

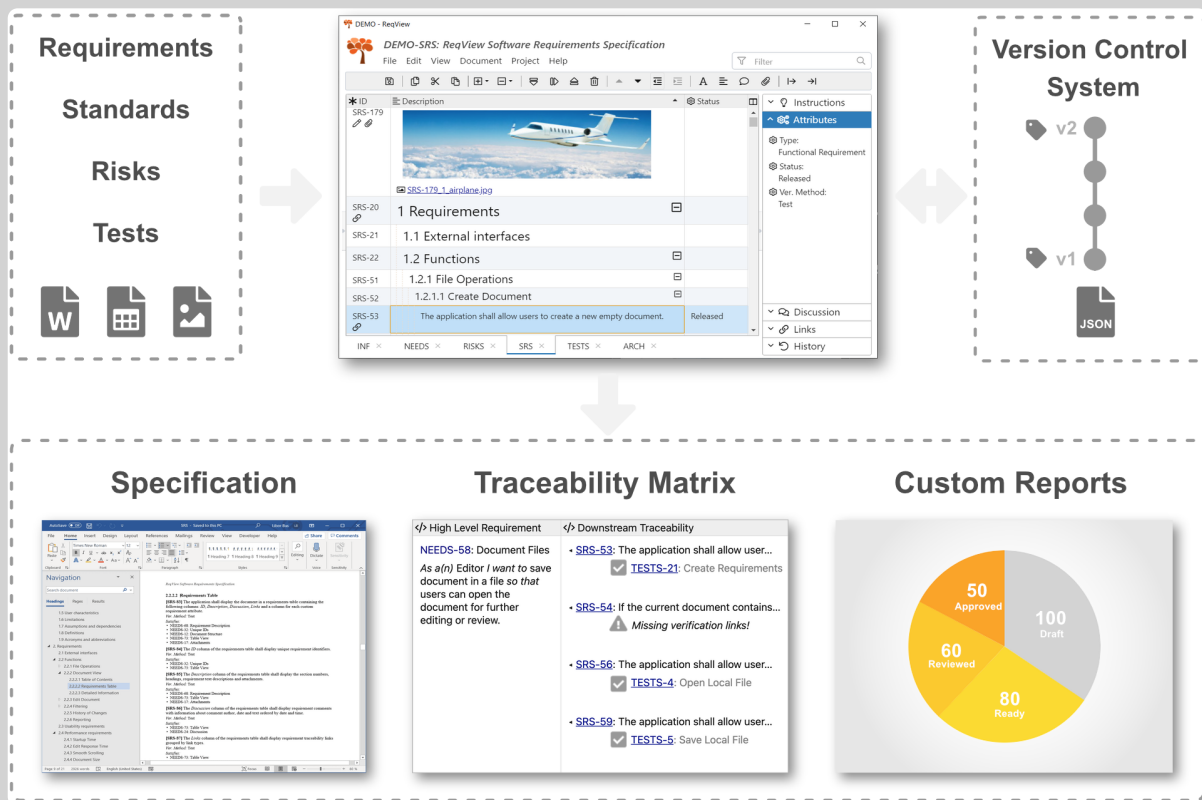
How To Progress from Traceability Matrix to Automatic Reports?

ReqView Team
Jan 28, 2021

Requirements Management Tool for SW and HW Systems

With ReqView you can get started very quickly. Just import your documents then elaborate requirements, risks, and tests. As you progress, keep track of project changes in a version control system. Finally, export specifications, or traceability reports, and share them with your team.

It's all easy with ReqView!



 [Download Reqview](#)

Table of contents

- [Why Is Requirements Traceability Important?](#)
- [What Is Wrong With Traceability Matrix?](#)
- [Manage Requirements Traceability in ReqView](#)
- [Link Requirements](#)
- [Analyze Requirements Traceability](#)
- [Customize Requirements Traceability Reports](#)
- [Try ReqView](#)
- [Learn More](#)

Why Is Requirements Traceability Important?

Requirements traceability represents a relationship between requirements and other project artifacts maintained through a software or system development process to **increase quality** of the product.

Detailed and consistent requirements traceability is essential for a demonstration that requirements were fully implemented and verified (*requirements coverage analysis*). It is also useful for estimation of related costs and risks by identification of consequences of each requirement change (*change impact analysis*).

For safety-critical systems, management of requirements traceability is the key for **compliance with functional safety standards**, such as:


- **General:** *IEC 61508* — Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems
- **Automotive:** *ISO 26262* — Road Vehicles – Functional Safety
- **Medical Devices:** *IEC 62304* — Medical Device Software — Software Life Cycle Processes
- **Aerospace:** *DO 178C* — Software Considerations in Airborne Systems and Equipment Certification
- **Railway:** *EN 50128* — Railway Applications — Communication, Signaling and Processing Systems — Software for Railway Control and Protection Systems

What Is Wrong With Traceability Matrix?

A *traceability matrix* is a table displaying relations between high-level stakeholder requirements, detailed software or hardware requirements, system design, and verification & validation (test) records.

For instance, a *requirements traceability matrix* (RTM) can represent detailed requirements as table rows and test cases as table columns. If a test case verifies a requirement then the table contains a cross character ("X") in the corresponding row, see the image below.

