



SafeSuite Web: User Manual



V 0.2.0, OCT. 2026

CYNOS

Contents

1	Introduction	6
1.1	SafeSuite Web and the SafeSuite desktop application	6
1.2	Who this manual is for	6
1.3	Conventions used in this manual	6
1.4	What you need.....	7
1.5	How this manual is organised	7
2	Getting started.....	9
2.1	Open SafeSuite Web	9
2.2	Try the demo	9
2.3	Sign in to your licence	10
2.4	Connect your first project	10
2.5	Open a document and make a change.....	11
2.6	Share your change.....	12
2.7	Get the team's updates.....	13
2.8	Where to go from here	13
3	How SafeSuite Web keeps your work.....	14
3.1	Projects, documents and the repository.....	14
3.2	Your working copy, in this browser	14
3.3	Save, Share, Get updates: the three steps	14
3.4	Versions and history.....	15
3.5	Working alongside others	15
3.6	Safety nets.....	16
3.7	Working offline, or signed out.....	16
3.8	Where everything is kept	17
3.9	Real situations, and how they are handled.....	17
3.10	Privacy and the flow of data	18
4	Licensing.....	19
4.1	How SafeSuite Web licensing works	19
4.2	Signing in	19

4.3	The licence status light.....	20
4.4	What happens without a licence.....	21
4.5	The licence states and their notes	21
4.6	Using your licence on another computer.....	22
4.7	Offline grace	22
4.8	Signing out of the licence	23
4.9	The demo	23
4.9.1	Personal demo links	24
4.10	Getting a licence and support.....	24
5	Connecting a repository.....	25
5.1	Host	25
5.2	Sign in	26
5.3	Project	26
5.4	Branch	27
5.5	Download	27
5.6	Check out.....	27
5.7	The start page: projects in this browser.....	28
5.8	For your administrator	28
6	The workspace	29
6.1	The parts of the screen	29
6.2	The top bar	30
6.3	The account menu.....	31
6.4	The tool rail	32
6.5	The side panel	34
6.6	Document tabs and the file header.....	34
6.6.1	Side by side.....	35
6.7	The status bar.....	35
6.8	Messages.....	37
6.9	Themes, layout, About and new versions	37
6.10	On a phone	38
7	The project explorer	41

7.1	The tree	41
7.1.1	Keys in the explorer.....	41
7.1.2	Drag and drop	41
7.2	Project actions.....	42
7.3	Folder and file menus.....	44
7.4	Creating folders, documents and predictions	45
7.5	Uploading files.....	46
7.6	Renaming, moving, duplicating and deleting	46
7.7	Hidden and excluded items.....	46
7.8	Find files and folders	46
7.9	Templates in this browser	46
7.10	Partial checkouts.....	46
8	Structured documents	48
8.1	What a structured document is	48
8.2	Opening a document and the toolbar.....	48
8.3	Items: adding, moving, deleting.....	50
8.4	Editing cells.....	51
8.4.1	The rich-text Format dialog.....	52
8.5	Selecting, copying and pasting rows	52
8.6	Links between items.....	53
8.7	Views and columns.....	53
8.8	Find and filter	55
8.9	Document configuration	55
8.10	Level styles.....	57
8.11	Problems inside a document	57
8.12	Item history.....	57
8.13	Saving and leaving	57
8.14	Check & Repair.....	58
8.15	Jira columns	59
8.16	Structured documents on a phone.....	59
9	Analyses: FMEA, FIT predictions and fault trees.....	60

9.1	The FMEA editor	60
9.2	The FIT prediction editor	61
9.3	The fault tree editor	62
9.4	Reservations for these files	63
9.5	What the editors do not do	63
10	Other files	64
11	Versions, history and baselines	67
11.1	Changes: what you have not shared	67
11.2	History: the versions of the branch	68
11.3	File history	69
11.4	Item history	69
11.5	Baselines	70
11.6	Compare	70
11.7	Change logs	71
12	Working with the team	72
12.1	Share	72
12.2	Get updates	73
12.3	How merging works	73
12.4	Differences to resolve	73
12.5	Decisions left for later	74
12.6	Files held back	74
12.7	Who is editing what	74
12.8	Backups in this browser, Revert and restore	75
12.9	The backup branch	76
12.10	Offline or signed out	76
12.11	Check server and updates waiting	76
13	Traceability and problems	77
13.1	Traceability	77
13.2	Problems	78
14	Import, export and PDF	80
14.1	Importing a ReqIF file	80

14.2	Importing Excel or CSV.....	80
14.3	Importing an SBOM	81
14.4	Imports into the open file.....	81
14.5	The Export dialog	82
14.6	PDF.....	83
14.7	Printing.....	84
14.8	Files to and from your computer	84
15	Governance	85
15.1	Switching governance on.....	85
15.2	Identity and signing key	85
15.3	The lifecycle of a document.....	86
15.4	Reviews	87
15.5	Sealed baselines.....	87
15.6	The Governance dialog	87
15.7	Sharing in a governed project.....	88
15.8	A governed project with governance off	89
15.9	Copy without governance.....	89
15.10	What the desktop application adds.....	89
16	Settings, shortcuts and troubleshooting	90
16.1	Settings	90
16.2	Keyboard shortcuts.....	91
16.3	Messages and new versions	93
16.4	Troubleshooting.....	93
17	Limits, and what the web application does not do	95
17.1	Left to the desktop application.....	95
17.2	Browser and device limits.....	95
17.3	Other known limits	96
18	Glossary	97
	Appendix A: Technical requirements (informative).....	99
	Appendix B: Support	99

1 Introduction

Welcome to SafeSuite Web, the browser edition of SafeSuite, the requirements and safety engineering platform by Cynoss. This manual describes version 0.2.0. It explains every feature of the web application, how to reach it, and what happens behind the scenes when you save, share and work with colleagues.

1.1 SafeSuite Web and the SafeSuite desktop application

SafeSuite exists in two forms that work on the same project files and the same repositories:

- **The SafeSuite desktop application** for Windows is the complete suite: structured documents, FMEA, FIT reliability predictions, fault trees, safety and cybersecurity assessments, the AI assistant, Jira integration, licensing administration and the full governance toolset.
- **SafeSuite Web**, described here, runs in a browser, with nothing to install. It gives you a quick, direct way to the everyday work: open a project, read and edit its documents and analyses, trace requirements, compare versions, share your changes with the team and get theirs. It is the right tool for the engineer who needs to update a requirement from a meeting room, the reviewer who wants to read an analysis on a tablet, or the team member who does not have the desktop application installed.

Both applications read and write the same files, in the same repository, and they can be used side by side on the same project. Whenever a task is best done in the desktop application, this manual says so. Chapter 17 lists what the web application leaves to the desktop.

Note: SafeSuite Web never stores your project data on Cynoss servers. Project files travel only between your browser and your organisation's own repository.

1.2 Who this manual is for

This manual assumes no prior knowledge of Git or of version control. The words *repository*, *version*, *share* and *get updates* are explained in chapter 3 in plain terms, and a glossary at the end of the manual gathers them. If you have used the desktop application, you will recognise the documents and most of the vocabulary; if you have not, chapter 2 gets you from the first page to your first shared change in a few minutes.

1.3 Conventions used in this manual

- Names of buttons, menus, fields and dialogs are shown in **bold**, exactly as the application spells them: press **Share**, open **Project actions**.
- A sequence of choices is written with arrows between the steps: **Import** ► **Excel or CSV...**
- Keyboard shortcuts are written as **Ctrl+S**. On a Mac, **⌘** works wherever Ctrl is shown.
- Messages the application displays are quoted in "double quotes".
- Boxes like the one above hold notes, tips and warnings.
- Step-by-step procedures are numbered. Follow them in order.

- Screenshots show the application in its light theme. Your copy may be in the dark theme, chosen from the account menu; the layout is the same.

1.4 What you need

You need	Details
A modern browser	Chrome or Edge give the full experience, including repositories in a folder of your computer. Firefox and Safari work with hosted repositories. On a phone, the application adapts its layout.
The address of your SafeSuite Web site	Your organisation's installation, for example https://safesuite.cynos.net . Ask your administrator.
A repository account	A GitLab or GitHub account, an account on your organisation's Git server, or simply a shared folder. Chapter 5 explains each.
A SafeSuite licence	Where the site is licensed, you sign in with your SafeSuite account to edit structured documents, exchange ReqIF files and act in governance. Reading, tracing, comparing and exporting are free. Chapter 4 explains licensing. Without an account, the demo lets you try everything.

1.5 How this manual is organised

Chapter	What it covers
2 Getting started	From the first page to your first shared change, in a few minutes
3 How SafeSuite Web keeps your work	The concepts: repositories, working copies, saving, sharing, getting updates, merging, reservations, backups. Read this once
4 Licensing	Signing in, what each licence unlocks, the demo
5 Connecting a repository	The Connect repository wizard for every kind of host
6 The workspace	The screen, panel by panel
7 The project explorer	Files and folders
8 Structured documents	The requirements editor, in full
9 Analyses	FMEA, FIT predictions and fault trees

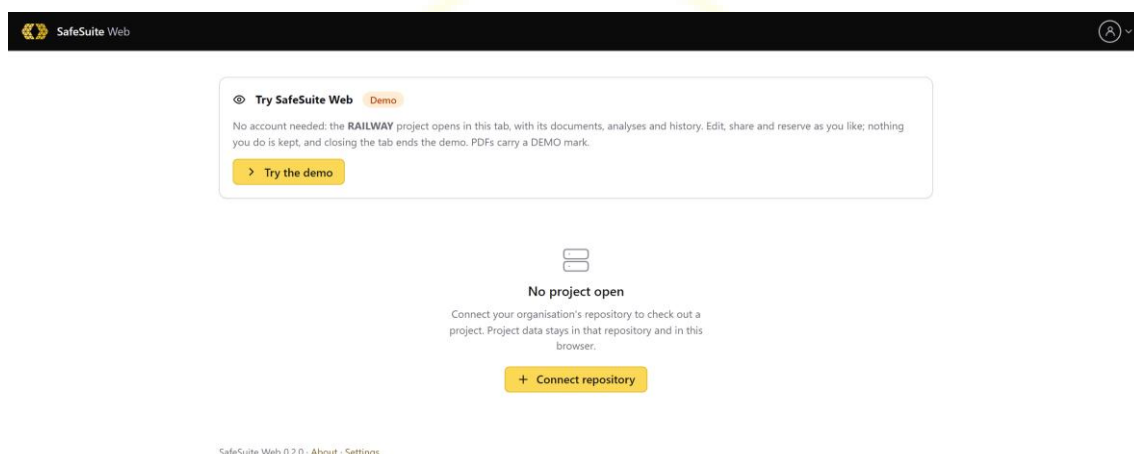
Chapter	What it covers
10 Other files	Text, Markdown, images, PDFs, SBOMs
11 Versions and history	History, file history, baselines, compare
12 Working with the team	Share, Get updates, conflicts, reservations, backups, offline
13 Traceability and problems	Coverage, gaps, broken links, review markers
14 Import, export and PDF	ReqIF, Excel, CSV, PDF reports, printing
15 Governance	Reviews, approvals, sealed baselines, policies (an optional feature)
16 Settings and troubleshooting	Settings, keyboard shortcuts, messages, recovering from problems
17 Limits and what the web application does not do	Known limits and the tasks that belong to the desktop application
18 Glossary	The words used in this manual

2 Getting started

This chapter takes you through a first session. Every step is described in more detail later in the manual.

2.1 Open SafeSuite Web

Type the address of your SafeSuite Web site in the browser. The first page is the start page: it lists the projects already in this browser (none, the first time), offers **Connect repository**, and, where the site offers one, the demo.



The start page, before any project is connected

The application loads once and then keeps an offline copy of itself in the browser, so later starts are fast and work without a network. You can also install it as an application from the account menu at the top right (**Install app**, when the browser offers it): it then opens in its own window and the browser keeps its local data even when the disk fills up.

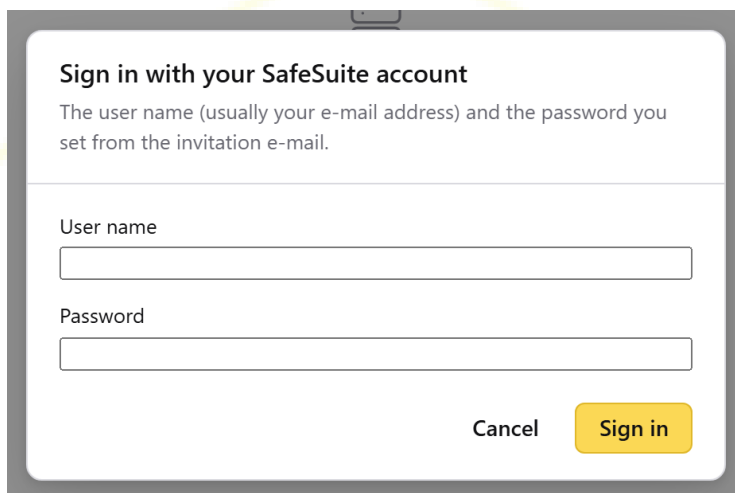
2.2 Try the demo

If you have no project yet, or want to look around first, press **Try the demo**. A demo project opens in this tab with its documents, analyses and history. You can edit, share and reserve as you like: nothing you do is kept, closing the tab ends the demo, and PDFs made in the demo carry a DEMO mark.

On the public site the demo starts from a personal link, which you receive by e-mail after asking for it on the Cynos web site. The link opens the demo directly; chapter 4 explains the rules attached to it.

2.3 Sign in to your licence

On a licensed site, a note at the bottom right says "Sign in with your SafeSuite account to use your licence." Press **Sign in**, enter the user name (usually your e-mail address) and the password you set from your invitation e-mail, and press **Sign in**. The licence stays with you for the working day. Without it you can read everything and export, but structured documents open read-only.

A sign-in dialog box with a white background and a grey border. At the top, it says "Sign in with your SafeSuite account" in bold. Below that, a smaller text says "The user name (usually your e-mail address) and the password you set from the invitation e-mail." There are two input fields: "User name" and "Password". At the bottom right, there are two buttons: "Cancel" and "Sign in" (which is highlighted in yellow).

The sign-in dialog of the licence

2.4 Connect your first project

1. Press **Connect repository**.
2. **Host**: choose where the project lives (GitLab.com, GitHub.com, your organisation's Git server, or a folder on this computer) and press **Continue**.
3. **Sign in**: GitLab sends you to its own page to approve the access; GitHub asks for an access token; a Git server asks for your user name and a token; a folder only asks for your name and e-mail.
4. **Project**: search your projects or paste the project's address, and pick the project.
5. **Branch**: pick the branch to work on, normally the one marked **default**.
6. **Download**: keep **Everything** (recommended), or choose the folders you need.
7. Press **Check out**. The files are downloaded into the browser and the project opens.

Connect repository

SafeSuite reads and writes project files only in your organisation's repository. Nothing is stored on SafeSuite servers.

Host › Sign in › Project › Branch › Download › Check out


GitLab.com
 gitlab.com


GitHub.com
 github.com


A Git repository in a folder
 On this computer or a network share

Only the hosts your administrator allowed are listed. They are the only servers that receive project data.

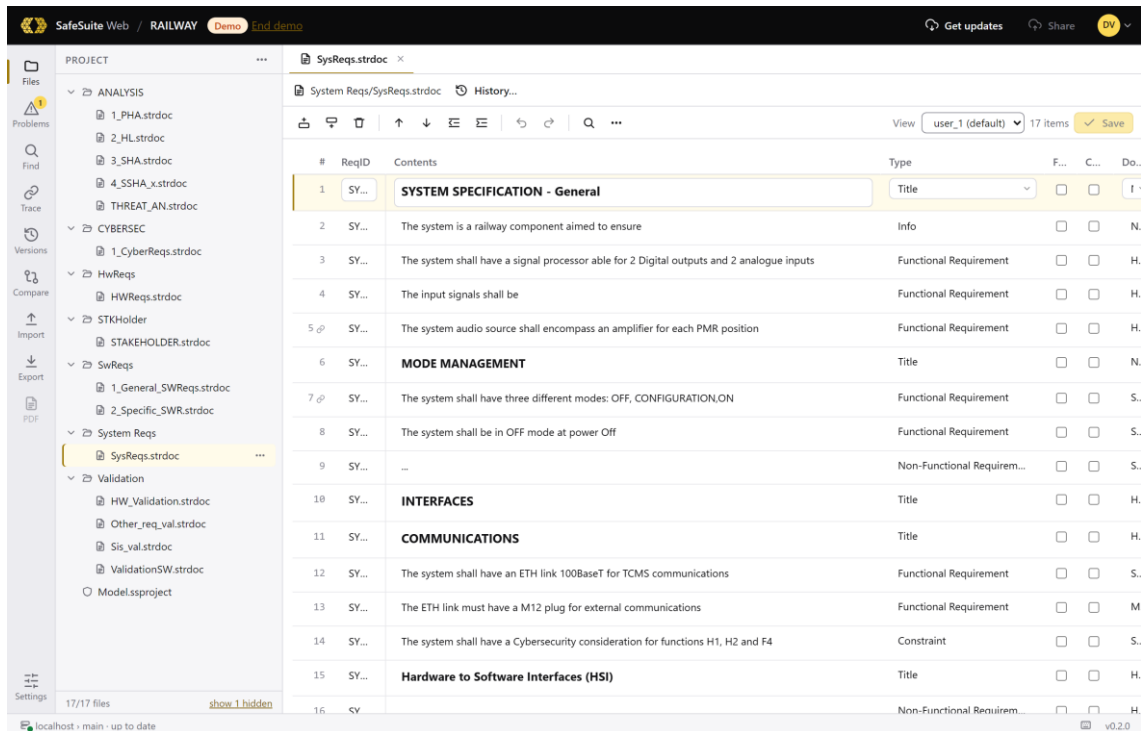
Continue

The Connect repository wizard

The project is now a *working copy in this browser*: it opens again from the start page, with or without a network connection.

2.5 Open a document and make a change

The project explorer on the left lists the folders and files. Click a structured document (a `.strdoc` file) to open it in a tab. It is a table: one row per requirement or item, one column per attribute.



The screenshot shows the SafeSuite Web interface for a project named 'RAILWAY'. The left sidebar contains a file explorer with a tree view of the project structure. The main workspace displays a structured document titled 'SysReqs.strdoc'. The document is organized into sections: 'SYSTEM SPECIFICATION - General', 'MODE MANAGEMENT', 'INTERFACES', 'COMMUNICATIONS', and 'Hardware to Software Interfaces (HSI)'. Each section contains a list of requirements with columns for ID, content, type, and status. The 'SYSTEM SPECIFICATION - General' section includes requirements 1 through 5, and 'MODE MANAGEMENT' includes requirements 6 through 9. The 'INTERFACES' section includes requirements 10 through 11, 'COMMUNICATIONS' includes requirements 12 through 14, and 'Hardware to Software Interfaces (HSI)' includes requirement 15. The bottom status bar shows 'localhost - main - up to date' and 'v0.2.0'.

A structured document open in the workspace

1. Click a cell and type. Text cells edit in place; rich-text cells show a small formatting toolbar.
2. To add a requirement, select a row and press **Below** in the toolbar (or the **Insert** key).
3. Press **Ctrl+S** or the toolbar's **Save**. The message "Saved ... in this browser. Share it to make it available to others." confirms the save.

Saving keeps the change in your browser only. The explorer marks the file **edited**, the file's header shows an **unshared** chip, and the **Share** button in the top bar turns gold with the number of changed files.

2.6 Share your change

1. Press **Share** in the top bar.
2. Check the list of files, type a short **Summary** of what you changed, and press **Share**.

ReqID

Contents

Share changes

Your changes become one version on main in localhost. Changes others shared meanwhile are taken in first.

1 of 1 file(s) to share

☒ **modified** System Reqs/SysReqs.strdoc

[all](#) [none](#)
[compare](#)

Summary (required)

Edited SysReqs.strdoc

Kind of change

Maintenance ▼

Reference (optional)

ticket, ECO or requirement id

Motivation (optional)

☒ Release my locks on the shared files afterwards

Cancel

Share

The Share dialog

Your files become one new version in the repository, visible to the whole team. If colleagues shared changes meanwhile, those are taken in first and merged with yours; where you both changed the same thing, the application asks you what to keep. The message "Shared N file(s) as version ..." closes the loop.

2.7 Get the team's updates

When others share, a dot appears on **Get updates**. Press it to bring their versions into your working copy. Open documents reload by themselves.

2.8 Where to go from here

- Chapter 3 explains what just happened: the working copy, the saved and unshared states, the repository and the merge.
- Chapter 6 tours the screen; chapters 7 to 14 describe each tool.
- The cheatsheet, delivered with this manual, condenses the everyday actions and shortcuts onto a few pages.

3 How SafeSuite Web keeps your work

This chapter explains the ideas behind the application. Nothing in it requires Git knowledge, but after reading it you will understand what Git does for you underneath.

3.1 Projects, documents and the repository

A SafeSuite **project** is a folder of files: structured documents (.strdoc), FMEAs (.sfmea), FIT predictions (.sfit), fault trees (.fta), plus notes, pictures, PDFs and anything else the team keeps with it. A small project file (.ssproject) marks the folder as a project.

The project lives in a **repository**. A repository is a place, on a server or on a shared drive, that keeps not only the current files but every **version** they ever had: who changed what, when, and why. Git is the technology behind it, and it is the same technology software teams use. GitLab.com, GitHub.com and your organisation's own Git server are services that host repositories; a bare folder on a network drive can be one too.

A repository can hold several **branches**: parallel lines of versions. Most teams work on the default branch, often called main. SafeSuite Web asks you which branch to work on when you connect, and shows it in the status bar.

Note: Your organisation's repository is the only place project data goes. SafeSuite Web talks to it directly from your browser; Cynoss servers never see your files.

3.2 Your working copy, in this browser

When you connect a project, SafeSuite Web **checks it out**: it downloads the files of the chosen branch into the browser's private storage. That is your **working copy**. It belongs to this browser on this computer: another browser, or another computer, has its own working copy.

The working copy records the **base version**, the repository version it was taken from or last updated to, and keeps your changes next to it. From then on the project opens from the browser, instantly, with or without a network.

The start page, "Projects in this browser", lists the working copies, most recently opened first, each with its branch and host. A trash button removes a copy from the browser; the repository is not touched.

3.3 Save, Share, Get updates: the three steps

Three actions move your work around. Keeping them apart is the key to understanding SafeSuite Web.

Action	What it does	Where the result is
Save (Ctrl+S)	Writes the open document into your working copy	This browser only. Nobody else sees it
Share	Sends your saved changes to the repository as one new version, with a summary	The repository: the whole team sees it
Get updates	Brings the versions others shared into your working copy	Your working copy, merged with your own changes

A saved change that has not been shared is an **unshared change**. The application shows unshared changes everywhere: the explorer marks files **new**, **edited** or **deleted**; the file's header shows an **unshared** chip with **Compare** and **Revert...**; the **Share** button turns gold with a count; and the Versions pane's **Changes** tab lists them all.

Tip: Save often and share at natural milestones, with a summary that tells the team what changed. Each share becomes one line of the project's history, as the desktop application writes it.

3.4 Versions and history

Every share creates a new version of the branch: a snapshot of every file, with the author, the date, the summary and the kind of change (Feature, Fix, Docs and so on). The Versions pane (chapter 11) lists this history for the project, for a folder, for a file and even for a single row of a structured document. Any version can be compared with another or restored.

