Collaborate & Review Requirements

This step may be optional depending on your project needs.

Once you have defined requirements and attributes, you want to discuss the details with subject matter experts and build on the tribal knowledge within the organization. The best way to do this is by

- grouping similar requirements together
- sharing requirements with your SMEs
- providing an easy forum for people to express their opinions
- saving ideas for future reference
- providing a search feature for requirements
- keeping the stakeholders (owners, approvers and reviewers) in a constant communication loop
- taking care to not overwhelm the stakeholders by spamming them with a deluge of irrelevant information

TraceCloud lets you

- logically group requirements in folders
- download Excel reports that can be mailed out to stakeholders
- provide hyperlinks in reports so reviewers can easily add comments

- search through requirements and comments
- retain comments as long as the requirement is not deleted
- deliver consolidated emails to share feedback with all stakeholders.

Approve Requirements

This step may be optional depending on your project needs.

Once you have defined requirements and received feedback from users, you may want to send requirements through a formal approval process. Ideally, you would define the approval process as part of your RMP (requirements management plan), as an event linked to your development cycle. Some examples are:

At the concept commit stage, when you commit to **what** you'll deliver in the release, you may get your business requirements approved by product managers and the working committee.

At execute commit stage, when you commit to **when** you'll deliver the features and at **what cost**, you may want to get your functional specifications approved by product managers, architects and QA team.

Please note that **who** needs to approve, **which** requirements and **when** should be completely driven by business needs. For guidance in setting up your RMP, please reach out to TraceCloud.

The approval process includes

- defining an approval path: which users need to approve which requirements
- Grouping Requirements and submitting them for Approval
- Tracking the progress of Approval and providing any required additional information
- Making the requested changes and resubmitting
- Once the approval is done, locking the folder so as to prevent any un-authorized changes.

Here are few brief videos that describe how you can achieve the above tasks using the TraceCloud system.

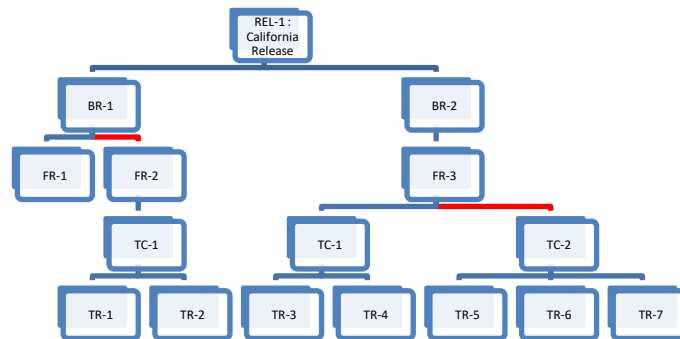
Managing Approval Work Flow Video TODO

Lock down folder Video TODO

Trace (Relate) Requirements

Traceability is a powerful feature that lets you relate one group of Requirements to another and lets you track the changes and dependencies between these Requirement groups. For example, as part of your Requirement Management Plan, you could decide to Trace (Relate) all your Business Requirements to a Release, and all your Functional Requirements to your Business Requirements.

So, your trace tree looks like this...



In the above example , you can visually figure out that,

- The California Release (Rel-1), will have Business Requirements BR-1, and BR-2 in it.
- You can also see that Functional Requirement FR-1 is not tested. Hence the quality of BR-1 may not be very good. In Requirements Management lingo, this is called a Dangling Requirement.
- You can see that Functional Requirement FR-3 is thoroughly tested by 2 separate test cases.
- You can see that BR-1 to FR-2 trace is suspect. This indicates a change between these two requirements that has not been acknowledged by the BR-1 and FR-2 owners.

A good Requirements Management tool will let you manage traceability by

- Permitting Traceability at the point of Requirement Definition
- Permitting traceability in Bulk
- Traceability using popular tools like Excel and Word
- It should let you view your traceability data both graphically and through tools like Excel , Word
- It should let you take a snapshot of Traceability at a point in time, for your auditing needs.
- It should let you easily generate reports that show where the change has happened, what its impact is and what can be done to manage and mitigate the issues arising from this.

Here are a few brief videos that cover some of those topics.

Creating and Managing Traceability Video TODO

Manage Change

In a long running project, with multiple users playing different roles, 'Change' is a certainty. What is also true is that Change is a constant source of friction between different user groups. In order to effectively manage a large project, it's important to acknowledge and manage the change.

There are times in a project life cycle, when flux is encouraged. For example, at the time of Requirements Definition, you want your Business Requirements to go through flux as people brainstorm, collaborate and refine their needs. However, as the project crosses over into the execution phase, you want your Business Requirements to be stable, and any change to happen only after the right people have approved it. Even this change should be such that, any stake holders (upstream, downstream Requirement owners) should be able to quickly locate and address it.