	A	B	C	D	E	F	G	H	I	J	K	L
1		Test01	Test02	Test03	Test04	Test05	Test06	Test07	Test08	Test09	Test10	Test11
2	SRS01	x										
3	SRS02		x									
4	SRS03			x	x							
5	SRS04				x		x					
6	SRS05				x							
7	SRS06		x									
8	SRS07											
9	SRS08			x				x				
10	SRS09			x					x			
11	SRS10	x										
12	SRS11									x		
13	SRS12									x		
14	SRS13									x		
15	SRS14									x		x
16	SRS15				x							
17	SRS16				x							
18	SRS17					x					x	x
19	SRS18						x				x	x

</> High Level Requirement	</> Downstream Traceability
<p>NEEDS-58: Document Files</p> <p>As a(n) Editor I want to save document in a file so that users can open the document for further editing or review.</p> 	<ul style="list-style-type: none"> • SRS-53: The application shall allow user... ✓ TESTS-21: Create Requirements • SRS-54: If the current document contains... ⚠ Missing verification links! • SRS-56: The application shall allow user... ✓ TESTS-4: Open Local File • SRS-59: The application shall allow user... ✓ TESTS-5: Save Local File

You can manage a traceability matrix in MS Excel, however, you will reach limits of this approach for a larger project quickly:

- Maintenance of a traceability matrix is very time-consuming and error prone — you need to manually update its rows and columns according to a list of requirements and tests described in two different documents.
- Reviewing a large traceability matrix is inconvenient — just imagine yourselves scrolling a traceability matrix with more than 100 rows and columns to find out if a requirement is verified by any test.
- Analysis of traceability analysis across multiple levels is difficult — you need to jump between traceability matrices when tracing coverage of high-level requirements (user needs) via detailed software requirements down to design elements.

Manage Requirements Traceability in ReqView

You really do not need to waste your time and efforts with manual maintenance of traceability matrices! ReqView requirements management solution will help you to manage and review end-to-end requirements traceability. See for yourself how easy it is.

Link Requirements

Linking requirements is a breeze in ReqView — just select source and target requirements and choose a link type. When you browse a requirements document you will see all needed information about linked downstream / upstream requirements in additional columns displayed next to requirements descriptions. ReqView keeps **traceability links consistent** in both directions automatically. [More info.](#)



Analyze Requirements Traceability

We know that each customer has unique needs so you can flexibly customize layout and styling how ReqView displays information about linked objects using **custom templates**. For instance, you can visualize downstream **requirements coverage** in a tree. Or, you can display additional information of the linked objects, such as status of linked tests.

When you need to export information about requirements traceability from ReqView you can choose to generate a [custom traceability report](#), or export the current table view with [custom traceability columns](#) to MS Word or Excel.



Customize Requirements Traceability Reports

Let's demonstrate how you can generate end-to-end requirements coverage for the [ReqView Demo Project](#). Our goal is to output an HTML traceability report displaying high-level user needs (*NEEDS*) in the left column and the related detailed software requirements (*SRS*) verified by test cases (*TESTS*) in the right column:

ID	Description	Traceability
NEEDS-30	Tool supports: <ul style="list-style-type: none"> displaying linked objects in the context of the related document object navigation to a chosen linked object 	◀ SRS-418 : Links Panel Implemented
NEEDS-29	Tool supports: <ul style="list-style-type: none"> printing a document with the same layout as it is displayed, saving the document as PDF with the same layout as it is displayed. 	◀ SRS-132 : User shall be able to print the current document. Implemented ◀ SRS-421 : User shall be able to export the printout to PDF. Implemented
NEEDS-24	Tool supports: <ul style="list-style-type: none"> commenting document objects with rich text comment description, discussion per a document object. 	◀ SRS-58 : Comments Implemented ◀ TESTS-22 : Comment Requirement Passed ◀ SRS-60 : Discussion Panel Implemented ◀ TESTS-22 : Comment Requirement Passed

Note: The report displays status of software requirements (*Implemented*) and test cases (*Passed*) to understand the complete development status of high-level user needs.

As the first step, we create a custom export template exporting an HTML table with two columns. ReqView custom export templates use Handlebars `{{ moustache }}` syntax and predefined helpers functions to generate a structured text output (e.g. HTML, CSV, XML, JSON) from the project data model conveniently. For more information see [Export Custom Templates](#).

Example: The following HTML template snippet renders the *Traceability* column of the HTML table without any styling:

```
<ul>
  {{#eachInlinkWith type="satisfaction"}}
  <li>
    <span>{{docId}}-{{id}}: {{shortDescription}}</span>
    <span>{{status}}</span>
    <ul>
      {{#eachInlinkWith type="verification"}}
      <li>
        <span>{{docId}}-{{id}}: {{shortDescription}}</span>
        <span>{{status}}</span>
      </li>
    </ul>
  </li>
  {{/eachInlinkWith}}
</ul>
```

The snippet first uses `eachInlinkWith` block helper to iterate all incoming satisfaction traceability links of the given *NEEDS* user story. For each such link, it renders a *SRS* requirement ID, a short description (*heading* or *text* attributes) and the value of the *status* custom attribute.

Still in the context of the linked *SRS* requirement, the snippet analogically uses `eachInlinkWith` helper to iterate all incoming verification links. For each *TEST* object it renders a test ID, a short description, and the value of the *status* custom attribute.

You can download the full custom export template for our example traceability report from the [ReqView Support Site](#) (log in needed). If you need a little help with your specific traceability reports then [contact us](#). We offer [custom services](#) and will be happy to tailor complex custom reports according to your specific needs.

Try ReqView

Find out how ReqView can simplify your daily requirements traceability tasks. Just [download](#) ReqView and [request PRO trial license](#) to start your free trial.



Learn More

- [Blog: Manage Requirements Better than in Excel](#)
- [Documentation: Traceability Links](#) — learn how to manage requirements traceability
- [Documentation: Export Custom Templates](#) — learn how to customize traceability reports
- [Support Portal: Downloads](#) — download example traceability templates