Versions are identified by a short code such as a3f9c21. The status bar shows the version your working copy is based on, and PDFs record it on their cover.

A **baseline** gives a version a name, for example "v1.0 — Type Approval". In the web application a baseline is a Git tag; with governance (chapter 15) it is also sealed and can be signed.

3.5 Working alongside others

Several people edit the same project at the same time. SafeSuite Web handles this in two ways, depending on the kind of file.

Files that can be merged. Structured documents are merged item by item and field by field; text files are merged line by line. When you share, the changes others shared since your base version are taken in first and combined with yours. Where the two sides changed the same field or the same lines, the application does not guess: it opens a decision list where you keep yours, take theirs, or edit the result (chapter 12). Nothing is lost silently, and the version you had is backed up in the browser before anything replaces it.

Files that cannot be merged. FMEAs, FIT predictions and fault trees are structured in ways no automatic merge can combine safely. Your first edit of one of them **reserves** it (a lock) in the repository for you; everyone else sees it read-only, with your name and the time, until you share or release it. Reservations last 14 days and renew while you work. Someone who must take over can break a lock, giving a reason that is recorded. The desktop application honours the same reservations.

Presence. While you edit any file, the team sees a small chip "... is editing since 10:42" on it, in the web and in the desktop application. Presence is informative; it does not block anybody.

3.6 Safety nets

SafeSuite Web keeps copies of your work at several levels, so that a merge, a mistake or a browser cleanup does not cost you anything.

Safety net	What it protects	Where
Backups in this browser	Every version of yours that a merge, Revert, Keep theirs, restore, repair, delete or upload replaced	Inside the working copy. Project actions ▶ Backups in this browser...
The backup branch	Your unshared changes, copied to the repository in the background after each change while you are signed in	A branch of your own in the repository, safesuite-wip/<you>/<branch>. Restored when you check the project out again
Undo and Redo	Explorer operations: rename, move, delete, paste, new items, uploads	Ctrl+Z and Ctrl+Y in the explorer; also in the editors
History	Every shared version of every file	The Versions pane: open, compare and restore any version

The status bar's safety item tells you the state of the backup branch: "backed up 10:42", "backup failed", "backup paused", "not protected" or "in this browser".

3.7 Working offline, or signed out

The working copy does not need the network. You can open the project, edit, save, compare, use the explorer and the backups while offline. What waits for the connection: sharing, getting updates, the backup branch, presence and reservations, and the licence beyond its grace period. The status bar then reads "not reachable", and the actions that need the repository say so when you try them.

3.8 Where everything is kept

What	Where	Survives a reload	Survives a browser cleanup
Working copies, unshared changes, backups in this browser	The browser's private storage for this site	Yes	No: browsers may delete site storage when the disk is nearly full. Installing the app makes this unlikely
Settings, layout, open tabs, expanded folders	The same storage	Yes	No
Sign-ins to repositories (tokens)	The tab, and with Stay signed in also the browser, encrypted	Yes, with Stay signed in	No
Unsaved edits in an open editor	The page's memory	No: the browser asks "Leave site?" first	No
The demo	The page's memory	No	No
The repository	Your organisation's server or shared folder	Yes	Yes

If the browser does delete the site's storage, the start page reports "The browser removed your local copy of ...", says whether unshared changes were lost, and offers **Check out again**. If the backup branch had copied your unshared changes, the new checkout offers to restore them.

Settings ▶ Troubleshooting ▶ Reset this browser... removes everything SafeSuite Web keeps in the browser, as if it had never been opened there. Shared work in the repositories is never touched.

3.9 Real situations, and how they are handled

Situation	What happens
Two people edit the same structured document	No lock. Each sees the other's "is editing" chip. The first to share wins as a plain update; the second's share takes it in and merges item by item. Only fields both changed are asked about, one by one
Two people edit the same FMEA, fault tree or FIT prediction	The first edit reserves the file. The other sees it read-only with the reason and Break lock.... If both did edit anyway, the share offers Keep both, Keep mine or Keep theirs
A colleague deletes a file you edited	At share: "You changed this file; X deleted it." You choose Keep both, Keep mine or Accept their deletion

Situation	What happens
A colleague shares while you have unsaved edits	Your editor keeps your edits and warns that the file changed in the repository; compare with the shared version before saving over it
You worked offline for a day	Save as usual. Once online, share: others' changes are taken in first, merged, and overlaps are asked about
The browser lost the site's storage	The start page says so. Check the project out again and restore the backup of your unshared changes
You rename or move files	A local change, shared as a delete plus an add. Open tabs, reservations and records follow the file. Renaming a file someone else reserved is refused
A repository holds several SafeSuite projects	Allowed side by side, never nested. The explorer offers New project... on a suitable folder. Problems and traceability work on the top-most project
A large project	Checkout downloads six files at a time with a progress count. Choose folders at checkout downloads only part of it; the rest is listed and downloaded on demand
You use two computers	Each has its own working copy. Share from one, get updates on the other. The backup branch follows you: a new checkout offers to restore your unshared changes

3.10 Privacy and the flow of data

- Project files move only between your browser and the repository your administrator allowed. The status bar's **Team server** panel states it: "Project data: Goes to this repository only. AI assistance is off."
- Sign-ins to repositories are kept in the browser, encrypted when **Stay signed in** is on; SafeSuite never sees your GitLab password, since GitLab's own page handles the sign-in.
- The licence gateway on the site talks to the licence server for the browser; it learns your licence state, never your project data.
- The demo runs in the page's memory and is forgotten when the tab closes.

4 Licensing

This chapter describes licensing from your side: what a licence is, how you sign in, what each licence unlocks, and what the application shows when something is not available. Where your SafeSuite Web site is not licensed (an internal installation, for example), nothing in this chapter appears in the application and every feature is open.

4.1 How SafeSuite Web licensing works

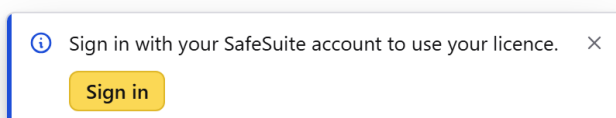
- **Your SafeSuite account.** Cynoss support creates your account and sends you an invitation e-mail. From it you set your password. The user name is usually your e-mail address.
- **Three licensed features.** A licence names the features you hold. Each can be held on its own:
 - **Structured Document Editor:** editing structured documents (.strdoc).
 - **ReqIF Import and Export:** exchanging requirements with other tools.
 - **Governance:** reviewing, approving, signing and changing the governance records of a project (chapter 15).
- **Everything else is free.** Opening and reading, searching, traceability, compare and history, PDF and Excel export, FMEAs, FIT predictions, fault trees, sharing and getting updates need no licence.
- **One computer at a time.** Your licence serves one computer at a time; all the tabs and windows of that computer share it. Using it on a second computer moves it there (see below).
- **Renewal.** While the application is open, the licence session renews itself about every minute. A new licence, a withdrawn licence or a network ban therefore reaches the application within a minute.
- **Duration.** A sign-in lasts a working day: it ends after 8 hours without activity, and the renewal keeps it alive while you work.

The application never names the licence server; it speaks of "your SafeSuite account" and "the licence server". A small gateway on the site talks to the licence server for the browser.

4.2 Signing in

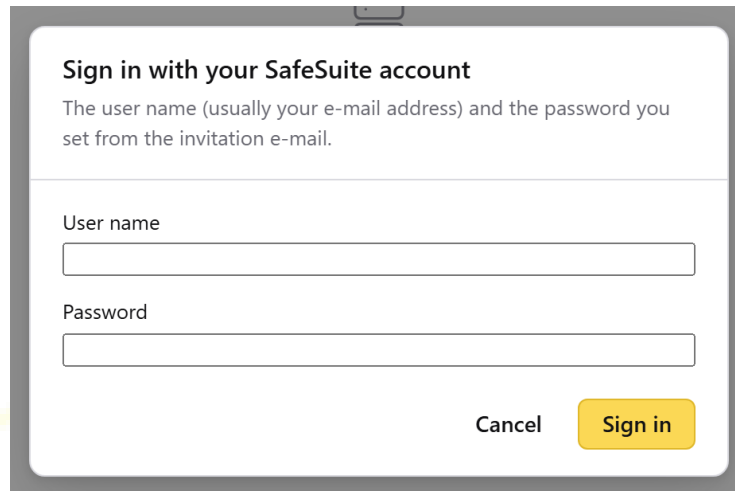
There are three ways to open the sign-in form:

- the licence note at the bottom right ("Sign in with your SafeSuite account to use your licence.") and its **Sign in** button, shown on every page including the start page;
- the licence status light in the status bar and its panel's **Sign in...** button;
- **Settings** ▶ **Licence** ▶ **Sign in....**



The licence note, at the bottom right of every page until you sign in

The dialog is titled **Sign in with your SafeSuite account**. Enter your **User name** and **Password** and press **Sign in**. The form works with password managers, remembers the last user name in this browser, and never stores the password.



Sign in with your SafeSuite account

The user name (usually your e-mail address) and the password you set from the invitation e-mail.

User name

Password

Cancel **Sign in**

Sign in with your SafeSuite account

Messages the form can show:

Message	Meaning
"The user name or password is wrong."	A wrong password, an unknown user, an account without a password yet, or a locked account
"Too many sign-in attempts. Try again in N seconds."	Wait, then try again
"Signing in is not possible from this network."	Your organisation's licence server does not accept this network
"The licence server cannot be reached. Try again later."	The licence server is down or unreachable
"The licence gateway answered with an error. Try again later."	A problem on the site; tell your administrator if it persists

4.3 The licence status light

In the workspace, the status bar shows a key with a coloured dot at the right: green when your licence is granted, amber when something needs you (sign-in required, offline grace, in use elsewhere), red when a feature is off. Next to it is your user name, "Demo" in a demo, or "Not signed in".

Click it to open the **SafeSuite licence** panel: the state in one sentence, each feature with its state, and the buttons the state needs: **Sign in...**, **Use it here**, **Try again**, **Sign out**.

Settings ▶ **Licence** shows the same information: the account line and each feature in bold with "Included.", "Off until you sign in.", "Your licence does not include ..." or "Off: ...".

Note: On a phone the status bar is hidden. Use the licence note and **More** ▶ **Settings** instead.

4.4 What happens without a licence

When a feature cannot be used, its control is disabled, or a quiet hint line above the editor says why. The hint takes one of two forms:

- "Your licence does not include Structured Document Editor." or "Your ... licence expired on 3 Oct 2026." when you are signed in but do not hold the feature;
- "Structured Document Editor needs your licence: sign in with your SafeSuite account first." (or another reason: "it is in use on another computer.", "the licence server cannot be reached.", "the licence session ended." and so on) when there is no working session.

Feature	Without it
Structured Document Editor	Documents open read-only. In the explorer, New structured document , Duplicate and Save as template... are disabled, with the reason. If the licence goes away while a document has unsaved edits, the document stays editable until those edits are saved or discarded: nothing typed is lost
ReqIF Import and Export	Import ▶ ReqIF exchange file... is greyed out, the ReqIF formats of the Export dialog are disabled, and a .reqif file opened from the explorer shows as a plain file. Downloading the file stays free
Governance	You can read every record, seal and history, verify seals and export the evidence pack. Reviewing, approving, signing and changing records is refused with the reason

4.5 The licence states and their notes

Each change of state is announced once by a note at the bottom right. Dismissed, it does not come back until the state changes again.

State	Note	Button
Sign-in required	"Sign in with your SafeSuite account to use your licence."	Sign in
Granted	no note	

State	Note	Button
Degraded	"The licence server cannot be reached. Licensed features keep working until 11:05."	
In use elsewhere	"Your licence is now in use on another computer: licensed features are off here."	Use it here
No licence	"You hold no SafeSuite Web licence: structured documents open read-only, and ReqIF and governance actions are off. Ask Cynos support for a licence."	
Withdrawn	"Your licence was withdrawn: licensed features are off."	
Network blocked	"Licences cannot be used from this network: licensed features are off."	
Session ended	"Your licence session ended."	Reconnect
Server unreachable	"The licence server cannot be reached: licensed features are off until it can."	Try again

If the licence is lost while you are working (taken over by another computer, withdrawn, banned, or unreachable for too long), a dialog titled **Your licence session ended** explains the reason once and adds: "What you were editing stays open, and its unsaved changes can still be saved. Licensed features stay off until the licence is back."

4.6 Using your licence on another computer

Start SafeSuite Web on the second computer and sign in. If the first computer still holds the licence, a dialog titled **Your licence is in use on another computer** names it, with the time it was last active. Press **Use it here** to move the licence: within a minute the other computer stops using it and keeps its open documents read-only. **Cancel** leaves the licence where it is.

If your organisation does not allow taking a licence over, the dialog says so: close SafeSuite Web on the other computer, wait a few minutes, then try again.

4.7 Offline grace

If the licence server cannot be reached but the site's gateway answers, your licence keeps working for a grace period, and the status light reads "offline grace". If the gateway itself cannot be reached, what you hold keeps working for a few minutes while the application retries every 20 seconds; then

licensed features stop with the note "The licence server cannot be reached: licensed features are off until it can." and a **Try again** button. Everything outside the three licensed features keeps working, including reading structured documents.

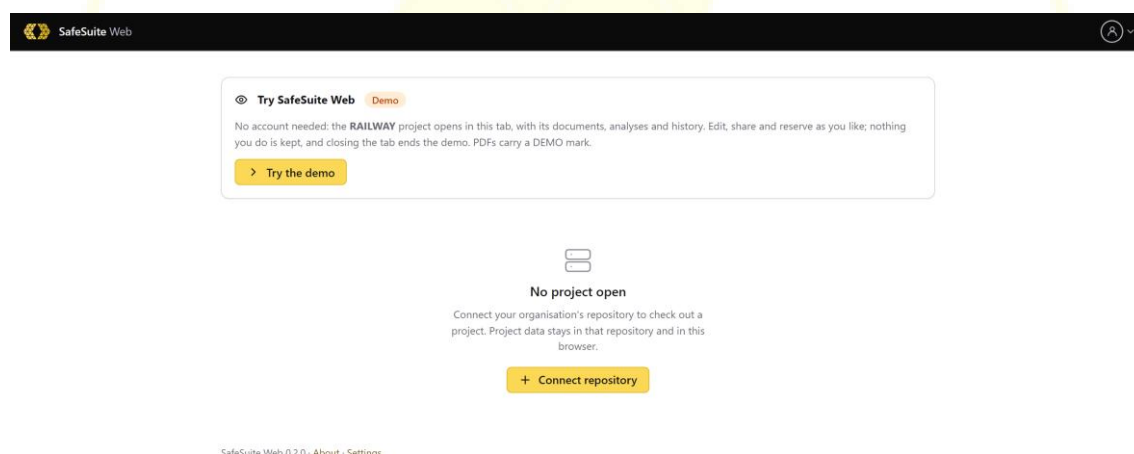
4.8 Signing out of the licence

- **Settings** ▶ **Licence** ▶ **Sign out of the licence**;
- the status light's panel ▶ **Sign out** ("Ends your licence session here, so your licence is free for another computer at once");
- **Sign out** on the Connect page also ends the licence session, as does **Reset this browser....**

Note: The **Sign out** in the account menu inside a project signs out of the repository host, not of the licence.

4.9 The demo

Where the site offers a demo, the start page shows the card **Try SafeSuite Web**. A demo lends you a seat for the visit: the demo project opens in this tab with its documents, analyses and history. You can edit, share and reserve as you like.



The demo card on the start page

What to know about the demo:

- **Nothing is kept.** The working copy and its repository live in the page's memory. Closing the tab, reloading it or pressing **End demo** drops everything. The demo never appears in "Projects in this browser".

- **The top bar** shows a **Demo** badge and **End demo**. The account menu shows "Demo visitor"; the status light reads "Demo".
- **PDFs** made during a demo carry "DEMO" diagonally across every page.
- **ReqIF** is not included in a demo seat; governance is always off in the demo.
- **Seats are limited**. When every seat is in use, the demo says "Every demo place is in use right now. Try again in a few minutes."

4.9.1 Personal demo links

On the public site the demo starts from a personal link, <https://safesuite.cynoss.net/try?t=...>, sent to you by e-mail after you ask for it on the Cynoss web site (the **Get your demo link** button takes you there). The rules:

Rule	Value
The link is valid for	7 days
It works in	the first browser that opens it
It can start the demo	3 times a day (reloading a running demo does not count)
From one network	at most 10 demo starts an hour

A link that cannot be used says so after "The demo did not start.", for example "This demo link has expired. Get a new one at ..." or "This demo link is already in use in another browser. Get your own at ...". The site keeps no e-mail address and no network address with the links.

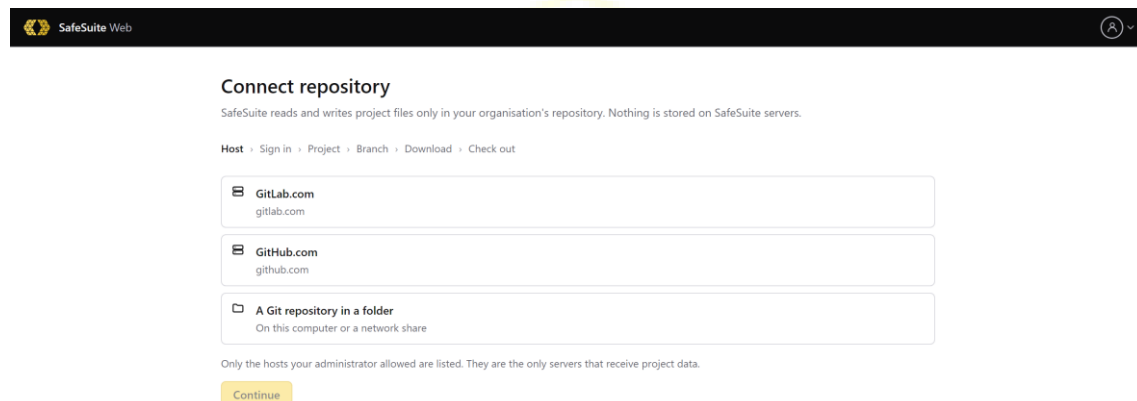
4.10 Getting a licence and support

Licences, invitations and account questions go to Cynoss support (support@cynoss.net). The application itself points there: "Ask Cynoss support for a licence." Pricing and the free trial are on www.cynoss.net.

5 Connecting a repository

The **Connect repository** wizard brings a project from your organisation's repository into this browser. Its first line states the rule: "SafeSuite reads and writes project files only in your organisation's repository. Nothing is stored on SafeSuite servers."

Open it with **Connect repository** on the start page. The steps are shown at the top: **Host** › **Sign in** › **Project** › **Branch** › **Download** › **Check out**.



Connect repository

SafeSuite reads and writes project files only in your organisation's repository. Nothing is stored on SafeSuite servers.

Host › Sign in › Project › Branch › Download › Check out

- ☐ **GitLab.com**
gitlab.com
- ☐ **GitHub.com**
github.com
- ☐ **A Git repository in a folder**
On this computer or a network share

Only the hosts your administrator allowed are listed. They are the only servers that receive project data.

[Continue](#)

The Connect repository wizard

5.1 Host

Choose where the project lives. Only the hosts your administrator allowed are listed; they are the only servers that receive project data.

Host	Sign-in	Typical use
GitLab.com (or your organisation's GitLab)	Through GitLab's own page (OAuth)	Teams with projects on GitLab
GitHub.com	A fine-grained access token	Teams with repositories on GitHub
A Git server	Your user name on that server and an access token	Gitea, Forgejo, a self-hosted GitLab or any Git server over HTTPS

Host	Sign-in	Typical use
A Git repository in a folder	Your name and e-mail; no account	A bare repository on a network drive, or a clone on this computer. Chrome and Edge on a computer only

A host that cannot be used is greyed with the reason: "not configured for this deployment" or "this browser cannot open folders: use Chrome or Edge on a computer".

5.2 Sign in

GitLab. Press **Sign in with GitLab.com**. You are sent to GitLab to approve the access, then brought back. SafeSuite never sees your password. The sign-in is kept in this browser for every tab of the application, also after the browser closes, until you sign out.

GitHub. Create a fine-grained personal access token on GitHub (the wizard links to the page). Under *Repository access* choose the repositories SafeSuite may open; under *Repository permissions* give *Contents: Read and write*. Paste the token (it starts with `github_pat_`) and press **Sign in**. An expired or revoked token is reported as "GitHub.com no longer accepts the token (it has expired or was revoked). Sign in with a new one."

A Git server. Enter your **User name on the host**, an **Access token** made in the server's settings (repository read and write), **Your name** and **Your e-mail** as they should appear in the project's history. The token is checked when a repository is opened.

A folder. Enter **Your name** and **Your e-mail**; folders have no accounts, so this is how the history names you. The browser then asks you to choose the folder and to allow read and write access. After a reload, Chrome and Edge ask for that permission again; the application tells you and offers **Allow access**.

Note: A folder must be the repository's folder: a clone with its `.git` folder, or a bare repository. A SafeSuite project folder inside it is refused with "... is a SafeSuite project folder, not the repository that holds it." To share a folder repository with colleagues, put a bare repository on a shared drive (your administrator can make one with `git clone --bare`).

Tokens and keys are kept like any sign-in: in this tab and, with **Stay signed in** (on by default), in this browser, encrypted. They are cleared from the page after the check.

5.3 Project

- **GitLab:** search your projects, or paste a project's address. Each result shows its path, description and last activity.
- **GitHub:** search your repositories, type owner/name, or paste the address. A fine-grained token sees only the repositories chosen when it was made.

- **A Git server:** type the repository's path as it appears in its clone address, for example `team/brake.git`, and press **Open**. Copies already in this browser are listed as "Checked out in this browser".
- **A folder: Choose a folder...** opens the browser's folder picker. Folders chosen before are listed.

Hosted hosts show "Signed in to ... as ..." with **Sign out**.

5.4 Branch

The branches of the repository are listed with badges: **default**, **protected**, and, for a folder clone, **checked out · read-only here**. Pick the branch to work on; normally the default one. The branches SafeSuite uses for itself (reservations and backups) are hidden.

A folder clone's checked-out branch can be read and edited locally, but not shared into: Git refuses a change under the clone's own working files. Work on another branch, or on a bare repository that everyone shares.

A repository with no branches yet shows "This repository has no branches yet": push a first commit to it from the desktop application or with Git, then come back.

5.5 Download

Choose how much of the project comes into this browser:

- **Everything:** all files, downloaded now. Recommended for most projects.
- **Choose folders:** tick the top-level folders you need. The others stay listed in the explorer, greyed, and can be downloaded later when you open a file in them. Files at the top level are always downloaded.