A good Requirement Management tool should let you create folders, and set access permissions at a folder level, so that you can control, who can Create, Read, Write, Update and Trace to / from Requirements in this folder. With this model you could

- Create a Folder and let the right group of people create Requirements in this folder
- Submit these requirements for review / feedback
- Get the right set of people to approve these requirements
- Once the project moves to execution phase, you can remove the Update / Delete / Create / Trace To & From permissions from everyone.
- Essentially this folder moves into a lockdown phase
- You can limit the Update access to a set of process people, who after the right Change Control Meetings can update these Requirements / create new Requirements.
- Any change to these Requirements will trigger suspect traces, both upstream and downstream. Hence the stakeholders working on related requirements can take appropriate action.

Here are a few short videos that describe these operations

Folder level permissions and Locking down a folder Video [TODO](#)

Folder Dashboard, Reports, Metrics and Trends Video [TODO](#)

Baselining

How often have you had discussion with your counterpart in Product marketing / Development / QA, where you wished you had a copy of what you both had agreed to in the past. You are absolutely certain that the other party is wrong, and if only you had a way to prove it.

Baselining is a concept that helps you avoid such pitfalls. One way to understand a Baseline is to visualize the group picture you took at your high school graduation with your buddies. It's a snapshot of a collection of people at a certain point in time. With the help of this snapshot, you can

- Figure out who were on your buddy list on that date
- What they looked like then
- Who have dropped out of your buddy list since then
- Who are the new people you have added to your buddy list since then
- How have the people in the snapshot changed over time
- How did their luck / fortunes / waistlines fare over time.

Baselining helps you solve a similar problem in the engineering world. In the engineering world, Baselining is an 'Arbitrary grouping of Requirements at a certain point in time'.

When you create a Baseline and add Requirements to it, it helps you

- Figure out which Requirements are part of this Baseline
- What did the Requirements look like when they were added to this Baseline
- How have these Requirements changed since they were added to this Baseline
- How do different parameters of these Requirements trend over time. Eg : Completed, Incomplete, Dangling, Orphan etc...

A good Requirements Management Tool should let you group Requirements into a Baseline, track them as an entity and let you compare the Baseline copy with the current state of affairs.

Here are a few quick videos that show how TraceCloud has implemented Baselining.

Creating Baseline and Leveraging information in Baselines Video TODO

Baseline Dashboard, Reports, Metrics and Trends Video TODO

Change in Bulk

While modifying each Requirement one at a time is feasible in small projects, this tends to quickly breakdown as your project starts taking on 100s, if not 1000s of Requirements. To handle such a scenario, a good Requirements Management tool should have a rich bulk editing capability. It can be a UI based bulk editor (i.e. select a group of Requirements and perform one operation at a time) or it can be something that lets you export in to Excel, perform your change and re-import these Requirements.

A good Requirements Management tool should let you perform some of these operations in bulk...

- Delete & Restore Requirements
- Purge Requirements
- Move Requirements to different folders

- Change Owners
- Set standard attributes (% complete, Priority etc..)
- Add to / Remove from Baselines
- Update Custom attributes
- Manage traceability
 - Create Traces
 - Delete Traces
 - Clear Traces
- Approval work flow
 - Submit for approval
 - Approve Requirements
 - Reject Requirements

Here is a video that describes how TraceCloud lets you perform these operations in bulk

Using Bulk Editor to modify large volumes of data Video TODO

Using Excel to Edit & Upload large volumes of data Video TODO

Track Progress

As your project goes through multiple releases, the number of Requirements you track will continue to increase. In such a scenario, the tool should help you easily identify problem areas and help you track progress. For example, you should be able to easily answer questions like

- How many incomplete Business Requirements exists
 - in this Folder
 - in this Release
 - in this Project
 - owned by a user
 - in a Baseline
- Quickly identify all Dangling Functional Requirements (i.e no downstream Requirements)
 - in this Folder
 - in this Release
 - in this Project
 - owned by a user
 - in a Baseline
- Quickly create an Excel file of all Requirements pending approval in
 - in this Folder
 - in this Release
 - in this Project
 - owned by a user
 - in a Baseline
- Show the trending of completion of Requirements in
 - in this Folder
 - in this Release

- in this Project
- owned by a user
- in a Baseline

The benefit of such a system is that the depending on your role, you may be interested in a different perspective of the project.