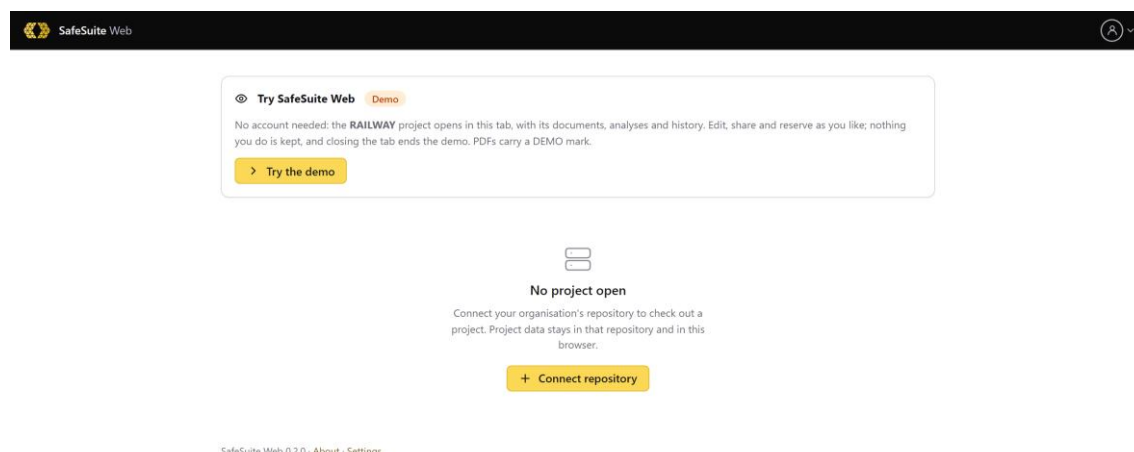
If the branch is already in this browser, the wizard offers **Open existing** or **Check out again** (which replaces the local copy; a copy with unshared changes asks first).

If your backup branch exists, because an earlier working copy of yours had unshared changes, the wizard says "Unshared work is backed up." and lists the files. Keep **Restore it into this checkout** ticked to get them back. The checkout then starts from the version the backup was made on; use **Get updates** afterwards.

5.6 Check out

Press **Check out**. A progress bar counts the files; each is checked against its repository identity before it is stored. When done, the project opens in the workspace. From now on it is listed on the start page and opens from the browser.

5.7 The start page: projects in this browser



The start page with the projects of this browser

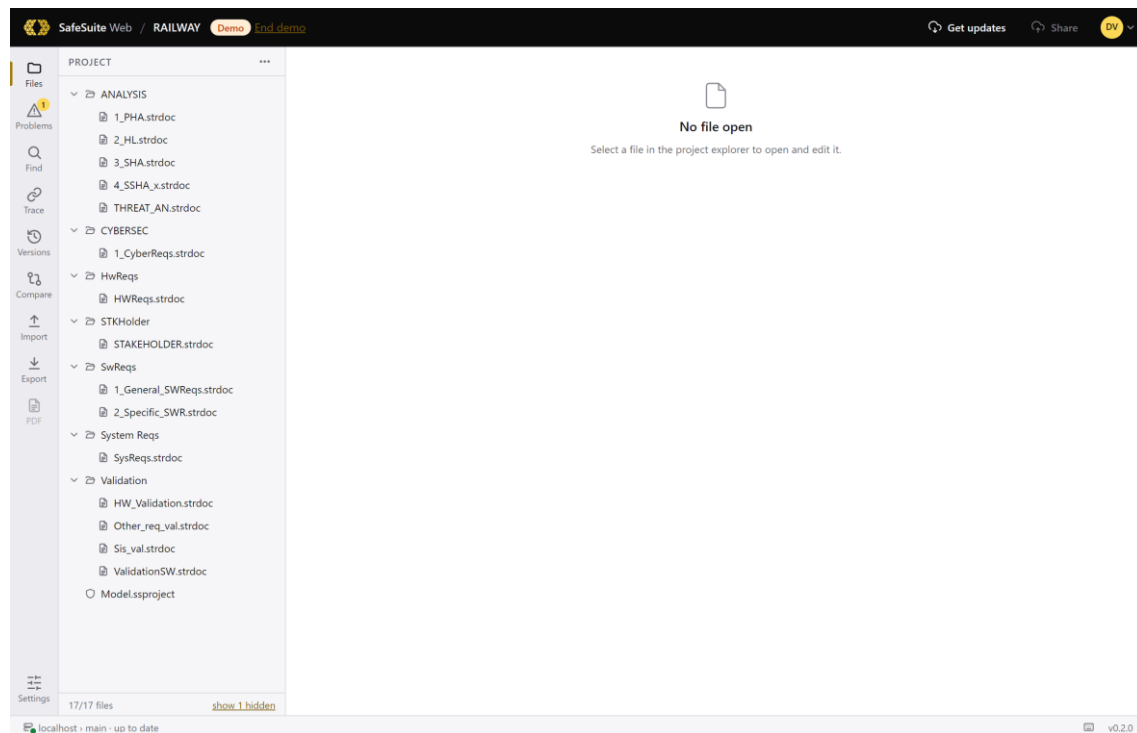
- Each working copy shows the project's name, the repository path, the branch, the host and when it was last opened. Click it to open the project.
- The trash button removes the copy from this browser. The dialog **Remove from this browser** says whether unshared changes would be lost ("N change(s) have not been shared...") or "Everything in it has been shared." Reservations you hold stay in the repository until released.
- If the browser deleted a copy on its own, the start page says so, with **Check out again**.
- Links from Jira, e-mail or chat (.../#document/.../item/...) open on the start page and find the document in any project of this browser: "The linked document opens here, in the browser." If the project is not in this browser, the message says which project to check out. **Open in the desktop app** is always offered as an alternative.

5.8 For your administrator

The hosts are listed in the site's configuration (Repositories:Hosts): GitLab needs an OAuth application whose redirect lists the site's /callback address; GitHub works with tokens; a Git server over HTTPS must answer the browser's cross-origin requests (the wizard shows the exact headers under **For your administrator: what the host needs**, with an nginx example); the folder entry can be removed to switch folders off.

6 The workspace

The workspace is the screen where a project is open. This chapter names its parts; the following chapters describe each tool in depth.



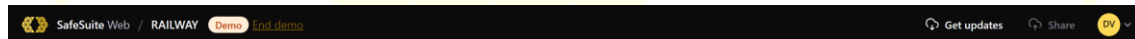
The workspace: top bar, tool rail, project explorer, document tabs, editor and status bar

6.1 The parts of the screen

Part	Where	What it holds
Top bar	Across the top	The SafeSuite Web mark (a link to the start page), the owner and name of the project, Get updates , Share , and the account button with your initials
Tool rail	Down the left edge	One button per tool: Files , Reviews , Problems , Find , Trace , Versions , Compare , Import , Export , PDF , Govern and, at the foot, Settings
Side panel	Next to the rail	The project explorer, or the Reviews list, or the Problems list, depending on the rail button pressed. Its width can be dragged

Part	Where	What it holds
Document tabs	Above the editor	One tab per open file; a dot marks unsaved edits
File header bar	Under the tabs	The open file's path, History... , the unshared chip, the lifecycle chip, Side by side... , editing and reservation chips
Editor	The centre	The open document, analysis or file
Versions pane	Docked on the right when open	Changes, History, File history, Item history, Baselines, and Activity log with governance
Status bar	Across the bottom	The repository state, the safety of your unshared work, your reservations, the licence, the keyboard shortcuts, the version and the messages bell

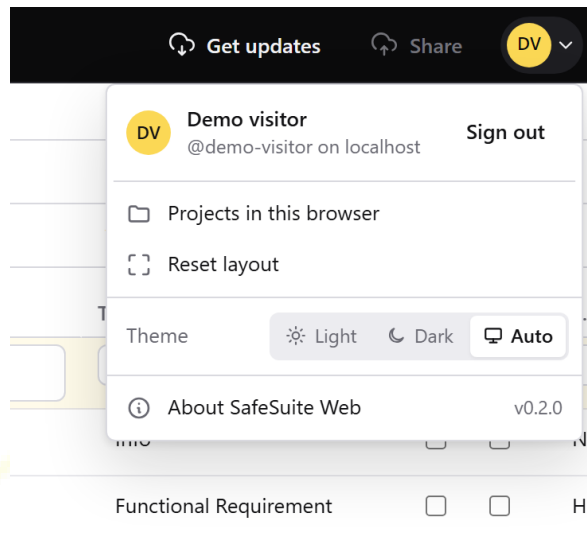
6.2 The top bar



The top bar

- The **SafeSuite Web** mark returns to the start page ("the projects in this browser").
- The project name, preceded by its owner, is a link: "switch to another project in this browser".
- In a demo, a **Demo** badge and **End demo** follow the name.
- **Get updates** takes the changes others have shared. A gold dot on it means others shared new versions; a number means files changed on both sides wait for your decision. The tooltip spells it out.
- **Share** turns gold with the number of files when you have unshared changes. With nothing to share it is disabled: "Share: nothing to share yet (your saved changes are shared from here)".
- A thin progress line under the bar shows while a repository operation runs.
- The **account button** shows your initials, or a person outline when nobody is signed in. It opens the account menu.

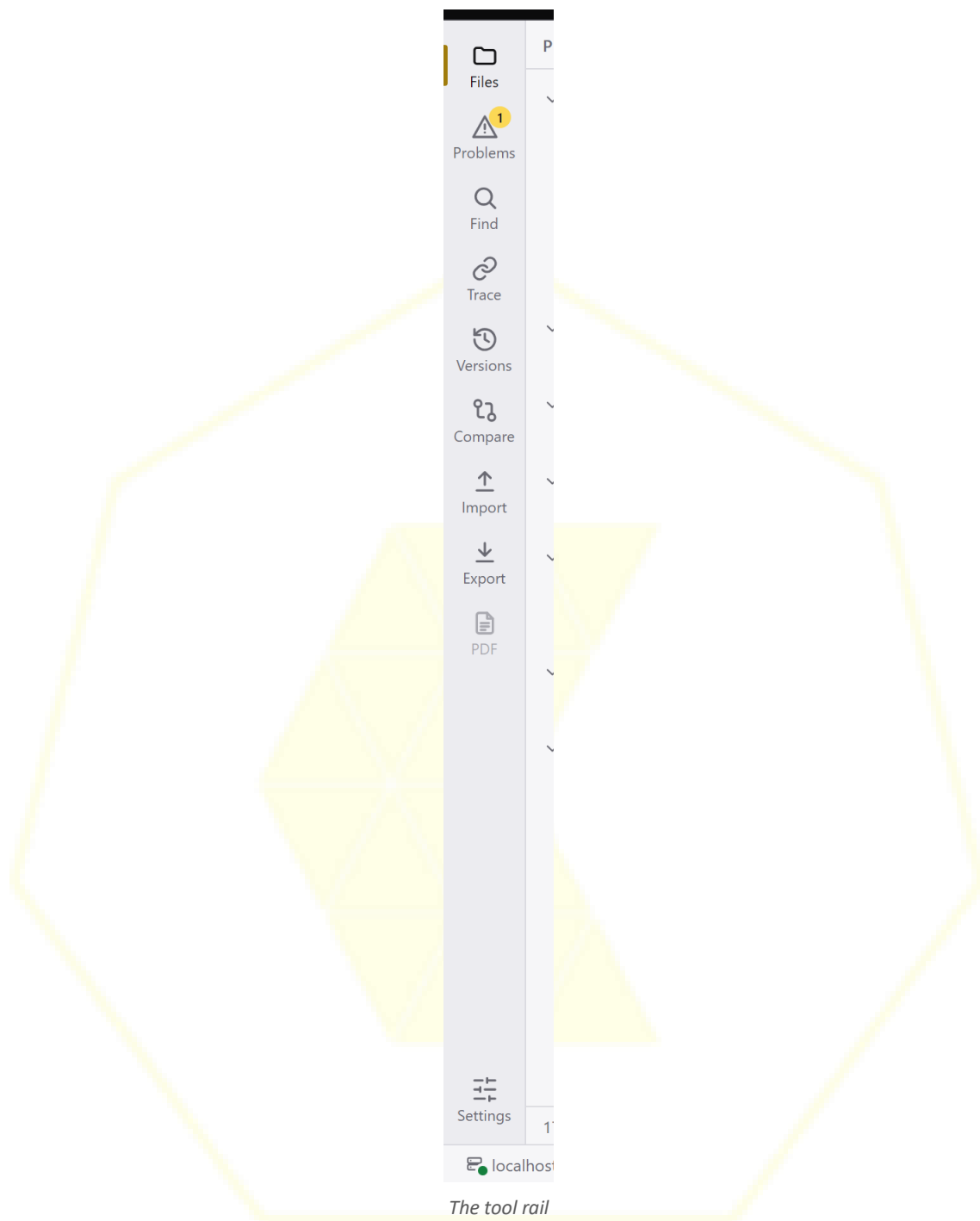
6.3 The account menu



The account menu

Entry	What it does
Your name and "@login on host", with Sign out	Who the repository knows you as. Sign out forgets this account in this browser, to sign in again or as someone else. Signed out, it shows Sign in instead
Identity and signing key...	With governance on: your signing identity (chapter 15)
Recent projects	Up to nine other projects opened in this browser, each with its host
Projects in this browser	The start page
Copy without governance...	Governed projects only (chapter 15)
Reset layout	Restores the panel widths and the layout as they first came
Theme: Light, Dark, Auto	Auto follows your device
Install app	While the browser offers it: SafeSuite opens in its own window, and the browser keeps its local copies even when the disk fills up
About SafeSuite Web	Version, build, copyright and third-party components

6.4 The tool rail



Each button's tooltip names its single-key shortcut, which works while you are not typing in a field and no dialog is open.

Button	Key	What it opens
Files	F	The project explorer in the side panel. A second press hides the panel
Reviews	R	Governance only: the records waiting for you, and the work sent for review
Problems	P	Review markers (TODO, TBD, FIXME, WARNING, ERROR), invalid values, broken links and documents that cannot be read. The badge counts them
Find	Ctrl+F in the tree	The explorer's find bar: files and folders by name
Trace	T	The Traceability dialog: links between documents, coverage and gaps
Versions	H	The Versions pane, docked on the right
Compare		Compare and merge two files from your computer
Import	I	ReqIF, Excel or CSV into new documents; imports into the open FMEA, FIT prediction or fault tree
Export	E	The open file's formats, and Excel, ReqIF or the traceability workbook for structured documents
PDF		The open document's PDF. Disabled until a structured document, an FMEA, a FIT prediction or a fault tree is open
Govern	G	Governance only: policy, seals, approvals, waivers, delegations, audit log
Settings	,	Experimental features, kept in this browser for every project

6.5 The side panel

The side panel shows the explorer by default. Its header reads **Project** with a **...** button, **Project actions**; its footer counts the files, "17/17 files" (downloaded of total), and offers **show N hidden**.

The splitter between the panel and the editor resizes it: drag it, or focus it and use the arrow keys; a double-click restores its width. The width and the panel shown are remembered per browser.

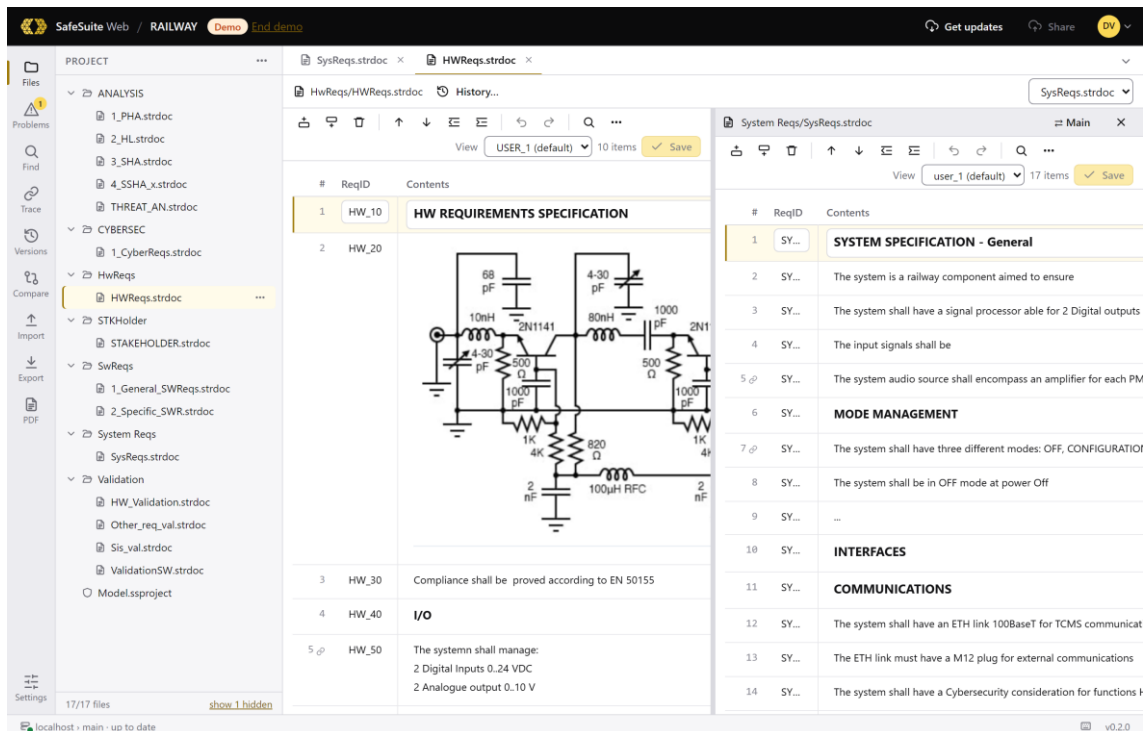
6.6 Document tabs and the file header

- Each open file has a tab. An unsaved file shows a dot. A middle-click closes a tab, tabs can be dragged to reorder, and a focused tab answers to the arrow keys and **Delete**.
- The tab's menu (right-click) offers **Close**, **Close All But This**, **Close All**, **Save All (Ctrl+Shift+S)**, and **Show Beside** or **Close Side by Side**.
- The list at the end of the tab strip, **The open documents**, names every open file ("name " when unsaved) and has **Save All***.
- Reopening a project restores up to ten open files and the expanded folders.



The file header bar shows the file's icon and path (the tooltip gives its kind, size and who reserved or is editing it), then **History...** When the file has unshared changes it adds an **unshared** chip with **Compare** and **Revert...** **Side by side...** lists the other open files to show one beside this one. A chip "Elena Vidal is editing since 10:42" appears while a colleague edits the file; **Locked by you · 13 d left** with **Release**, or **Break lock...** for someone else's reservation, when the file is reserved.

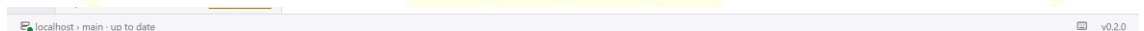
6.6.1 Side by side



Two documents side by side

Side by side... (or **Show Beside** in a tab's menu) shows a second file next to the open one. The file beside has its own header with **Main** to swap sides and **x** to close the view (the file stays open). Drag the divider to share the width; double-click it for half and half. Keyboard shortcuts act on the pane used last. A row dragged from one structured document onto a row of the other creates a link between them (chapter 8).

6.7 The status bar

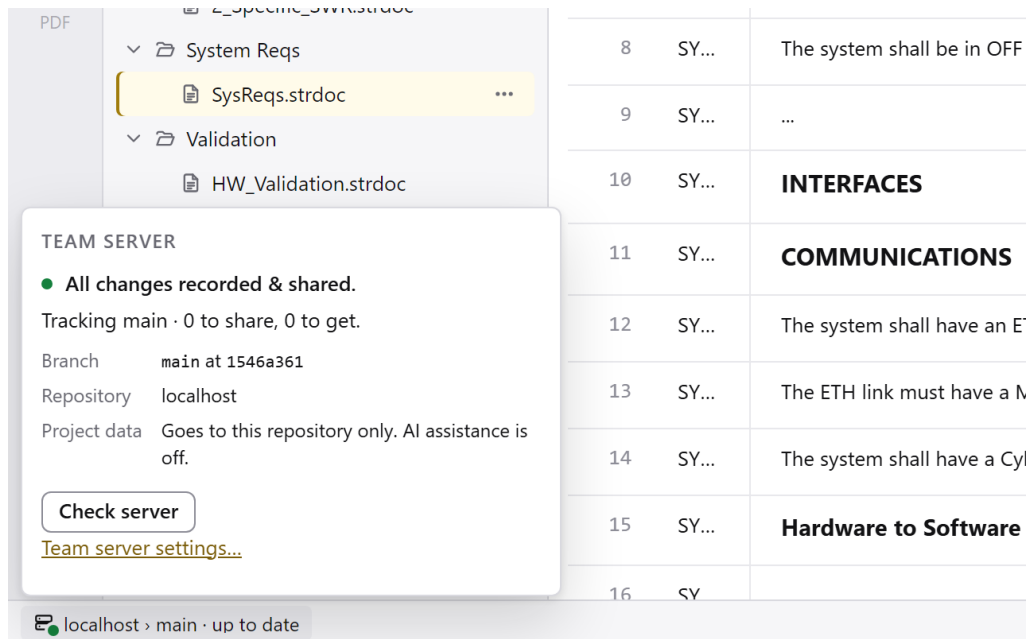


The status bar

The status bar is hidden on phones; its facts are then on the phone's Home and More pages.

Item	Where	What it shows and opens
Repository	Left	"host › branch · state". States: "up to date", "N to share", "N to get · M to share", "N decisions", "not reachable", "not checked", "checking...". Its panel, Team server , gives the branch and version, the repository, "Project data: Goes to this repository only. AI

Item	Where	What it shows and opens
		assistance is off.", the last outcome, and the buttons Check server , Get updates or Resolve decisions... , Share...
Safety	Left, with unshared work	"backed up 10:42", "backup failed", "backup paused", "not protected" or "in this browser". Its panel, Your unshared work , explains the backup branch and offers Share... , Restore it , Discard it... , Local backups...
Reservations	Left, when you hold any	A lock with a count. Its panel, Reserved by you , lists each file with Release
Identity or Governed project	Right	With governance on, your name or "Identity not set up". In a governed project with governance off, Governed project (opens Settings)
Licence	Right, on licensed sites	The licence status light (chapter 4)
Keyboard	Right	The Keyboard shortcuts legend (also the ? key)
Version	Right	"v0.2.0": opens About
Messages	Right	The bell lists the latest messages

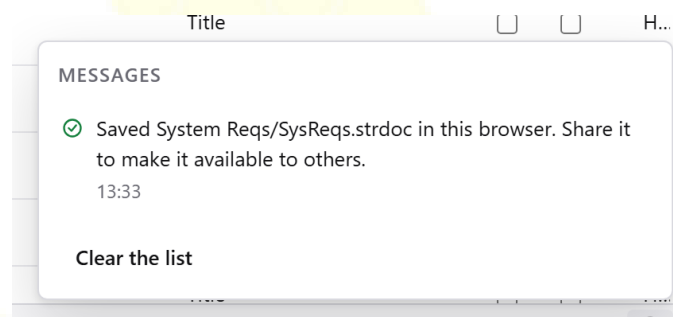


The Team server panel

6.8 Messages

Messages appear as small cards at the bottom right. Successes leave on their own after a few seconds; warnings and errors stay until dismissed; a repeated message counts "x2". Many carry a button that acts on the message, for example **Show backups** or **Sign in**.

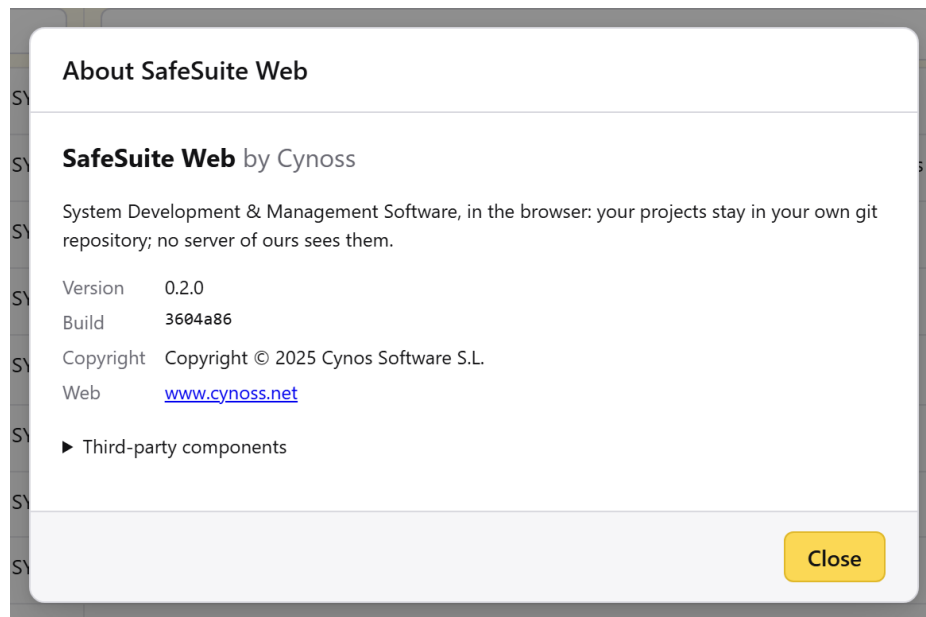
The bell in the status bar (key **M**) lists the latest thirty messages with their buttons while they still apply, and offers **Clear the list**. **Esc** dismisses the newest message when no panel or dialog is open.



The messages list

6.9 Themes, layout, About and new versions

- **Theme**, in the account menu: **Light**, **Dark** or **Auto** (as the system).
- **Reset layout** restores the panel widths and views: "The layout is back as it first came."
- **About SafeSuite Web** shows the version, the build, the copyright, the web address and the list of third-party components.
- When a new version of the application has been downloaded, a message says "A new version of SafeSuite Web is ready. Save your open edits, then reload to use it." with **Reload**. Nothing changes until you reload.

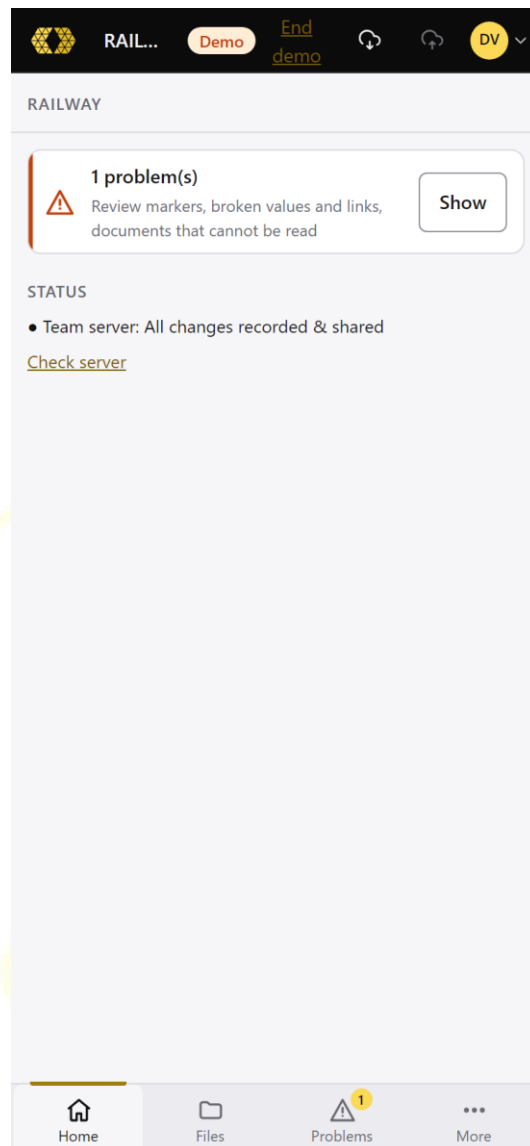


About SafeSuite Web

6.10 On a phone

In a window at most 760 pixels wide the application changes its layout:

- the top bar keeps the mark, the project name, **Get updates** and **Share** as icons with their dot or count, and the account button;
- the tool rail becomes a bottom bar: **Home**, **Files**, **Reviews** (with governance), **Problems**, **More**;
- there is no status bar: its facts are on **Home** under **Status**, and at the foot of **More**;
- dialogs become full-screen sheets, and the open tabs fold into one button that opens the **Open documents** sheet.



The phone layout: Home

Home on a phone is called "What needs you": cards for what waits for your review, a team update, your unshared changes and the problems; then **Recent documents** and **Status**. When nothing waits it says so.

More lists **Find**, **Traceability**, **Versions**, **Compare files**, the imports and exports, **PDF**, **Governance** and **Identity** with governance, **Settings**, **Keyboard shortcuts**, **Messages** and **Projects in this browser**, with a footer line naming the branch, the version, the repository and the application version.



The phone layout: a structured document as cards, and the More sheet

Structured documents read as cards, each opening a reading view and an edit form (chapter 8). FMEAs, FIT predictions and fault trees read as lists on a phone: they are edited on a larger screen, and they are never reserved from a phone.

7 The project explorer

The project explorer, under **Files** in the tool rail, lists the folders and files of the project and is where files are created, renamed, moved, uploaded, downloaded and deleted.

7.1 The tree

- Folders come first, then structured documents, then other files.
- A folder that is not downloaded (a partial checkout) is greyed with the hint "not downloaded". Open a file in it and press **Download folder** to fetch it.
- Files with unshared changes carry a hint: **new**, **edited** or **deleted** ("Saved in this browser, not shared yet").
- A file with a problem shows a problem icon instead of its file icon, with the problem in its tooltip: a file that cannot be read, an identifier that is not unique in the project, or repeated automatic identifiers.
- A lock means a reservation: "Locked by you" or "Locked by Elena Vidal". A grey pencil means "Elena Vidal is editing".
- The footer's **show N hidden** reveals dot-files and dot-folders (.safesuite, .gitignore...) and the items excluded from the listing.

Clicking a file opens it. Right-click, the row's ... button, **Shift+F10** or the Menu key open the item's menu; the empty area below the tree opens the project's menu.

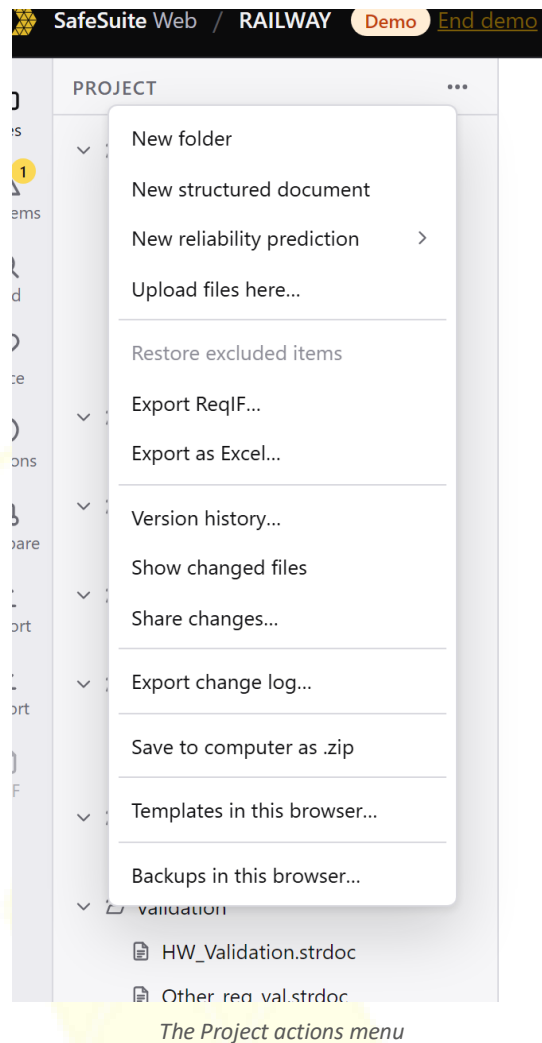
7.1.1 Keys in the explorer

Key	Action
↑ ↓, Home, End	Move; → and ← expand and collapse a folder
Enter or Space	Open a file; expand or collapse a folder
F2	Rename
Delete	Delete (asks first)
Ctrl+X, Ctrl+C, Ctrl+V, Ctrl+D	Cut, copy, paste, duplicate
Ctrl+Z, Ctrl+Y (or Ctrl+Shift+Z)	Undo and redo
Ctrl+F	Find files and folders by name; Enter and Shift+Enter move between matches
Shift+F10	The item's menu

7.1.2 Drag and drop

Drag a row onto a folder to move it; dropping on a file means its folder, dropping on the empty space means the project root. The target folder is highlighted while you drag; a refused place is not. Files cannot be dropped from the computer: use **Upload files here....**

7.2 Project actions

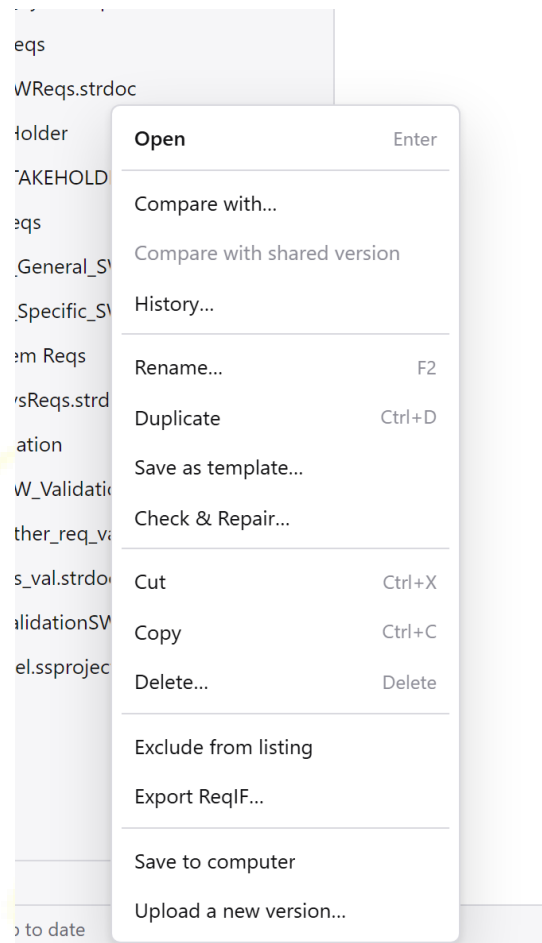


The ... button in the panel's header, or a right-click on the empty area, opens the project's menu:

Entry	What it does
Undo: ... / Redo: ...	The last explorer step and the step undone, by name
New folder	A new folder, named in a small dialog
New structured document	A new document from a template. With several templates, a submenu lists them: the built-in Blank Requirements Document ("The standard requirements table: ID, Contents, Type, FuSa and Cyber flags."), the project's own templates (from its .templates folder), and your personal templates, marked "(your own, kept in this browser)"

Entry	What it does
New reliability prediction ▶ IEC 62380, MIL-217, IEC 61709	A new FIT prediction of that standard
New project...	Only where no project exists yet
Upload files here...	Files or a folder from your computer
Paste	After a cut or a copy
Restore excluded items (N)	Brings back items excluded from the listing
Show effective policy here...	Governance only
Export ReqIF..., Export as Excel...	The whole project (chapter 14)
Version history...	The project's history in the Versions pane
Show changed files	The Versions pane at Changes
Share changes...	The Share dialog
Export change log...	A CSV of the project's history
Save to computer as .zip	The whole project as a zip. Files not downloaded are left out, and the message says so
Templates in this browser...	Your personal document templates
Backups in this browser...	The versions replaced in this browser (chapter 12)

7.3 Folder and file menus



A file's menu

A folder's menu repeats the **New ...** entries and **Upload files here...**, then **Rename...**, **Cut**, **Paste**, **Delete...**, **Exclude from listing** or **Restore to listing**, **Export ReqIF...**, **Export as Excel...**, **Version history...**, **Show changed files**, **Share changes in folder...**, **Export change log...** and **Save to computer as .zip**. With governance it adds **Create baseline from folder...** and **Show effective policy here...**

A file's menu:

Entry	What it does
Open	Opens the file (Enter)
Release reservation	When you hold the file's reservation
Project FIT results	On a FIT prediction: writes its results as a new structured document
Compare with...	Another project file of the same type, or a file from your computer

Entry	What it does
Compare with shared version	Your unshared change against the shared file
Share...	The Share dialog with this file ticked, when it changed
History...	The file's history in the Versions pane
Show change log	Governance only
Rename...	A new name (F2)
Duplicate (Ctrl+D), Copy (Ctrl+C)	Structured documents and fault trees. A duplicate keeps the content; links from other documents are not duplicated
Save as template...	Structured documents: keeps the document's configuration as a personal template
Check & Repair...	Structured documents: diagnoses and repairs a damaged file (chapter 8)
Cut, Paste, Delete...	Move or remove the file
Exclude from listing / Restore to listing	Hides the item from the explorer, here and in the desktop application (.ssexclude)
Export ReqIF...	Structured documents
Save to computer	Downloads the file
Upload a new version...	Replaces the file with one from your computer; the replaced version is backed up

Some items are protected and refuse with a reason: the governance records (.safesuite), the project file (.ssproject), the repository's own files, and the folder that holds a project ("move or rename projects in the desktop app").

7.4 Creating folders, documents and predictions

1. Open the menu of the folder where the item goes (or **Project actions** for the project root).
2. Choose **New folder**, **New structured document** (and a template), or **New reliability prediction** and its standard.
3. Type a **Name** and press **Create**. Suggested names are offered.

A new document opens at once. Names follow the rules of the desktop application: none of \ / : * ? " < > |, no leading dot (a name starting with a dot would be hidden), no trailing dot or space, at most 200 characters, unique in its folder, and not a Windows device name such as CON or LPT1.

Note: New fault trees and new FMEAs are not created in the web application: duplicate an existing one, or make it in the desktop application. Chapter 17 lists these limits.

7.5 Uploading files

Upload files here... opens the **Upload files** dialog. Choose several files, or a folder with its subfolders. Tick **Replace a file with the same name** to overwrite; otherwise a number is added to the name. Replaced files are marked "replaces the file there" and the replaced versions are backed up in this browser. Limits: 50 files at a time, a folder of at most 200 files, 100 MB per file; larger files go through Git or the desktop application. Dot-files and project files in an uploaded folder are left out.

7.6 Renaming, moving, duplicating and deleting

- **Rename...** (F2) asks for the new name. Changing the extension asks **File rename:** "If you change a file name extension, the file may become unusable." with **Change it** or **Keep .ext.**
- **Cut** and **Paste**, or drag and drop, move an item. A name already used in the target folder asks **Name already used**, with **Move as** a new name or **Move**.
- **Duplicate** makes "Copy of ..." next to the original; **Copy** then **Paste** puts the copy elsewhere.
- **Delete...** explains that unshared files are kept in **Backups in this browser** and that **Undo** (Ctrl+Z) brings them back.
- Every explorer operation is a local change until you share it. Open tabs, reservations, presence and governance records follow a renamed or moved file.
- Moving items between two SafeSuite projects, outside every project, or into .safesuite or .git is refused, as is renaming, moving or deleting a file someone else has reserved.

7.7 Hidden and excluded items

Dot-files and dot-folders are hidden; **show N hidden** in the footer reveals them. **Exclude from listing** hides any other item from the explorer, in the web and in the desktop application, by listing it in the project's .ssexclude; **Restore to listing** and **Restore excluded items (N)** bring them back.

7.8 Find files and folders

Press **Find** in the rail, or **Ctrl+F** in the tree. Type part of a name: every match is marked, the bar counts "x of y", and Enter, Shift+Enter, ↓ and ↑ move between them. **Esc** closes the bar.

7.9 Templates in this browser

Save as template... on a structured document keeps its configuration (item types, attributes, views) under a **Name** and an optional **Description**, in this browser. **Project actions** ▸ **Templates in this browser...** lists them, with **Add from your computer...** and **Remove**. Templates kept in the project's .templates folder are shared with the whole team.

7.10 Partial checkouts

If the project was checked out with **Choose folders**, the folders left out are listed greyed. Opening a file in one shows "Not downloaded yet" with **Download folder**. Problems, traceability and PDFs leave

documents that are not downloaded out, and say so; **Download and analyse them** fetches them. **Get updates** downloads only the folders in your scope and updates the listing elsewhere.



8 Structured documents

A structured document (.strdoc) is the heart of SafeSuite: a table of items, requirements, hazards, test cases or anything else the team describes, with typed attributes as columns. This chapter covers the editor in full. Editing needs the Structured Document Editor licence; without it the document opens read-only and everything else in this chapter still works.

8.1 What a structured document is

- **Items** are the rows. Each has an identifier and an **item type** (for example Title, Info, Functional Requirement, Constraint) which decides which attributes it carries.
- **Attributes** are the columns: Plain Text, Rich Text, Integer, Real, Boolean, Date/Time, Selection or Computed. A document defines its own attributes in its **configuration**.
- **Outline levels** give the document a structure of sections and sub-sections: a title item with the items under it. Rows indent by level, and a section can be collapsed.
- **Links** connect an item with items of other documents, for traceability.
- **Views** decide which columns are visible, in which order and width.

8.2 Opening a document and the toolbar

Click the document in the explorer. It opens in a tab with its toolbar:

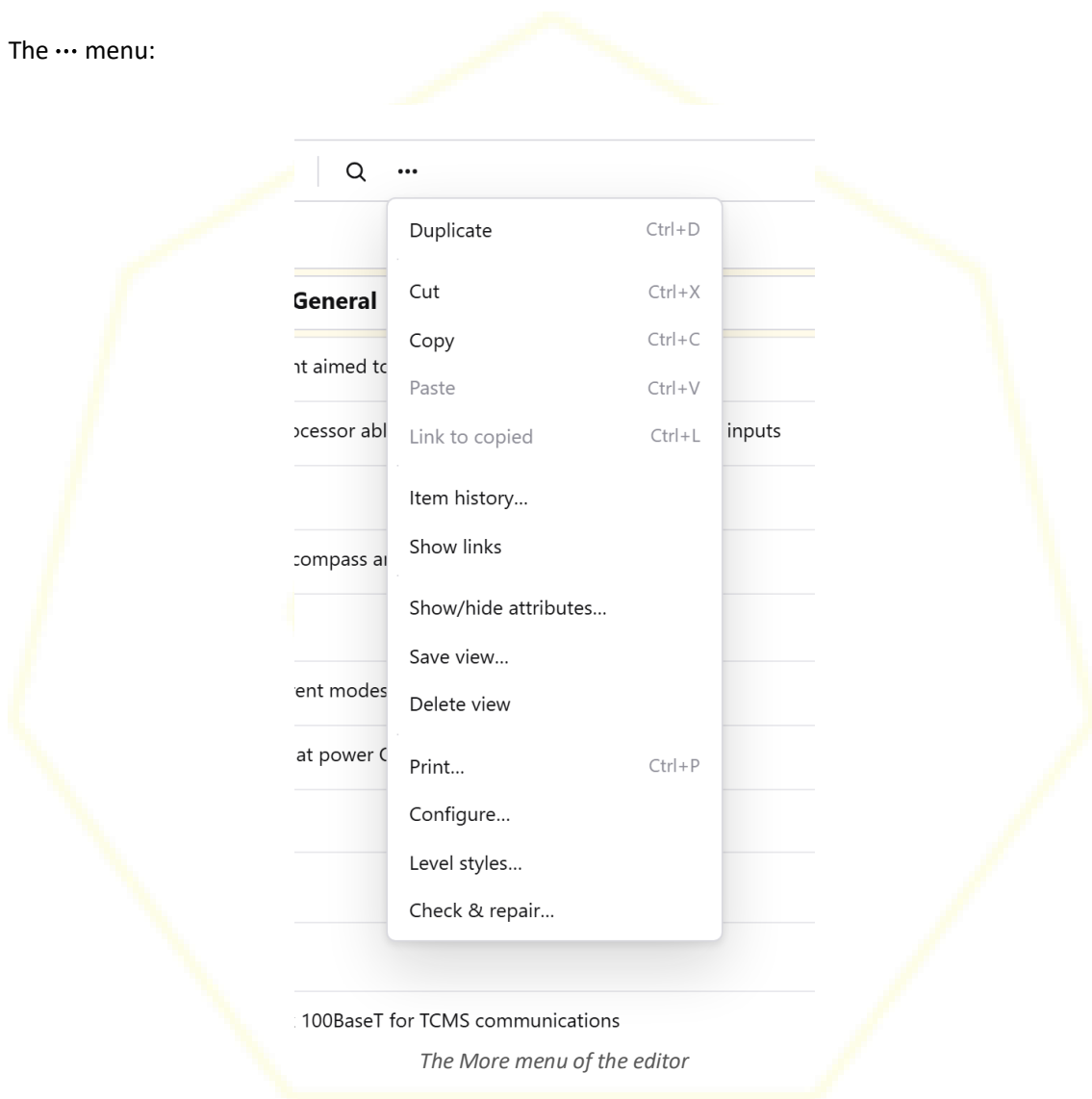


The toolbar of a structured document

Button	Key	Action
Above	Shift+Insert	A new item above the selected row
Below	Insert	A new item below the selected row
Delete	Delete	Deletes the selected rows
Move up, Move down	Alt+Shift+↑, Alt+Shift+↓	Moves the selected rows, with their section
Promote, Demote	Alt+Shift+←, Alt+Shift+→	Changes the outline level
Undo, Redo	Ctrl+Z, Ctrl+Y	By step; the tooltip names it, "Undo: Add document item"
Find	Ctrl+F	The find and filter bar
... More		The rest of the commands (below)
View		The list of saved views

Button	Key	Action
Problems count		Opens the document's problems bar
"17 items"		The count, or "n of m items" while filtered
Save	Ctrl+S	Saves into this browser

The ... menu:



Entry	Key	Action
Duplicate, Cut, Copy, Paste	Ctrl+D, Ctrl+X, Ctrl+C, Ctrl+V	Rows, within the same document
Link to copied	Ctrl+L	Links the selected rows to the rows copied last

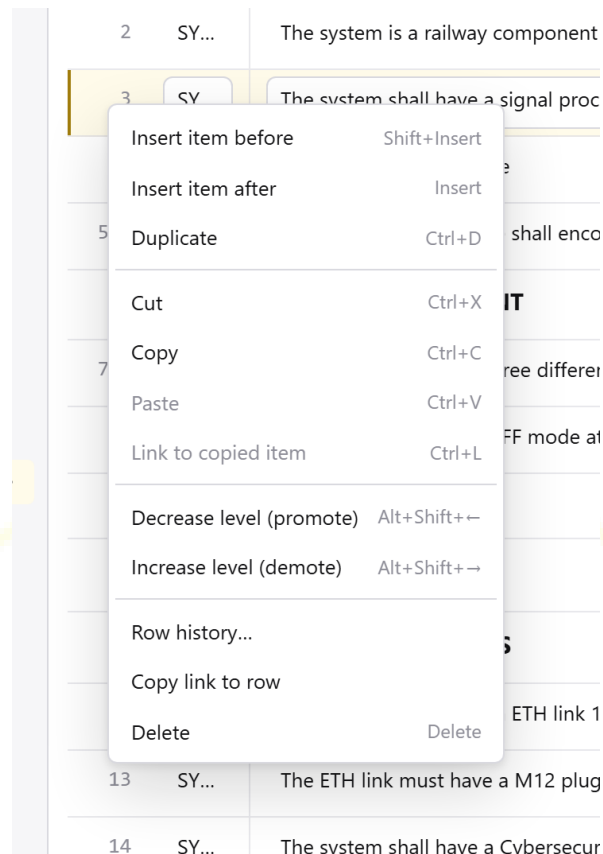
Entry	Key	Action
Item history...		The selected row's history in the Versions pane
Show links / Hide links		The Links panel
Show/hide attributes..., Save view..., Delete view		Views and columns
Print...	Ctrl+P	The print dialog (chapter 14)
Configure...		The document's configuration
Level styles...		How each outline level looks
Check & repair...		Diagnosis and repair of the file

8.3 Items: adding, moving, deleting

1. Select the row the new item relates to (click its row number; **Ctrl**+click and **Shift**+click extend the selection; **↑**, **↓**, **PageUp**, **PageDown**, **Home**, **End** and **Ctrl+A** also work).
2. Press **Below** (Insert) or **Above** (Shift+Insert). The new item takes the type and level of its neighbour.
3. Fill its cells.