For example, a Release manager may be interested in the metrics of a Release and be able to quickly find trouble spots in the Release. A Product Manager may simply be interested in tracking all the Business Requirements that he has taken a baseline of. A Development Engineer may only be interested in all the Requirements assigned to him.

TraceCloud has a rich collage of Dashboards that you can leverage. For example, when you log in, you can

- See the Project Dashboard to track the progress of the entire Project
- See the Release Dashboard to track the progress of any given Release
- See the User Dashboard to track the progress of Requirements that you own
- See the Baseline Dashboard to track the progress of an arbitrary grouping of Requirements
- See the Folder Dashboard to track the progress of all Requirements in this Folder.

Here are few videos that show how this is done.

Project Dashboard Video TODO

Release Dashboard Video TODO

User Dashboard Video TODO

Baseline Dashboard Video TODO

Folder Dashboard Video TODO

Report on Requirements

In any project, there is typically a need to generate reports , either from an auditing / documentation perspective, or from a operations review perspective. A Good Requirements Management solution should let you query and generate reports in multiple output formats. Some common reporting criteria are ...

List Reports

There are reports where you apply a filtering query, and generate a table of data of all Requirements that match the filtering criteria. These reports should let you filter on common scenario's like Dangling,

Orphan, Completed, Incomplete, Suspect, Pending, Approved, Rejected, Owner, Folder or filter on any standard or custom attributes.

These reports should also let you configure which attributes you want displayed, and once you generate the report, you should be able to save these reports as either Public (visible to any one) or Private (visible only to you).

It also helps to be able output the data in to Word, Excel, PDF formats. Another Requirement could be to be able to link back to the original source requirement from these static reports (Excel, Word, PDF).

Here is a video that describes how TraceCloud implements List Reports. TODO

Trace Tree Reports

These are reports where you apply a filtering query, and generate a hierarchical view of Requirements that match the filtering criteria and show their relationship to other Requirements in the project.

Trace Tree reports are a good visual indicator of the status of relationships between different requirements in your project. Depth of a Trace Tree indicates how many levels of these relationships you are interested in, and this should be a filterable criteria.

These reports should let you filter on common scenarios like Dangling, Orphan, Completed, Incomplete, Suspect, Pending, Approved, Rejected, Owner, Folder or filter on any standard or custom attributes.

These reports should also let you configure which attributes you want displayed, and once you generate the report, you should be able to save these reports as either Public (visible to any one) or Private (visible only to you).

It also helps to be able output the data into in to Word, Excel, PDF formats. Another Requirement could be to be able to link back to the original source requirement from the static reports (Excel, Word, PDF).

One thing to look for is that your Trace Tree output document should retain the color coding that describes the relationship between different Requirements.

Here is a video that describes how TraceCloud implements Trace Tree Reports TODO

MS Word Reports

You may have worked in ISO, TL 9000, GEM (Great Engineering Methodology) and Six Sigma, and each of these processes require you to generate documentation regarding your project and store it in a document control system for future reference. Also from a knowledge retention perspective, it's a good idea to be able to generate word documents, that can be referred by people who will own this project, long after you have moved on.

A good Requirements Management tool should let you generate a word template with the content of your choice and then give you the ability to embed Requirements at will. This system should let you

generate dynamic word reports on the fly, that will reflect the latest & greatest Requirement collateral in your generated document.

Here is a video of how TraceCloud has solved this problem TODO

Scalability, Usability, Performance, TCO, Open Formats

As you evaluate your Requirements Management tool, you want to look out for tools that are Scalable (can support large number of users and work with large projects), are Usable (Simple to understand, and add value to the users), have quick response times, lower your TCO (Total Cost of Ownership) and support Open Formats (SQL supporting database that can be queried etc...)

Scalability, Usability and Performance of TraceCloud Video TODO

About Us

TraceCloud is a start up based in Silicon Valley (San Jose, California) . Our founders graduated from IIT, and UC Berkeley before working at Valley stalwarts like Cisco and Apple. At Cisco and Apple, they were involved with developing applications and managing systems for Engineering, Manufacturing and IT organization. They took away a lot of lessons from the way Requirements were managed in these organizations and decided to build a better mouse trap for Requirements Management.

The market leaders in this space (Requisite Pro and Doors) are based on a dated architecture. When they were initially designed, the concept of Web 2.0 , AJAX and Collaboration, did not exist. So, we decided to approach the problem with a clean slate and TraceCloud is the end result of this effort.

We are designed for both a SAAS (Software As A Service) implementation and for an On Premises implementation.

If this white paper has piqued your interest and you would like us to come over and discuss your Requirements Management implementation, please reach out to us at sami@tracecloud.com