Rows can be dragged by their number to move them with their section; this is disabled while a filter is on. **Promote** and **Demote** change the level. The row number column shows a chevron on sections, to collapse or expand them, a link mark with the count of outgoing and incoming links, and the Jira key when the row is linked to an issue.

The row's menu (right-click):



A row's menu

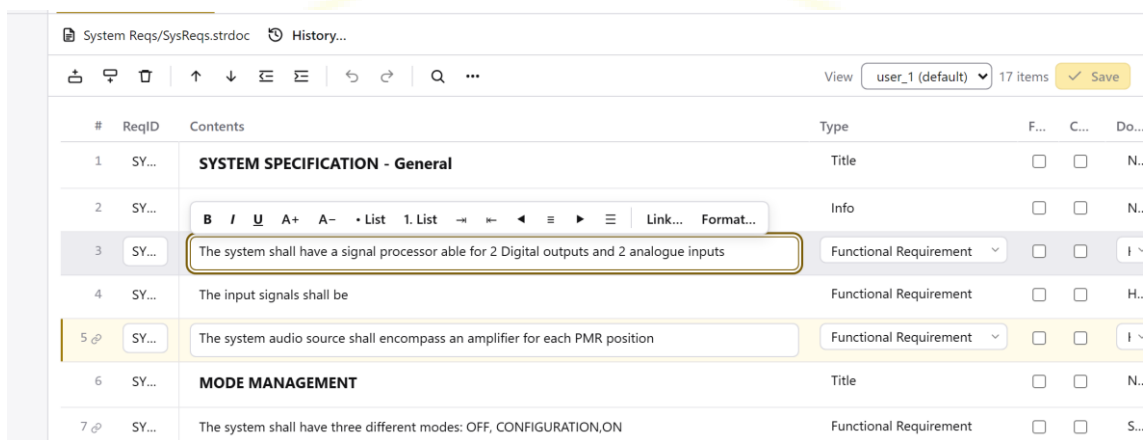
Insert item before, **Insert item after**, **Duplicate**, **Cut**, **Copy**, **Paste**, **Link to copied item**, **Decrease level (promote)**, **Increase level (demote)**, the Jira entries where the document has Jira columns, **Row history...**, **Copy link to row** and **Delete**. The empty area below the rows offers **Insert item** and **Paste**.

8.4 Editing cells

Click a cell to edit it. **Tab** and **Shift+Tab** move to the next or previous cell.

Attribute type	How it is edited
Plain Text	A multi-line text box
Rich Text	Edited in the cell, with a compact toolbar: B , I , U , larger and smaller text, lists, indent, alignment, Link... and Format... for the full dialog. Esc or Ctrl+Enter leaves the cell
Selection	One choice from a list, with <None> and a clear button; or several choices as chips with a drop-down
Boolean	A checkbox

Attribute type	How it is edited
Integer, Real	A number; the attribute's minimum and maximum are enforced
Date/Time	A date and a time (HH:mm); kept in universal time, shown in yours
Computed	Read-only: the value of an expression over the row's other attributes
Automatic identifier	Read-only: SafeSuite numbers it when the row is created



#	ReqID	Contents	Type	F...	C...	Do...
1	SY...	SYSTEM SPECIFICATION - General	Title	☐	☐	N..
2	SY...	B / I U A+ A- • List 1. List	Info	☐	☐	N..
3	SY...	The system shall have a signal processor able for 2 Digital outputs and 2 analogue inputs	Functional Requirement	☐	☐	I \
4	SY...	The input signals shall be	Functional Requirement	☐	☐	H..
5	SY...	The system audio source shall encompass an amplifier for each PMR position	Functional Requirement	☐	☐	I \
6	SY...	MODE MANAGEMENT	Title	☐	☐	N..
7	SY...	The system shall have three different modes: OFF, CONFIGURATION, ON	Functional Requirement	☐	☐	S...

A rich-text cell being edited

A row shows only the columns of its own item type; the other cells stay blank and name the type on hover.

8.4.1 The rich-text Format dialog

Format... opens the full editor for a rich-text cell: bold, italic, underline, text size (Small, Normal, Large, Title), bullet and numbered lists, indent and outdent, alignment, undo and redo. **Link...** points to a web address or to a project document; a link to a document opens it when clicked. **Picture...** inserts a PNG, JPEG, GIF or BMP of up to 10 MB, and **Picture properties...** sets its size. **Table** inserts a 2x2 table, extended with **+ Row** and **+ Column**. **Apply** writes the result into the cell. Pasted text is cleaned of foreign formatting. Formatting the web editor does not handle yet is shown but kept read-only, with a hint to edit it in the desktop application.

8.5 Selecting, copying and pasting rows

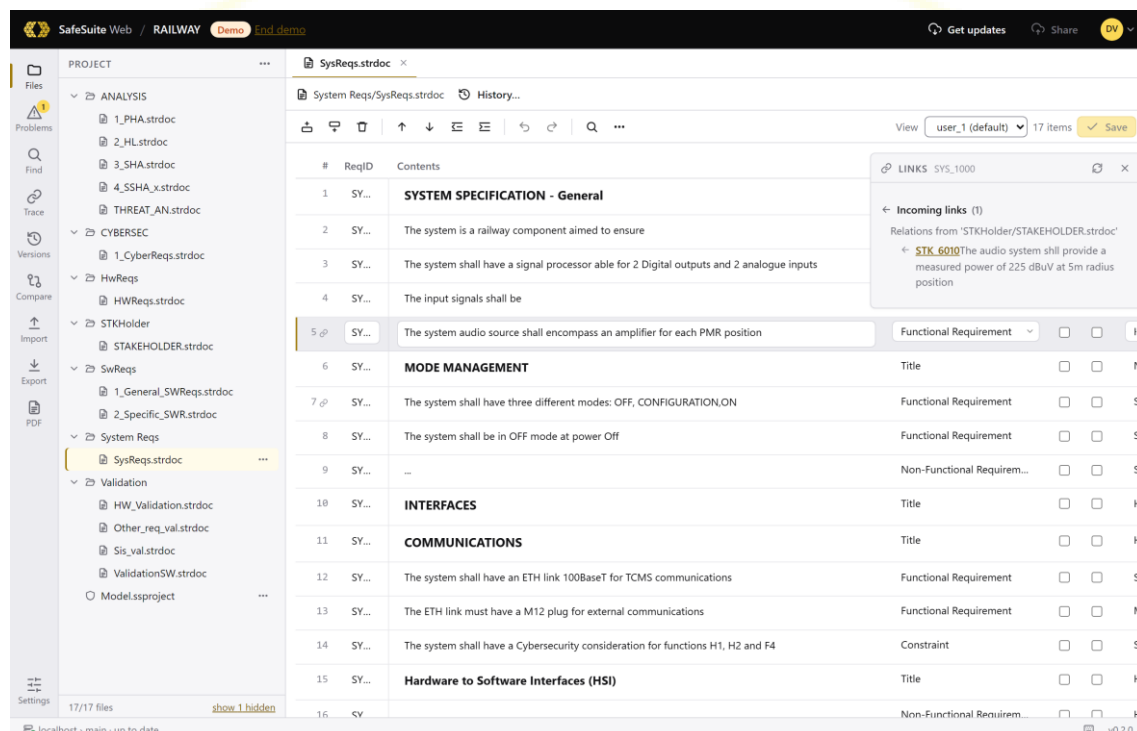
Copied or cut rows paste into the same document, in the same project, as new items. Closing the source document or switching project clears this row clipboard. (The FMEA worksheet and the FIT grids, by contrast, exchange cells with spreadsheets through the system clipboard; see chapter 9.)

8.6 Links between items

Links are what traceability is made of: a requirement links to the requirements it derives from, a test case links to what it verifies. Three ways to make one:

- **Copy, then link.** Copy the target rows (**Ctrl+C**) in their document, select the source rows in yours and press **Ctrl+L** (**Link to copied**). This works across documents of the project.
- **Drag between documents.** Show the two documents side by side and drag a row onto a row of the other document.
- **From the Links panel,** remove a link with **Remove Link**.

Show links (in the ... menu) opens the Links panel for the selected row: **Outgoing links** and **Incoming links**, grouped by document. Click a link to go to its row, or to open the other document at that row. Documents that are not downloaded are counted, with **Download them**.



#	ReqID	Contents	LINKS
1	SY...	SYSTEM SPECIFICATION - General	
2	SY...	The system is a railway component aimed to ensure	
3	SY...	The system shall have a signal processor able for 2 Digital outputs and 2 analogue inputs	
4	SY...	The input signals shall be	
5	SY...	The system audio source shall encompass an amplifier for each PMR position	Functional Requirement
6	SY...	MODE MANAGEMENT	
7	SY...	The system shall have three different modes: OFF, CONFIGURATION, ON	
8	SY...	The system shall be in OFF mode at power Off	
9	SY...	...	
10	SY...	INTERFACES	
11	SY...	COMMUNICATIONS	
12	SY...	The system shall have an ETH link 100BaseT for TCMS communications	
13	SY...	The ETH link must have a M12 plug for external communications	
14	SY...	The system shall have a Cybersecurity consideration for functions H1, H2 and F4	
15	SY...	Hardware to Software Interfaces (HSI)	
16	SV		

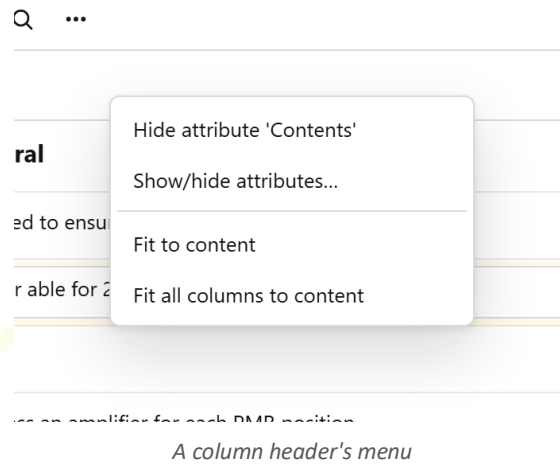
The Links panel of a row

Copy link to row puts an address on the clipboard, `.../#document/<id>/item/<id>?project=...`, that opens this document at this row in SafeSuite Web, from Jira, an e-mail or a chat. The link follows the document through renames and moves.

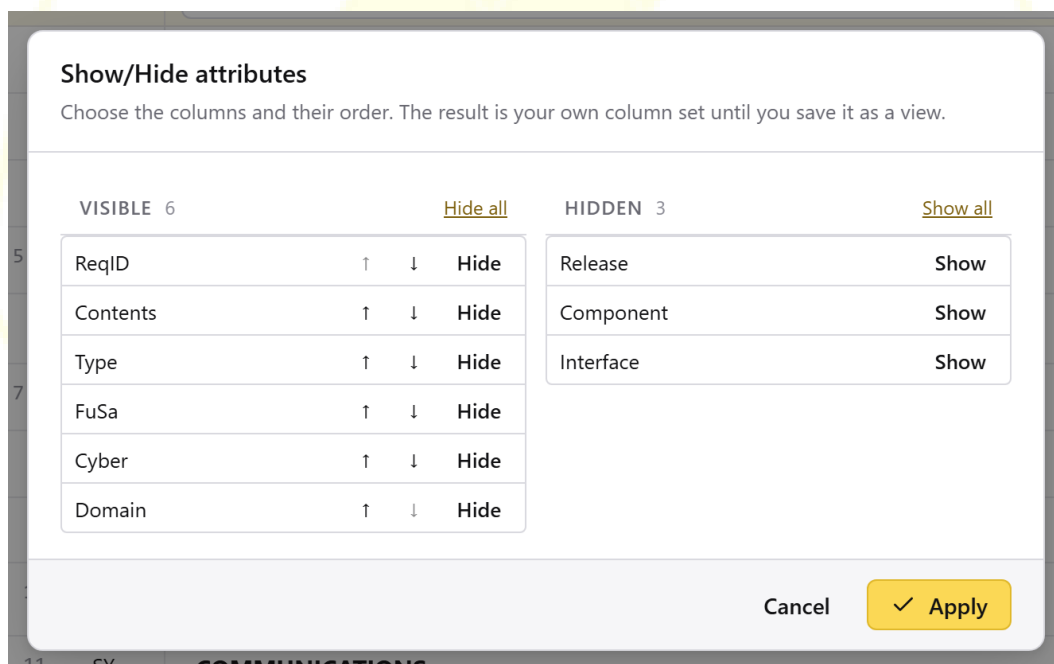
8.7 Views and columns

A **view** is a set of visible columns, in an order and with widths. The **View** list in the toolbar offers the document's shared views (one marked "(default)"), your private views "(private)", **All attributes**, and **Your columns** while you have changed columns without saving them. The last view used is remembered per document.

- Drag a column header to reorder; drag its edge to resize; double-click the edge to fit the content.
- The header's menu offers **Hide attribute**, **Show/hide attributes...**, **Fit to content**, **Fit all columns to content** and **Reset column widths**.



Show/hide attributes... lists the **Visible** and **Hidden** columns with **↑ ↓**, **Hide**, **Show**, **Hide all** and **Show all**; **Apply** confirms.



Show/hide attributes

Save view... asks for a **Name**, a **Description**, the scope, **Shared with the document** or **Only for me (kept in this browser)**, and whether it is **The document's default view**. Saving over an existing view asks for confirmation. **Delete view** removes the selected view.

Save view

Keeps the columns shown now, in their order and widths, under a name.

Name

Description

Scope

☒ Shared with the document

☐ Only for me (kept in this browser)

☒ The document's default view

This replaces the shared view of that name.

Cancel

✓ Save view

Save view

Tip: Views are part of the document when shared with it, so the whole team gets them at the next share. Private views stay in this browser.

8.8 Find and filter

Find: 1/3 1 1 Filter: Column: contains X

ReqID Contents Type F... C... Do...

The find and filter bar

Find (Ctrl+F) opens the bar. **Find** searches every column, hidden ones included ("in X, hidden in this view"), with the count "n/m" and ↑ ↓ (also **F3** and **Shift+F3**). **Filter** keeps only the rows that match, and **Column ... contains** narrows it to one attribute; the count reads "n of m items". Closing the bar clears the filter. The **/** key opens the find of the open document from anywhere in the workspace.

8.9 Document configuration

... ► **Configure...** opens **Document configuration**, where the document's item types and attributes are defined.

RAILWAY Demo

Get updates

Document configuration

The item types and their attributes, as in the desktop's Configuration. Changes apply at once; Undo takes them back one by one.

Item type

Default

+

Add type

✕

Remove type

Type name

Default

Type description

☐

Heading type

ATTRIBUTES

9

Component

Selection

Contents

Rich Text

Cyber

Boolean

Domain

Selection

FuSa

Boolean

Interface

Selection

Release

Selection

ReqID

Plain Text

Type

Selection

Plain Text

+

Add attribute

Contents

Rich Text

1

1

✕

Remove

Name

Contents

Description

Shown on the column header

TYPE

Change Rich Text to

—

▼

The values are converted; you see what would be lost before anything changes.

VALUES

Initial value

Blank lines separate paragraphs

Format...

☐

Main content

AUTOMATIC ID

☐

Generate identifiers for new items

The # between braces are the zero-padded number: REQ-{###} gives REQ-001, REQ-002... Existing values are kept; empty ones are numbered.

Every change is one step of the editor's Undo.

Undo

Redo

✓ Done

Document configuration

Item types. The list of types with **Add type** and **Remove type**; each has a **Type name**, a **Type description** and the flag **Heading type** for the types that make sections.

Attributes. The list of attributes in name order, a type selector and **Add attribute**. For each attribute:

Setting	Meaning
Name, Description	The column's name; the description is shown on the column header
Type	Plain Text, Rich Text, Integer, Real, Boolean, Date/Time, Selection or Computed. Changing the type of an attribute that holds data shows what would be lost, then Convert to ... or Convert anyway
Initial value, Maximum length, Main content	Text attributes
Minimum, Maximum	Numbers
New items start checked	Booleans

Setting	Meaning
Multiple selection, Allow no selection , the choices	Selections. Add choice and Remove choice (which asks what the rows holding the removed value get instead)
Expression	Computed attributes: an expression over the row's attributes in square brackets, for example <code>[Voltage] * [Current]</code> ; the Insert buttons add the names
Automatic ID	For a plain-text attribute: Generate identifiers for new items , a Pattern such as <code>REQ-{###}</code> , and Unique within the document or the item type

Undo, **Redo** and **Done** close the dialog. The dialog opens read-only on a read-only document. An attribute can only be removed while its column holds no data, which protects the content made by others.

8.10 Level styles

... ▶ **Level styles...** sets, per outline level, the size, bold, italic and colour of the item text, with a preview, so that titles and sub-titles stand out in the table and in the PDF.

8.11 Problems inside a document

The toolbar's problems count opens a bar with **Errors**, **Warnings** and **Messages** toggles. It lists invalid values (a number outside its limits, a selection out of range), broken links and configuration problems; each entry goes to the field. The project-wide view of the same problems, including the review markers TODO, TBD, FIXME, WARNING and ERROR written in the text, is the **Problems** panel (chapter 13).

A save is refused while the configuration has errors: "Not saved. The document contains errors. Please correct them before saving." with **Show the errors**.

8.12 Item history

... ▶ **Item history...** (or the row menu's **Row history...**) opens the Versions pane at **Item history**: each shared version that added, changed or removed the row, with the author, the date and the old and new value of every attribute; "not shared yet" marks your unshared change. **Follow the selected row** keeps it in step with the selection. In a governed document, **Changes since last approved** compares the row with the last approved version.

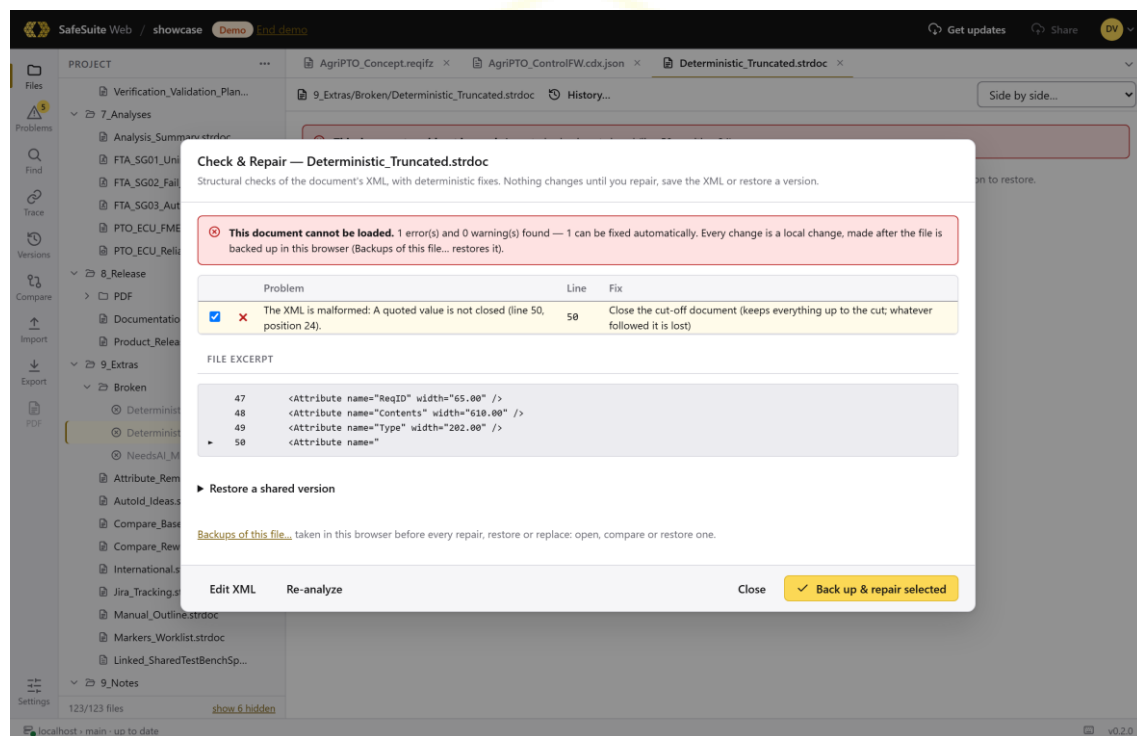
8.13 Saving and leaving

Ctrl+S or **Save** writes the document into this browser: "Saved ... in this browser. Share it to make it available to others." Closing the tab or leaving the project with unsaved edits asks **Unsaved**

modifications: Save, Don't save or Cancel. If a colleague shared a new version of the document while you were editing, the editor keeps your edits and warns you: compare with the shared version (**History...**) before saving over it.

8.14 Check & Repair

A document that has been edited by hand, merged by another tool or damaged in transit may refuse to open. **Check & Repair...** (in the file's menu, or by opening the broken document or its entry in Problems) diagnoses it:



Check & Repair

- A banner summarises the state. A table lists every problem with its severity, line and the fix that will be applied; where two fixes are possible, you choose. **File excerpt** shows the lines concerned.
- Tick the fixes and press **Back up & repair selected**. A backup is taken first; **Backups of this file...** lists them.
- **Edit XML** shows the raw file with line numbers for the cases a rule cannot cover; **Save & re-analyze** checks again.
- **Restore a shared version** replaces the file with the version in the repository, after a confirmation that offers to download the current file first.

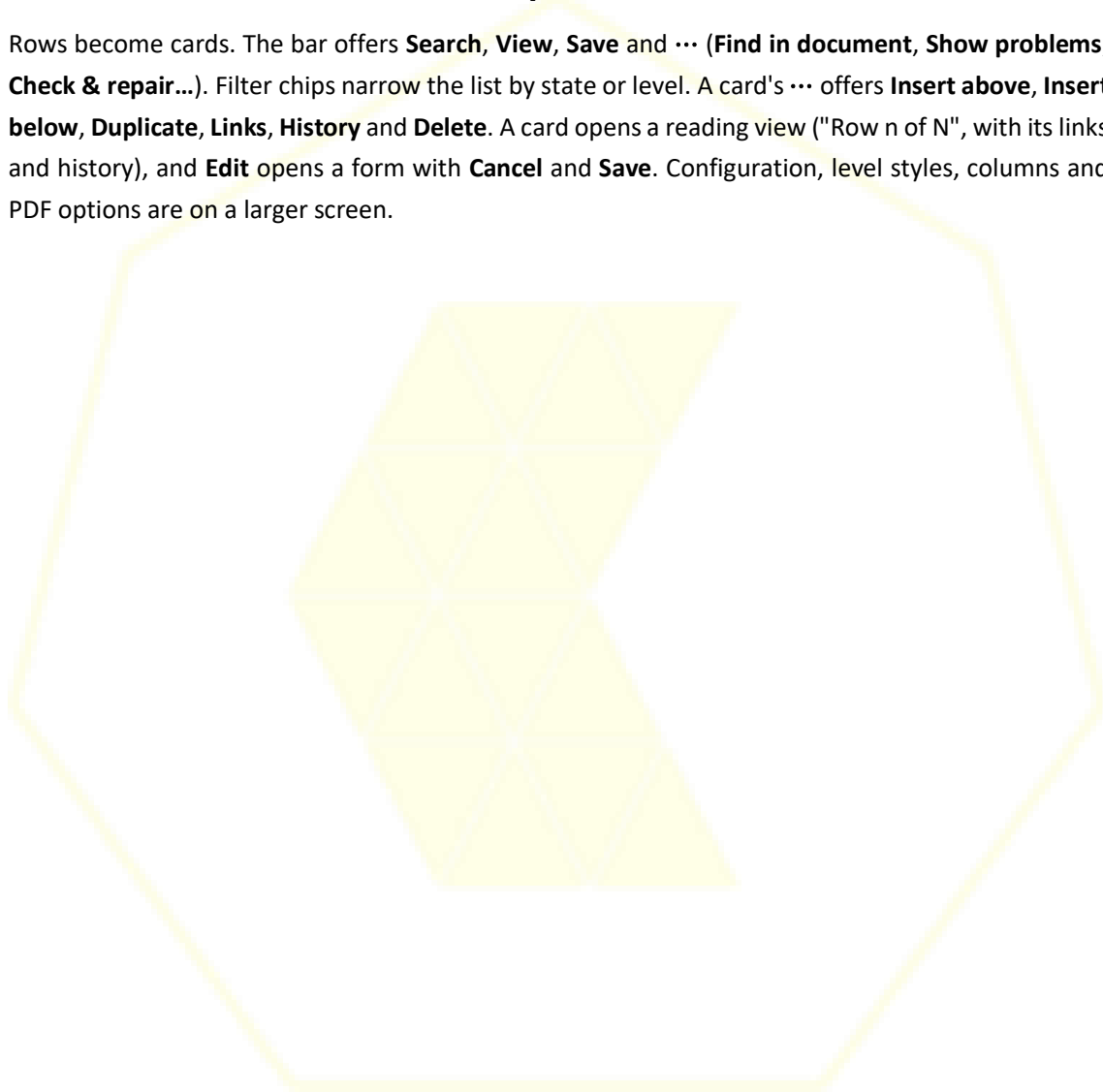
The desktop application's Check & Repair can additionally hand the file to the AI assistant, which the web application does not include.

8.15 Jira columns

A document bound to Jira in the desktop application shows the key of the linked issue in the row number column and the **Jira Status** and **Jira Assignee** columns. In the web application these cells are read-only: the browser does not reach Jira, and the statuses are refreshed by the desktop's **Refresh Jira statuses**. The row menu's **Link to Jira issue...** records an issue key or address, **Open in Jira** opens it in the browser and **Unlink from Jira** removes the link.

8.16 Structured documents on a phone

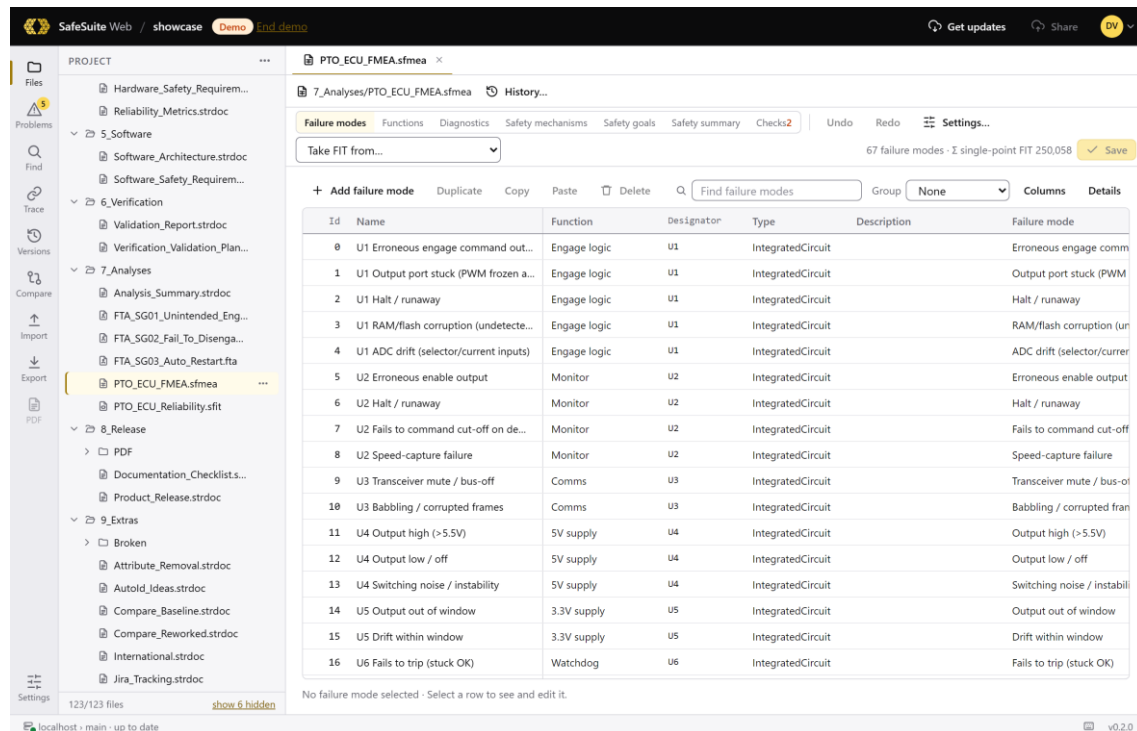
Rows become cards. The bar offers **Search**, **View**, **Save** and ... (**Find in document**, **Show problems**, **Check & repair...**). Filter chips narrow the list by state or level. A card's ... offers **Insert above**, **Insert below**, **Duplicate**, **Links**, **History** and **Delete**. A card opens a reading view ("Row n of N", with its links and history), and **Edit** opens a form with **Cancel** and **Save**. Configuration, level styles, columns and PDF options are on a larger screen.



9 Analyses: FMEA, FIT predictions and fault trees

SafeSuite Web opens and edits the three analysis formats of the suite: FMEAs (.sfmea), FIT reliability predictions (.sfit) and fault trees (.fta). They are files that cannot be merged, so your first edit of one reserves it for you (chapter 12). None of them needs a licence.

9.1 The FMEA editor



Id	Name	Function	Designator	Type	Description	Failure mode
0	U1 Erroneous engage command out...	Engage logic	U1	IntegratedCircuit		Erroneous engage comm
1	U1 Output port stuck (PWM frozen a...	Engage logic	U1	IntegratedCircuit		Output port stuck (PWM
2	U1 Halt / runaway	Engage logic	U1	IntegratedCircuit		Halt / runaway
3	U1 RAM/flash corruption (undetecte...	Engage logic	U1	IntegratedCircuit		RAM/flash corruption (un
4	U1 ADC drift (selector/current inputs)	Engage logic	U1	IntegratedCircuit		ADC drift (selector/curre
5	U2 Erroneous enable output	Monitor	U2	IntegratedCircuit		Erroneous enable output
6	U2 Halt / runaway	Monitor	U2	IntegratedCircuit		Halt / runaway
7	U2 Fails to command cut-off on de...	Monitor	U2	IntegratedCircuit		Fails to command cut-off
8	U2 Speed-capture failure	Monitor	U2	IntegratedCircuit		Speed-capture failure
9	U3 Transceiver mute / bus-off	Comms	U3	IntegratedCircuit		Transceiver mute / bus-o
10	U3 Babbling / corrupted frames	Comms	U3	IntegratedCircuit		Babbling / corrupted fran
11	U4 Output high (>5.5V)	5V supply	U4	IntegratedCircuit		Output high (>5.5V)
12	U4 Output low / off	5V supply	U4	IntegratedCircuit		Output low / off
13	U4 Switching noise / instability	5V supply	U4	IntegratedCircuit		Switching noise / instabi
14	U5 Output out of window	3.3V supply	U5	IntegratedCircuit		Output out of window
15	U5 Drift within window	3.3V supply	U5	IntegratedCircuit		Drift within window
16	U6 Fails to trip (stuck OK)	Watchdog	U6	IntegratedCircuit		Fails to trip (stuck OK)

The FMEA editor

An FMEA opens on its tabs: **Failure modes**, **Functions**, **Diagnostics**, **Safety mechanisms**, **Safety goals**, **Safety summary** and **Checks** (with a count); which tabs show depends on the FMEA's settings. The toolbar has **Undo**, **Redo**, **Settings...**, **Take FIT from...** (the component failure rates from a FIT prediction of the project, matched by designator), **Used by N fault trees** (a menu that opens each tree), the totals "N failure modes · Σ single-point FIT ...", and **Save**.

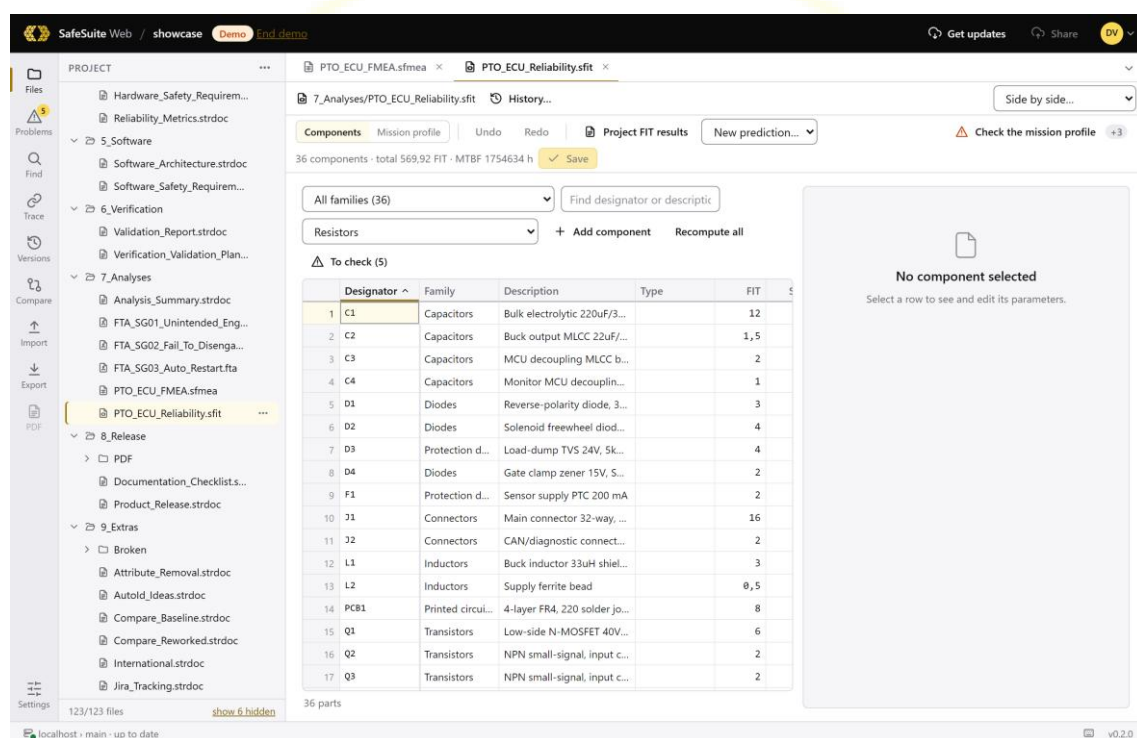
The worksheet. The bar offers **Add failure mode**, **Duplicate**, **Copy**, **Paste**, **Delete**, a **Find failure modes** box, **Group** (none, by component, by function), **Columns** with the hidden ones, and a **Details** toggle that opens the sections Identification, Modes and effects, Quantitative, Process, Diagnostics, Component, Safety/Criticality and Other for the selected mode. Cells edit in place; **Ctrl+Enter** fills a value into every selected failure mode; the row menu repeats the actions. Copy puts the rows on the system clipboard as tab-separated text, and **Ctrl+V** pastes cells from a spreadsheet or failure modes copied from another FMEA.

The supporting tables (Add function, Add diagnostic, Add safety mechanism, Add safety goal) each show the identifier and a "Used by" count. **Safety summary** gives the metrics SPFM and LFM against the ISO 26262-5 targets. **Checks** lists what to fix, with **Recompute** and **Record them**.

Settings... sets the FMEA's mode: Process or Product, Component quantitative, Diagnostic, Safety/Criticality, the ISO 26262 safety metrics and the Functional approach, and the FIT data file.

In the rail: **Export** ▶ **SafeFMEA CSV**; **PDF** ▶ **FMEA report...** (complete or modular, Modern or Classic design, page size, classification); **Import** ▶ **Into <file>: Items..., Functions..., Diagnostics..., Safety Mechanisms..., Safety Goals...** from CSV files of up to 64 MB.

9.2 The FIT prediction editor



The screenshot shows the SafeSuite Web interface for the FIT prediction editor. The main workspace displays the 'Components' tab, which lists 36 components with their FIT values. The 'To check' list shows 5 items. The sidebar on the right indicates 'No component selected'.

Designator	Family	Description	Type	FIT
1 C1	Capacitors	Bulk electrolytic 220uF/3...		12
2 C2	Capacitors	Buck output MLCC 22uF/...		1,5
3 C3	Capacitors	MCU decoupling MLCC b...		2
4 C4	Capacitors	Monitor MCU decouplin...		1
5 D1	Diodes	Reverse-polarity diode; 3...		3
6 D2	Diodes	Solenoid freewheel diod...		4
7 D3	Protection d...	Load-dump TVS 24V, 5k...		4
8 D4	Diodes	Gate clamp zener 15V, S...		2
9 F1	Protection d...	Sensor supply PTC 200 mA		2
10 J1	Connectors	Main connector 32-way; ...		16
11 J2	Connectors	CAN/diagnostic connect...		2
12 L1	Inductors	Buck inductor 33uH shiel...		3
13 L2	Inductors	Supply ferrite bead		0,5
14 PCB1	Printed circui...	4-layer FR4, 220 solder jo...		8
15 Q1	Transistors	Low-side N-MOSFET 40V...		6
16 Q2	Transistors	NPN small-signal, input c...		2
17 Q3	Transistors	NPN small-signal, input c...		2

A FIT prediction (IEC 62380)

A FIT prediction estimates the failure rate of the components of a design. The editor follows the standard of the file.

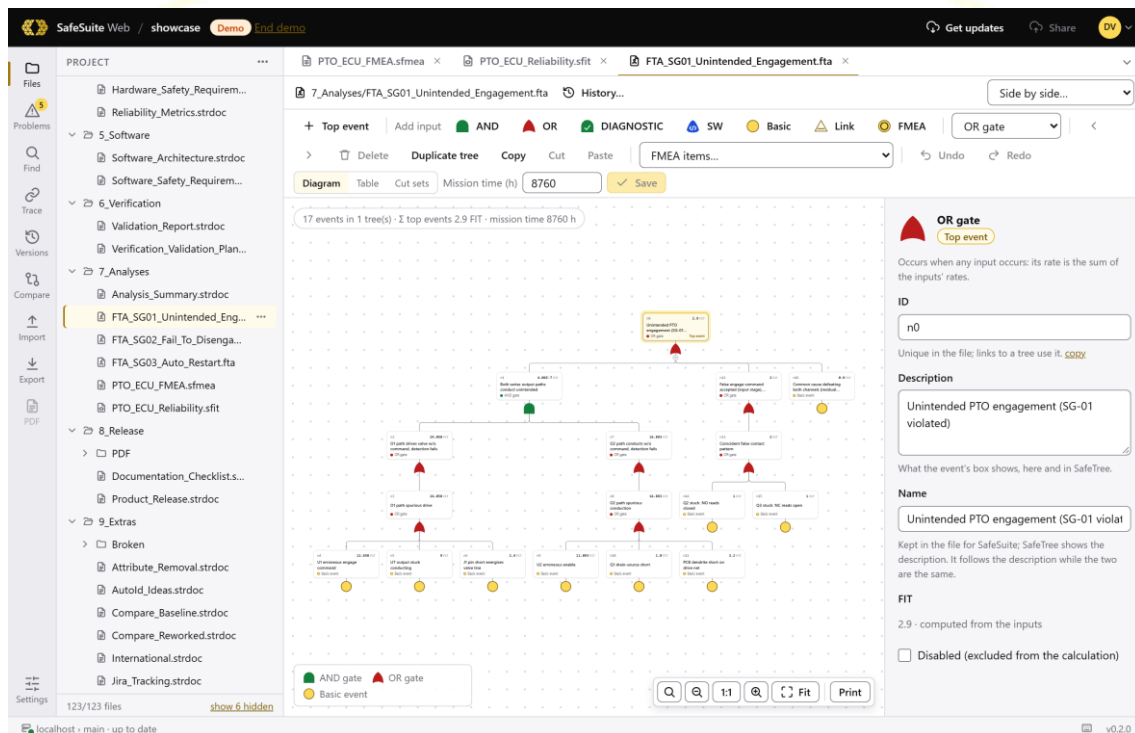
IEC 62380 has the tabs **Components** and **Mission profile**. The toolbar offers **Undo**, **Redo**, **Project FIT results** (the results as a new structured document), **New prediction...**, the checks, the totals "N components · total x FIT · MTBF y h", and **Save**. The Components tab lists the families with their counts and FIT, a find box, **Add component**, **Recompute all**, **To check (N)**, and the selected part's form: designator, description, the entered FIT that overrides the formula, the computed FIT and the parameters of the family. The Mission profile tab holds the profile in use, the saved profiles, and the grids of thermal phases and cycles.

MIL-217 and **IEC 61709** show the standard's badge, the tabs **Parts** and **Profile** or **Parts**, **Reference data** and **Settings**, and a **Compute** button.

The grids behave like a spreadsheet: arrows, **F2**, **F4**, **Tab**, **Enter**; **Ctrl+C** and **Ctrl+X** copy cells as tab-separated text, **Ctrl+V** pastes from the active cell and adds parts past the last row, **Ctrl+D** fills down. The row menu offers **Open the part's family**, **Copy**, **Paste**, **Fill down**, **Clear cells**, **Clear column**, **Add part**, **Duplicate row**, **Renumber designators...** (with a preview, and a list of the other files that still use the old designators), **Delete selected row** and **Delete all rows...**

In the rail: **Export** ► **SafeFit CSV...** (current, compact, fixed or complete layout); **PDF** ► **FIT report...**; **Import** ► **SafeFit CSV...** (complete or fixed, with **Overwrite**). The explorer's **New reliability prediction** creates a new file for any of the three standards.

9.3 The fault tree editor



The fault tree editor

The fault tree editor is on by default and can be switched off in **Settings** ► **Experimental** ► **Fault tree editor (FTA)**; off, a fault tree shows an empty state with **Turn on in Settings...** and **Download**.

The toolbar offers **Top event**, **Add input** with the kinds **OR**, **AND**, **DIAGNOSTIC**, **SW**, **Link**, **FMEA** and **Basic**, a type selector, move left and right, **Delete**, **Duplicate tree**, **Copy**, **Cut**, **Paste**, the FMEA items selector (**FMEA items...**, **Take from <fmea>**, **Clear the FMEA items**), **Undo**, **Redo**, the views **Diagram**, **Table** and **Cut sets**, the **Mission time (h)**, and **Save**.

Diagram draws the tree with a find box, **One tree**, **Unfold all**, zoom, **Fit** and **Print**, and a legend of the symbols. An event's menu adds inputs, changes its type, copies, cuts, pastes, duplicates the tree,

copies the identifier or deletes. The side panel lists the problems SafeTree cannot open, the trees, and the properties of the selected event: identifier, FMEA item, linked tree, description, name, FIT, diagnostic coverage and whether it is disabled. **Cut sets** lists the minimal cut sets and the events by their share of the top event.

In the rail: **Export** ▶ **Diagram as SVG, Diagram as PNG, Trees, events and cut sets as Excel; PDF** ▶ **Fault trees as PDF...** (Modern or Classic design); **Import** ▶ **FMEA items from a SafeFMEA CSV...** (up to 16 MB).

Fault-tree branches copy and paste across the fault-tree files open in the page.

9.4 Reservations for these files

Because an FMEA, a FIT prediction or a fault tree cannot be merged, SafeSuite Web reserves it for you at your first edit, with the note "editing in SafeSuite Web", and the desktop application honours the reservation. The file's header shows **Locked by you · 13 d left** with **Release**; the status bar counts your reservations. While someone else holds it, the file is read-only for you, with the reason and **Break lock....** Chapter 12 explains the rules.

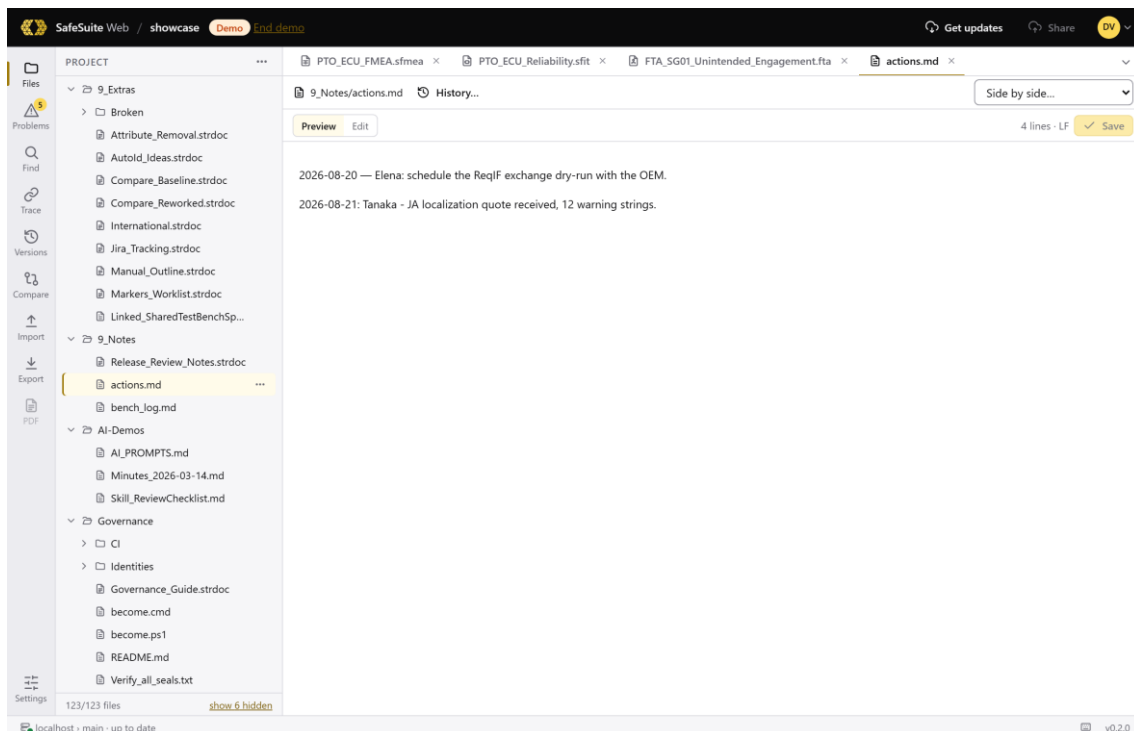
9.5 What the editors do not do

Phones show these three formats as read-only lists. New FMEAs and new fault trees are not created in the web application (duplicate one, or use the desktop application). FMEAs do not offer the AIAG-VDA action priority, rating tables or action tracking, which are not in SafeFMEA either; renumbering designators in a FIT prediction does not update the FMEAs that use them.

10 Other files

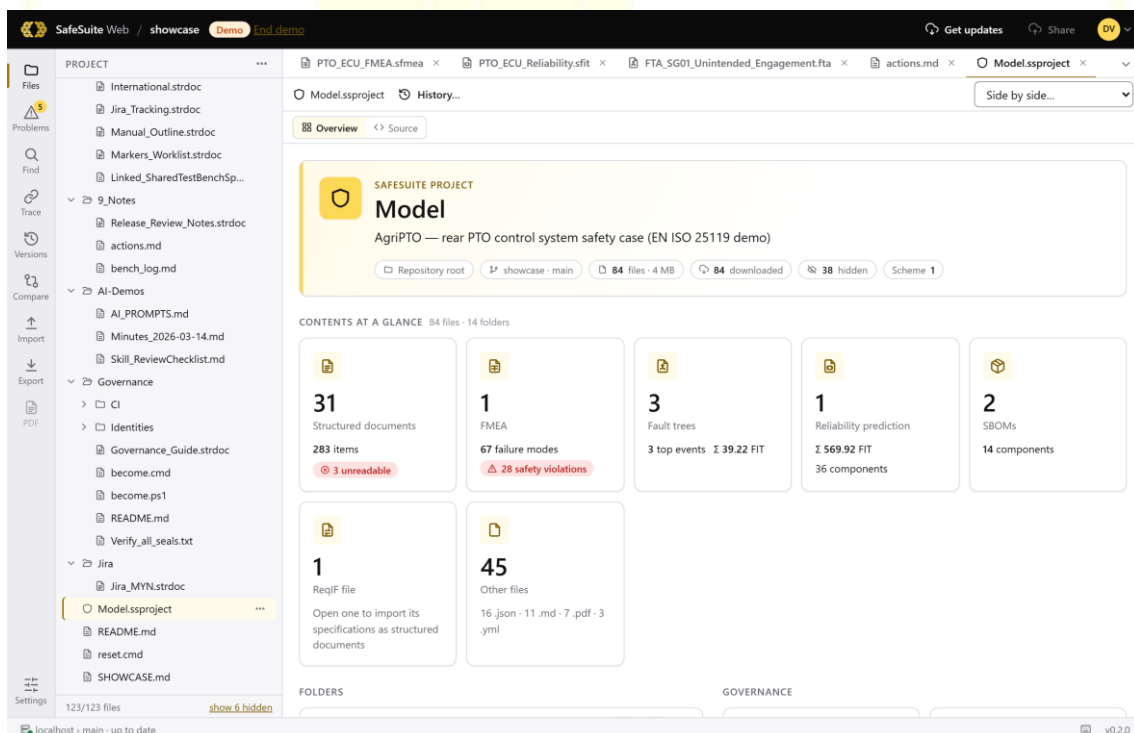
Every file of the project opens in the workspace, in an editor or a viewer that fits its kind.

File	Opens as
.strdoc	The structured document editor (chapter 8)
.sfmea, .sfit, .fta	The analysis editors (chapter 9)
.ssproject	The project overview: Contents at a glance , Folders , Governance , and a Source view. The file is maintained by the desktop application and is read-only here
.reqif, .reqifz	The Import ReqIF view (chapter 14)
A CycloneDX SBOM (.cdx.json, bom.json, sbom*.json or .xml)	The Import as structured document view
Text files, Markdown	The text editor
Images	The image viewer: fitted to the pane, a click switches to actual size
PDF	A viewer with Open in a new tab and Save a copy
.safesuite records	Read-only: they are signed by the desktop application
Anything else	A viewer with Save a copy and a hint about the application that opens it



The text editor on a Markdown file

The text editor keeps the file's line endings, byte-order mark and Windows-1252 encoding on save, and shows them in its bar ("N lines · LF · UTF-8"). Markdown files have **Preview** and **Edit**. Files over 1 MiB, binary or UTF-16 files, and files marked UTF-8 that are not valid UTF-8 are shown but not edited, with **Download**. A large text file shows its first 1024 KiB with **Download the whole file**.



The project overview of the .ssproject file

Documents from other tools. A .docx or .xlsx is downloaded to be opened with an office application. A .schdoc schematic is read by the desktop application.

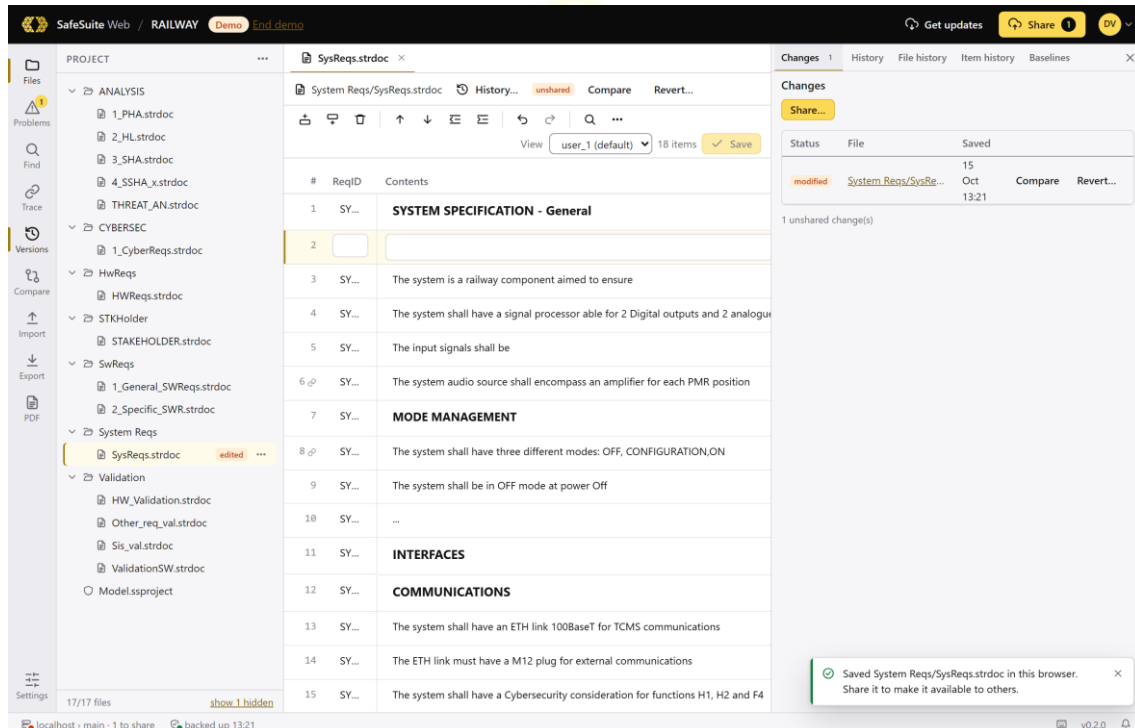


11 Versions, history and baselines

The Versions pane, opened with **Versions** in the rail or the **H** key, docks on the right of the editor. Its tabs are **Changes**, **History**, **File history**, **Item history**, **Baselines** and, with governance, **Activity log**. Close it with its **x** or the rail; drag its edge to resize it. On a phone it is a full-screen sheet under **More**

- **Versions**.

11.1 Changes: what you have not shared

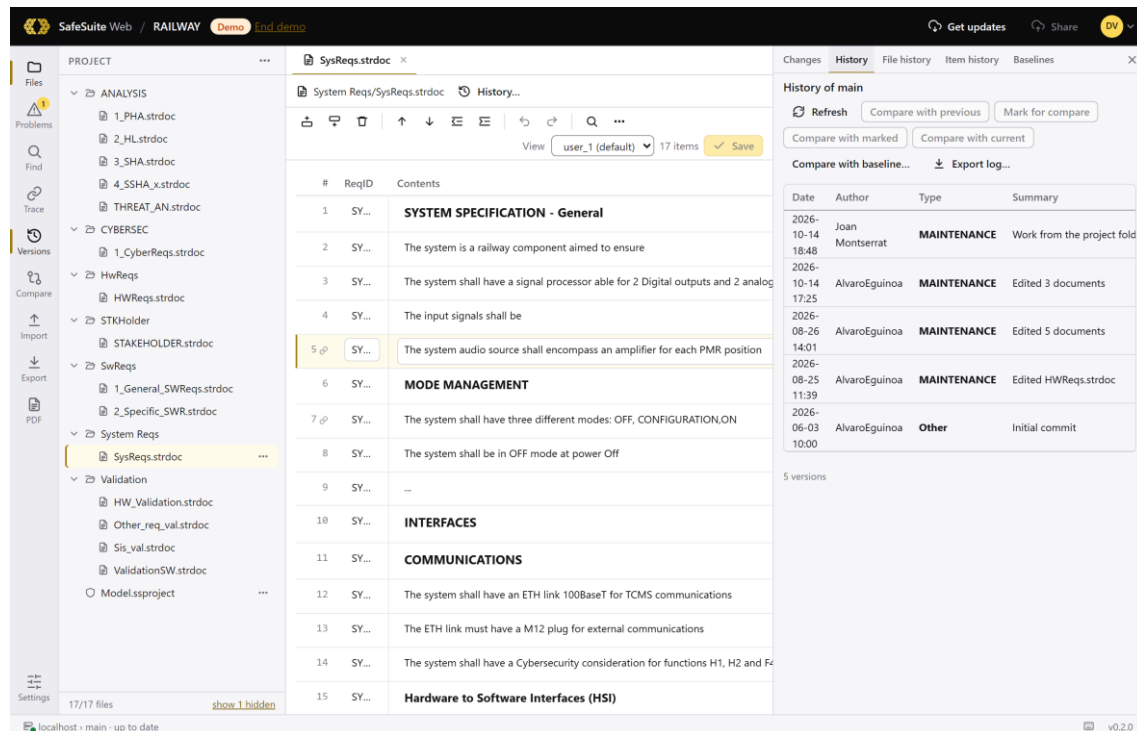


The screenshot shows the SafeSuite Web interface with the 'Versions' pane open on the right. The 'Changes' tab is selected, displaying a table of unshared changes for the file 'System Reqs/SysReqs.strdoc'. The table has columns for Status, File, Saved, Compare, and Revert... The changes listed include 'SYSTEM SPECIFICATION - General', 'MODE MANAGEMENT', 'INTERFACES', and 'COMMUNICATIONS'. A notification at the bottom right indicates that the file has been saved in the browser and can be shared.

The Changes tab

Changes lists your unshared changes: status (modified, added, deleted), file and the time saved, with **Compare** (your version against the shared one) and **Revert...** per row, and **Share...** at the top. With nothing unshared it reads "Nothing to share: every change in this browser is shared."

11.2 History: the versions of the branch



The screenshot shows the SafeSuite Web interface for a project named "RAILWAY". The "History" tab is selected, displaying a list of versions for the "SysReqs.strdoc" file. The list includes columns for Date, Author, Type, and Summary. The versions are listed in descending order of date, with the most recent version at the top. The interface also shows a "Compare with baseline..." section and an "Export log..." button.

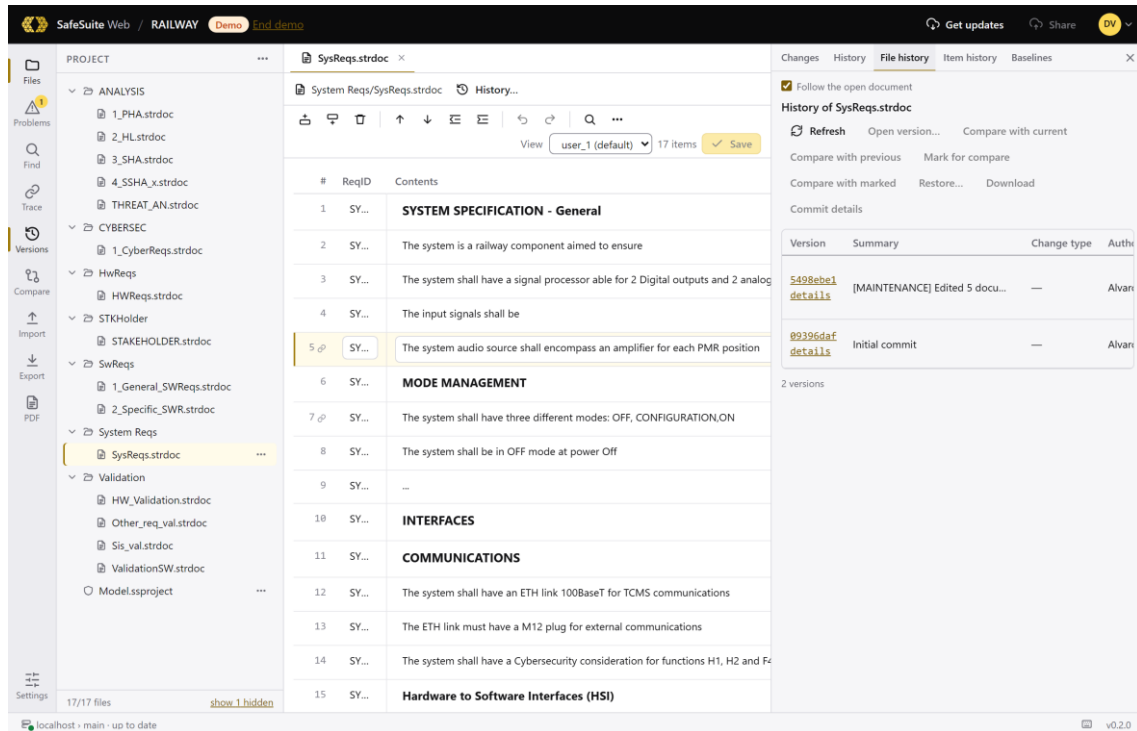
The History tab

History lists the versions of the branch, newest first: date, author, kind of change, summary, the number of files, the baseline or tag that names it, and the version code. Badges mark the version your copy is based on ("your copy"), the versions still to get ("to get") and the one marked for comparison. Selecting a version shows its message, parent, reference and the changed files, each with **Compare** and **Open file**.

Button	Action
Refresh	Reads the history again
Compare with previous	The selected version against the one before it
Mark for compare, Compare with marked	Any two versions
Compare with current	The selected version against your working copy
Compare with baseline...	Against a named baseline
Export log...	The history as a CSV file

A version-to-version comparison lists the changed documents, each with **Compare** and **Open**.

11.3 File history



The screenshot shows the SafeSuite Web interface. The left sidebar contains a file explorer with a tree view of the project structure. The main area displays the document 'SysReqs.strdoc' with a table of requirements. The right-hand pane is titled 'File history' and shows the history of the selected document. It includes a 'Follow the open document' checkbox, a 'Refresh' button, and a table of versions.

Version	Summary	Change type	Auth
5498e8e1	[MAINTENANCE] Edited 5 docu...	—	Alvar
00396daf	Initial commit	—	Alvar

The File history tab

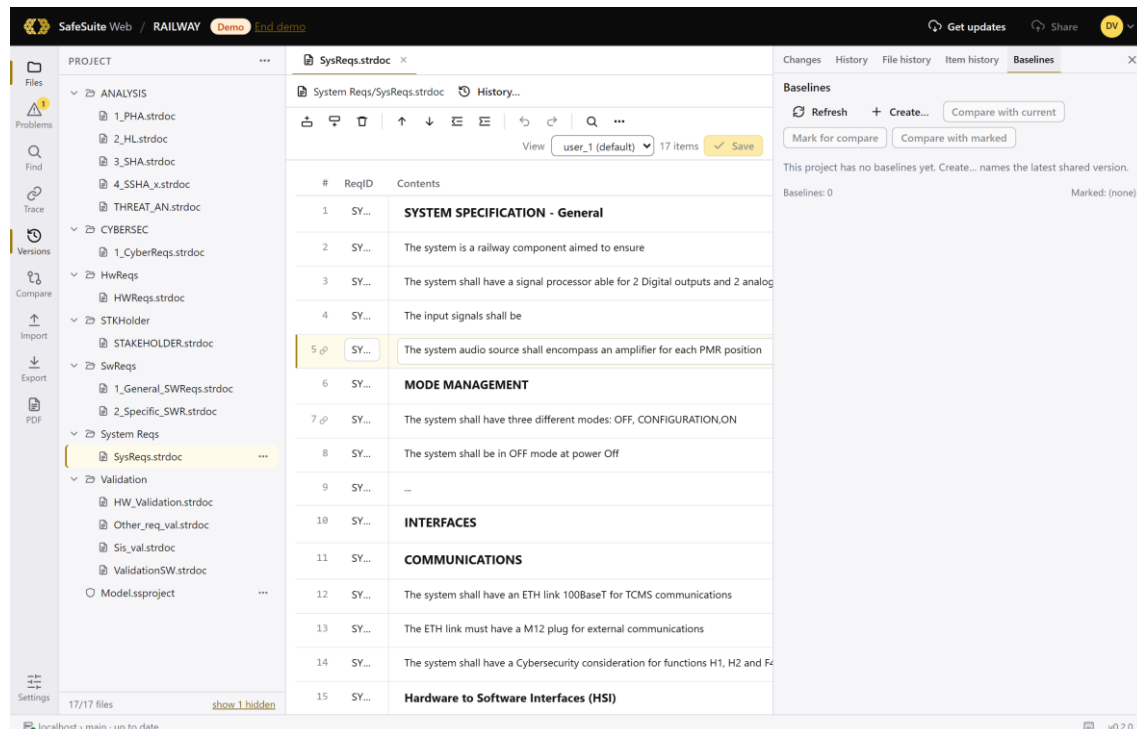
File history follows the open document (**Follow the open document**) or the file chosen with **History...** in the file header or the explorer. Each version shows its summary, kind of change, author and date. The toolbar and the row menu offer **Open version...** (the file at that version, read-only, with the same compare actions), **Compare with current**, **Compare with previous**, **Mark for compare**, **Compare with marked**, **Restore...**, **Download** and **Commit details**.

Restore... makes that version your unshared change: "Restored ... to version ... as your change in this browser. The version it replaced is kept in Backups in this browser." For a folder or the whole project (**Version history...** in the explorer), the restore dialog lists what is restored, what is deleted because it was added since, and what is left as it is.

11.4 Item history

Item history is the history of one row of a structured document: the versions that added, changed or removed it, with the old and new value of each attribute. It follows the selected row with **Follow the selected row**. Chapter 8 describes it.

11.5 Baselines



The Baselines tab

A baseline names a version so that the team can refer to it: "v1.0 — Type Approval". **Baselines** lists them with their name, status, scope, author, date, hash and version, and offers **Compare with current**, **Mark for compare** and **Compare with marked**.

Create... opens **New baseline**: a **Name** and an optional **Note**, then **Create baseline**. The baseline becomes a tag in the repository: "Baseline v1.0 created: version a3f9c21 is named by the tag v1.0 in the repository." A plain baseline names a version and nothing more; it has no seal to sign or verify.

With governance on (chapter 15), baselines are sealed and signed instead, through a wizard, and the tab adds **Sign...**, **Verify...**, **Evidence pack...** and **Governance...**. In a project that has a governance policy, creating a baseline needs governance on.

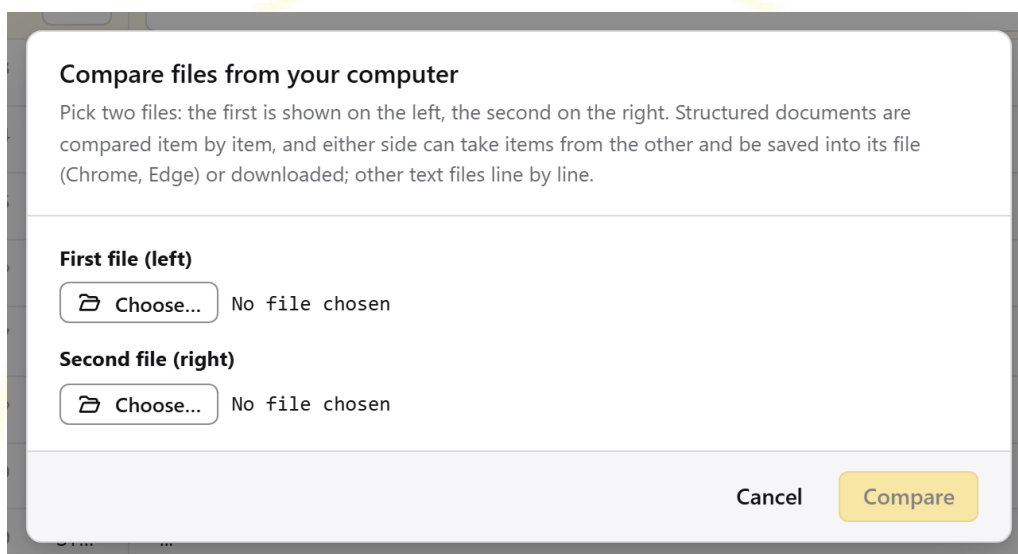
11.6 Compare

Comparisons open as a view with the two sides:

- **Structured documents** are compared item by item: the **Only changes** filter hides the unchanged rows; items and attributes can be copied across with **Alt+←** and **Alt+→**, and one-sided items dropped; **Save left** / **Save right** or **Download left** / **Download right** keep the result.
- **Text files** show a line diff with **Show N unchanged lines**.
- **Binary files** offer the download of each side.
- A comparison of files over 1 MB asks first: "Large file ... Loading may be slow. Continue?"

Where to start a comparison:

From	What is compared
Compare in the rail	Two files from your computer: First file (left) , Second file (right) , Compare
The explorer's Compare with...	The file against another project file of the same type, or a file from your computer
The explorer's Compare with shared version , or the header's Compare on an unshared file	Your unshared change against the shared file
The Versions pane	Versions, baselines and the working copy, as described above



Compare files from your computer

11.7 Change logs

The explorer's **Export change log...** (on the project or a folder) downloads a CSV of the history with a progress dialog; **History** ► **Export log...** does the same for the branch.

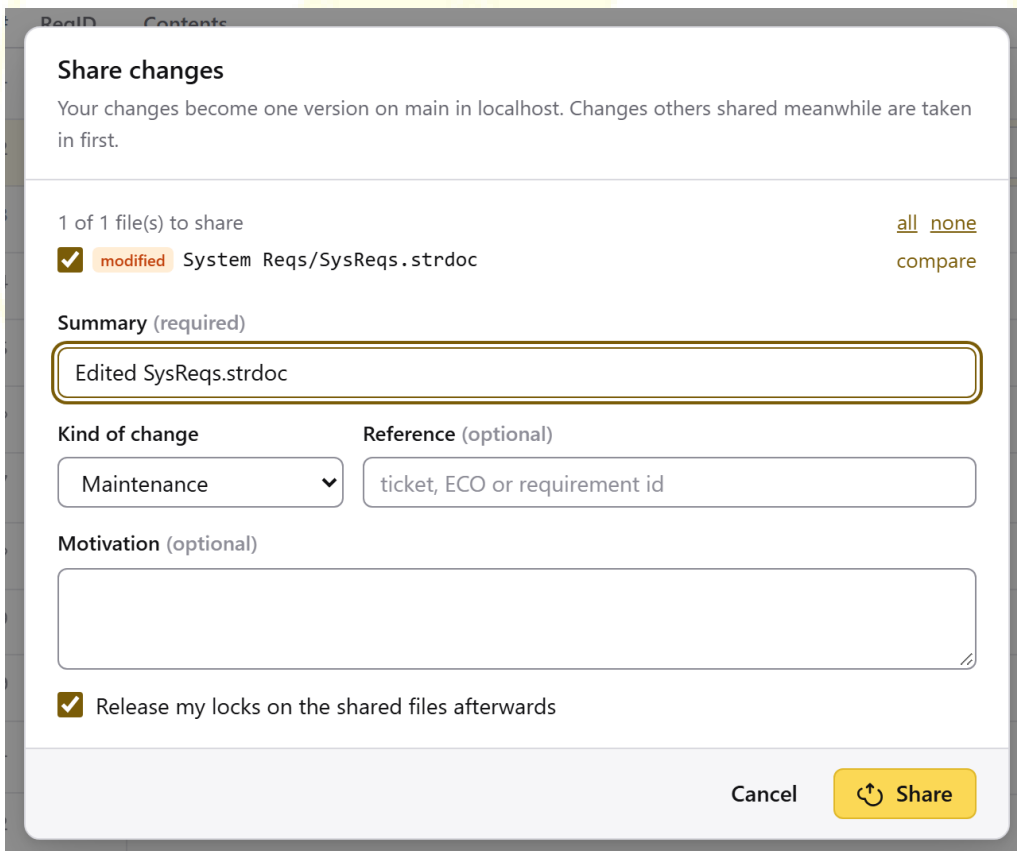
12 Working with the team

This chapter describes sharing and getting updates in detail, what happens when two people change the same thing, the reservations of files that cannot be merged, and the safety nets.

12.1 Share

Share sends your saved changes to the repository as one new version.

1. Press **Share** in the top bar (or **Share changes...** in the explorer, which ticks only that folder's files, or **Share...** in the Versions pane).
2. If a document has unsaved edits, **Unsaved modifications** asks **Save**, **Don't save** or **Cancel**; "Don't save" shares the saved files and keeps your edits open.
3. In **Share changes**, keep ticked the files to share. Each row shows its kind (modified, added, deleted), a badge when someone else is editing it, and **compare**.
4. Type the **Summary** (up to 120 characters; it is pre-filled with "Edited ..." until you type), choose the **Kind of change** (Feature, Fix, Refactor, Docs, Breaking or Maintenance), and add a **Reference** (a ticket, an ECO or a requirement id) and a **Motivation** if useful.
5. Leave **Release my locks on the shared files afterwards** ticked unless you keep working on a reserved file.
6. Press **Share** (Enter also shares).



Share changes

Your changes become one version on main in localhost. Changes others shared meanwhile are taken in first.

1 of 1 file(s) to share [all](#) [none](#)

☒ modified System Reqs/SysReqs.strdoc	compare
--	-------------------------

Summary (required)

Edited SysReqs.strdoc

Kind of change **Reference (optional)**

Maintenance ticket, ECO or requirement id

Motivation (optional)

☒ Release my locks on the shared files afterwards

Cancel Share

The Share dialog

The dialog says what happens: "Your changes become one version on main in GitLab.com. Changes others shared meanwhile are taken in first." The outcome names the version: "Shared 3 file(s) as version a3f9c21.", followed when relevant by "Got updates first: 2 file(s) others changed were taken in.", "Merged automatically: ..." and "Your replaced version(s) were backed up in this browser (N)." with **Show backups**.

Sharing needs you to be signed in to the repository: "Sign in to share your changes: a shared version records who made it." A refusal by the repository (a protected branch, for example) is shown in the dialog: "The repository refused the change: ...". If the repository keeps changing while you share, the application retries three times and then asks you to try again; nothing of yours is lost.

In a governed project with governance off, the dialog becomes **Send for review** (chapter 15).

12.2 Get updates

Get updates brings what others shared into your working copy. It never creates a version; it only updates your copy, merging into the files you also changed where it can.

The message tells what happened: "Already up to date.", or "Got 2 update(s), now at b17e4d0: 5 file(s) taken from the repository.", with "Merged with your changes: ...", "Your earlier version(s) were backed up in this browser (N)." and, when needed, "N file(s) changed on both sides need your decision."

Open documents without edits reload by themselves. A document you are editing keeps your edits and warns: "... changed in the repository while you were editing; your unsaved edits are kept. Saving them replaces what arrived: compare with the shared version first (History)."

12.3 How merging works

When you share or get updates, every changed file is looked at in three versions: the base (the version your copy came from), yours, and the repository's.

- A file changed on one side only is simply taken.
- A **structured document** changed on both sides is merged item by item, matched by item identifier, and field by field. Only a field changed on both sides, or an item changed on one side and removed on the other, is asked about.
- A **text file** changed on both sides is merged line by line, as Git does; only overlapping lines are asked about.
- **FMEAs, FIT predictions, fault trees**, pictures, PDFs, archives and office files are never merged by lines; they are asked about as a whole.
- Your version is always backed up in this browser before a merge replaces it.

12.4 Differences to resolve

When the two sides changed the same thing, the dialog **Differences to resolve** opens, from Share ("These files were changed both here and in the repository since you last got updates. Choose what to keep for each; nothing is lost silently.") or from Get updates. Each file explains itself: "You and

Elena Vidal both changed this file.", "You deleted this file; Elena Vidal changed it.", "You changed this file; Elena Vidal deleted it.", "You and Elena Vidal both created a file with this name."

For a **structured document** changed on both sides, the dialog opens on **Merge both, decide field by field**: one card per disputed field, with **Keep mine**, **Take theirs** or **Edit (Use this instead)**. An item changed by you and removed by them asks **Keep the item** or **Remove the item**; one removed by you and changed by them asks **Leave it out** or **Bring it back**. The footer counts "N of M decisions made" and the fields combined automatically. **Merge both, prefer mine where we overlap** and **Merge both, prefer theirs where we overlap** decide everything at once. (There is no "keep both" for documents: a copy would duplicate the identifiers.)

For a **text file**, **Merge both, decide where we overlap** offers, per place, **Keep mine**, **Take theirs**, **Keep both** or **Edit**.

For **any file**, three whole-file choices: **Keep both** (theirs stays at its path; yours is kept next to it as "Name (conflict login 20261015-1042).ext"), **Keep mine** (or **Keep my deletion**), and **Keep theirs** (or **Accept their deletion**), which backs your version up first. **See both versions...** opens a side-by-side comparison.

Decide later leaves the files as they are; **Resolve** (or **Resolve and share**) applies your decisions. Nothing is shared while a decision is pending.

12.5 Decisions left for later

Undecided files come back at the next **Get updates** or **Share**. Meanwhile **Get updates** shows their count ("N file(s) changed on both sides wait for your decision"), the status bar reads "N decisions", and the **Team server** panel offers **Resolve decisions....**

12.6 Files held back

Get updates does not write a conflicted file that has unsaved edits, nor a file that cannot be merged while someone else has reserved it. The message says so: "... was not merged: it has unsaved edits; save them, then get updates again." or "... Elena Vidal has reserved it." The file waits for a later **Get updates** or a **Share**.

12.7 Who is editing what

Files open editable; there is no reserve button. What you edit is announced to the team in the background:

- **Presence.** Your first edit of any file tells the others "... is editing since 10:42". It lasts 15 minutes and is renewed while you keep editing; it is removed when you share, revert, close the tab or leave the project. Presence informs; it never blocks.
- **Reservations.** Your first edit of an FMEA, a FIT prediction or a fault tree reserves the file in the repository, for 14 days, renewed while you work. The desktop application honours it. If someone took it a moment earlier, the application tells you at once.

What you see when a colleague is at work on a file:

Situation	What the file shows
Someone edits a file that can be merged	A chip "Elena Vidal is editing since 10:42" (" in the desktop app" when so), a one-time message, and a grey pencil in the explorer. You can edit it too: your changes and theirs are merged when you share
Someone holds a reservation on a file that can be merged	A lock "Locked by Elena Vidal". It informs, and it blocks rename, move and delete
Someone holds a reservation on a file that cannot be merged	The file is read-only: "Read-only: Elena Vidal is editing this file since 10:42. FMEAs cannot be merged, so it stays read-only for you while they hold its reservation (until 23 Oct at the latest)." with Break lock...
Someone edits a file that cannot be merged without a reservation (offline, for instance)	The same warning, with Edit anyway

Your own reservation shows as **Locked by you · 13 d left** with **Release**; the status bar's lock item lists **Reserved by you**. Share releases the reservations of the shared files (unless you untick it), and so do Revert and leaving the file with nothing of yours left in it.

Break lock... asks for a reason ("e.g. on leave until 20 Oct; agreed by phone"), which is recorded in the lock history, and warns that the other person's unshared work could end up in conflict. A repository may limit who can break locks; an expired lock can be taken by anyone.

12.8 Backups in this browser, Revert and restore

Before anything replaces a version of yours, that version is copied aside in the working copy: a merge by Get updates or Share, **Keep theirs**, **Revert**, a restore from a backup or from the history, Check & Repair, the deletion of an unshared file, and an upload that replaces a file. Backups are never pruned on their own.

Project actions ▶ **Backups in this browser...** (also the status bar's **Local backups...**, the **Show backups** button of messages, and **Backups of this file...** in Check & Repair) lists them, grouped by time, each with its reason ("Replaced by Get updates: merged with changes from the repository", "Discarded by Revert", "Deleted in the project panel"...) and the actions **Open**, **Compare with current**, **Download**, **Restore...** and **Delete...**; **Delete all...** empties the list. A restored backup becomes your unshared change, and what it replaces is backed up first.

Revert... (in the file header's unshared chip, the Versions pane's **Changes** and the explorer) asks "Revert ... to the shared version?" or "Remove ... from this browser?" for a new file, backs your version up, and warns that unsaved edits of it are lost too.

12.9 The backup branch

While you are signed in, your unshared changes are also copied to the repository in the background after every change, to a branch of your own, `safesuite-wip/<you>/<branch>`. It holds one version with all your unshared changes, replaced at each copy, and is deleted once nothing is unshared. It protects you against a browser that loses its storage, a lost profile or a new computer: a new checkout offers **Restore it into this checkout**. Anyone with read access to the repository can see the branch; it is never merged into the team's branch.

The status bar's safety item reports it: "backed up 10:42", "backup failed", "backup paused" (an older backup that this copy did not make is kept until you restore or discard it), "not protected" (signed out or offline) or "in this browser".

[!The status bar after a save: one change to share, and the backup of the unshared work](#) Its panel, **Your unshared work**, offers **Share...**, **Restore it**, **Discard it...** and **Local backups...** When a backup from another copy of yours exists, the panel says "Unshared work from another copy is backed up." and lists the files; **Restore it** is offered while you have no changes of your own.

12.10 Offline or signed out

Without a network, or signed out of the repository, you can still open the project, edit and save, use the explorer, compare, and the backups. The status bar reads "not reachable"; **Check server** says "Sign in to check the server." or "Could not reach the server"; sharing and getting updates fail with a clear message. Nobody sees you editing meanwhile, and the files that cannot be merged warn: "Signed out or offline: whether someone else is editing this file cannot be checked, and nobody can see that you are." Your reservations from the last contact stay yours. When the application learns who you are again, it records your presence and reserves the files you edited in the meantime.

A folder repository asks for its permission again after a reload: "The browser asks again whether SafeSuite may open the folder ... Allow access to see what others shared and to share." with **Allow access**.

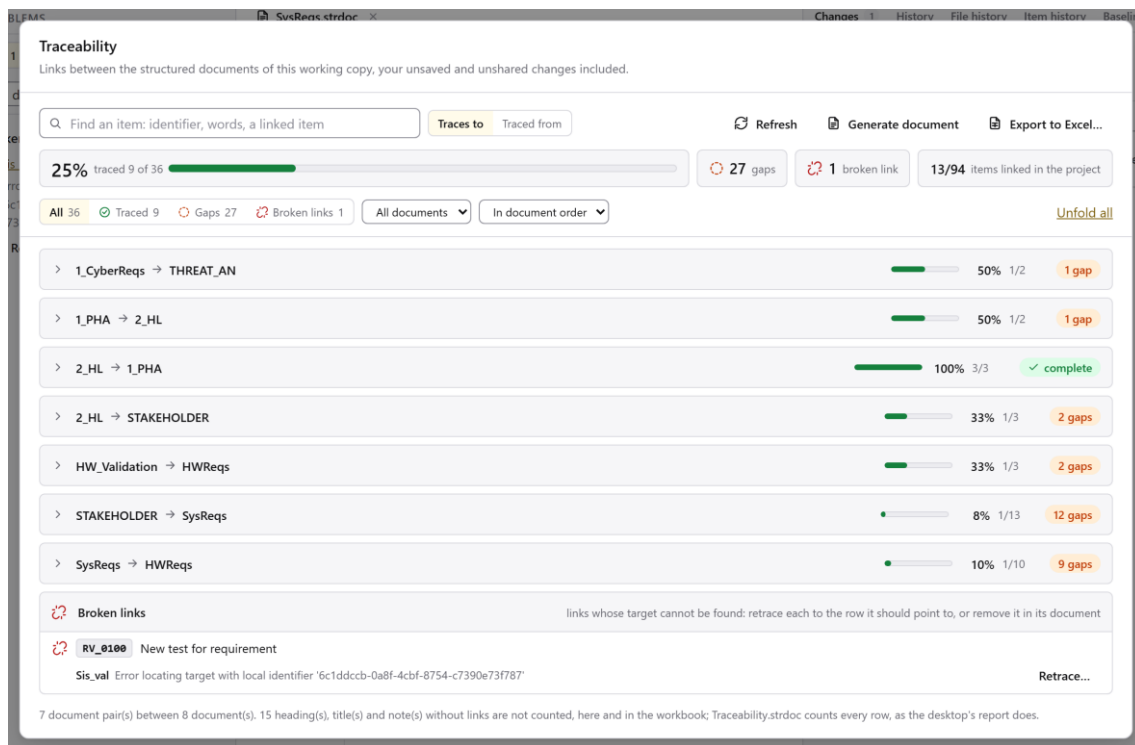
12.11 Check server and updates waiting

Check server, in the **Team server** panel (and on the phone's Home), asks the repository for the latest version, the reservations and the presence, without changing anything: "Checked the server: 2 version(s) to get." When the repository is ahead of your base, **Get updates** shows a gold dot, the status reads "2 to get · 1 to share", and the panel says "Team updates available." The latest version is also read when the project opens and at every Get updates and Share.

13 Traceability and problems

13.1 Traceability

Trace in the rail (key **T**) opens the **Traceability** dialog: the links between the structured documents of the project, how much of each document is covered, and where the gaps are.



The Traceability dialog

- The figures at the top: "% traced (x of y)", the gaps or orphans, the broken links, and "x/y items linked in the project".
- **Direction:** **Traces to** or **Traced from**.
- The search box finds an item by identifier, words or a linked item.
- Filters: **All**, **Traced**, **Gaps**, **Broken links**; by document; by link type; the order (**In document order**, **Gaps first**, **By identifier**); **Clear filters**; **Fold all** and **Unfold all**.
- Each pair of documents shows a coverage meter and its gaps, or "complete".
- **Broken links** (links whose target no longer exists) offer **Retrace...**: type the identifier or the text of the row the link should point to, then **Point the link here**.
- Clicking an item opens its document at that row and hides the dialog; **Trace** brings it back as you left it.

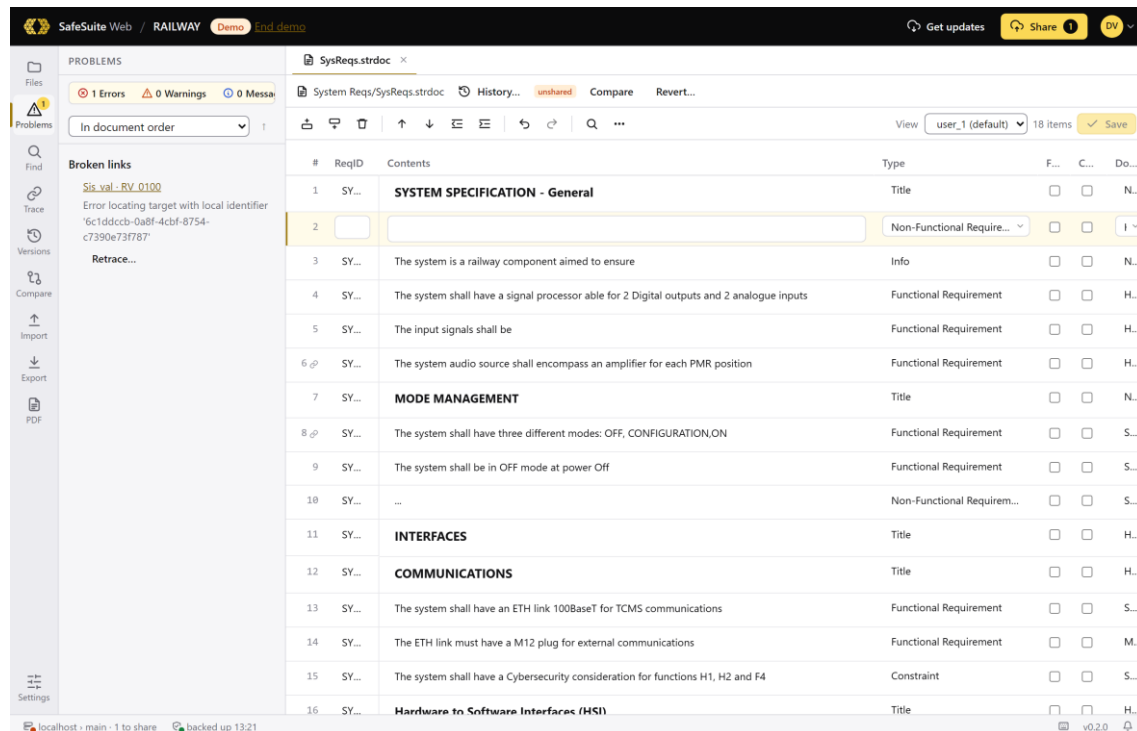
Two deliverables come from the same dialog:

- **Generate document** writes the traceability matrix as a regular structured document, `Traceability.strdoc`, at the project root, with computed coverage columns that stay up to date.
- **Export to Excel...** opens the Export dialog on the **Traceability workbook**.

Documents that are not downloaded are counted with **Download and analyse them**. A project without links says "No links between documents yet".

13.2 Problems

Problems in the rail (key **P**) lists, for the whole project, what needs attention:



The Problems panel

Section	What it lists
Cannot be read	Documents that fail to open; each opens Check & Repair
Identifier not unique in the project	The same identifier in two documents
Repeated automatic identifiers	Automatic identifiers that collide
Invalid values	A number outside its limits, a selection out of range, a bad date
Configuration	Problems in a document's configuration
Broken links	Links to rows that no longer exist, each with Retrace...
Errors and warnings	The markers WARNING and ERROR written in the text
To do	The markers TODO, TBD and FIXME

Toggles **Errors**, **Warnings** and **Messages** narrow the list; a sort list orders it in document order, by marker, by document, by item or by text. Each entry opens the document at the row. The badge on the rail counts the problems; the footer notes the documents not downloaded, which are not checked.

Tip: Review markers are a lightweight way to hand work around: write "TODO check the limit with HW" in a requirement, and the Problems panel lists it for everyone until it is removed.

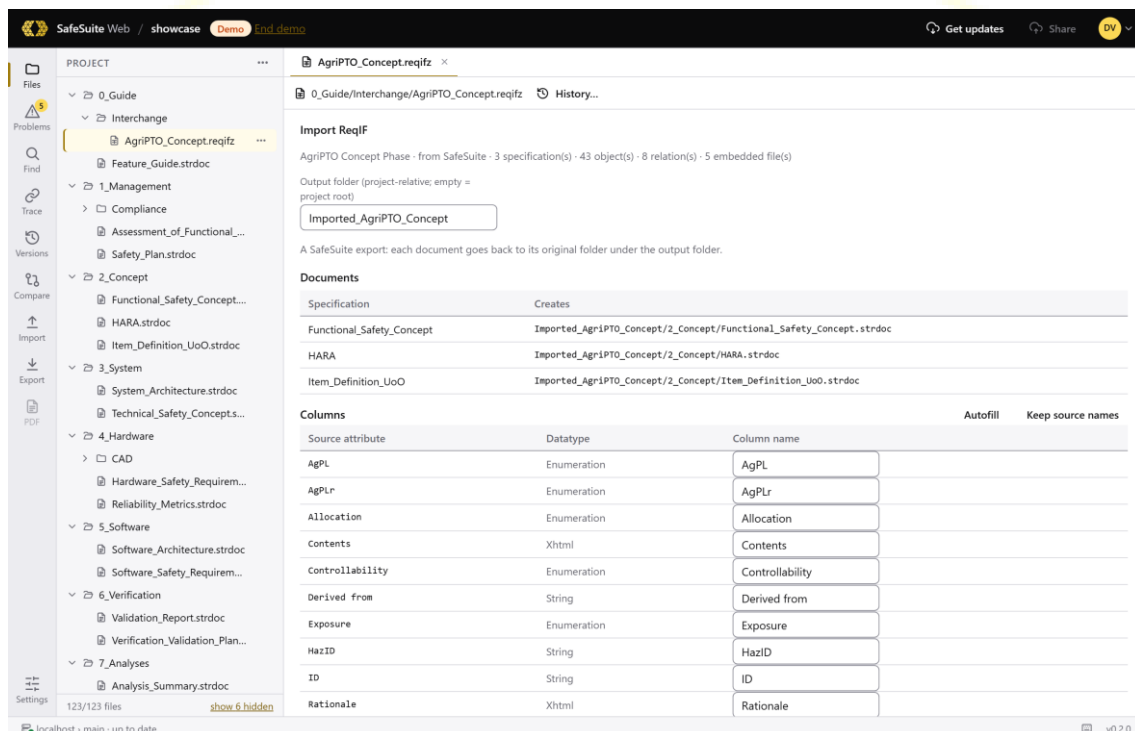


14 Import, export and PDF

14.1 Importing a ReqIF file

ReqIF is the standard the industry uses to exchange requirements between tools such as DOORS and Polarion. The import needs the ReqIF Import and Export licence.

1. **Import** ► **ReqIF exchange file...** and choose a .reqif or .reqifz file (up to 200 MB); or open a ReqIF file already in the project.
2. In the **Import ReqIF** view, set the **Output folder** (project-relative; empty for the project root), check the **Documents** the file creates, one per specification, and the **Columns** with their types (**Autofill** maps them for you; **Keep source names** keeps the original names).
3. Press **Import**. The new documents appear in the explorer as unshared changes. A log lists every decision the importer took; **Copy log** copies it.



Import ReqIF

AgriPTO Concept Phase - from SafeSuite - 3 specification(s) - 43 object(s) - 8 relation(s) - 5 embedded file(s)

Output folder (project-relative; empty = project root)

Imported_AgriPTO_Concept

A SafeSuite export: each document goes back to its original folder under the output folder.

Documents

Specification	Creates
Functional_Safety_Concept	Imported_AgriPTO_Concept/2_Concept/Functional_Safety_Concept.strdoc
HARA	Imported_AgriPTO_Concept/2_Concept/HARA.strdoc
Item_Definition_UoO	Imported_AgriPTO_Concept/2_Concept/Item_Definition_UoO.strdoc

Columns

Source attribute	Datatype	Column name	Autofill	Keep source names
AgPL	Enumeration	AgPL		
AgPLr	Enumeration	AgPLr		
Allocation	Enumeration	Allocation		
Contents	Xhtml	Contents		
Controllability	Enumeration	Controllability		
Derived from	String	Derived from		
Exposure	Enumeration	Exposure		
HazID	String	HazID		
ID	String	ID		
Rationale	Xhtml	Rationale		

The Import ReqIF view

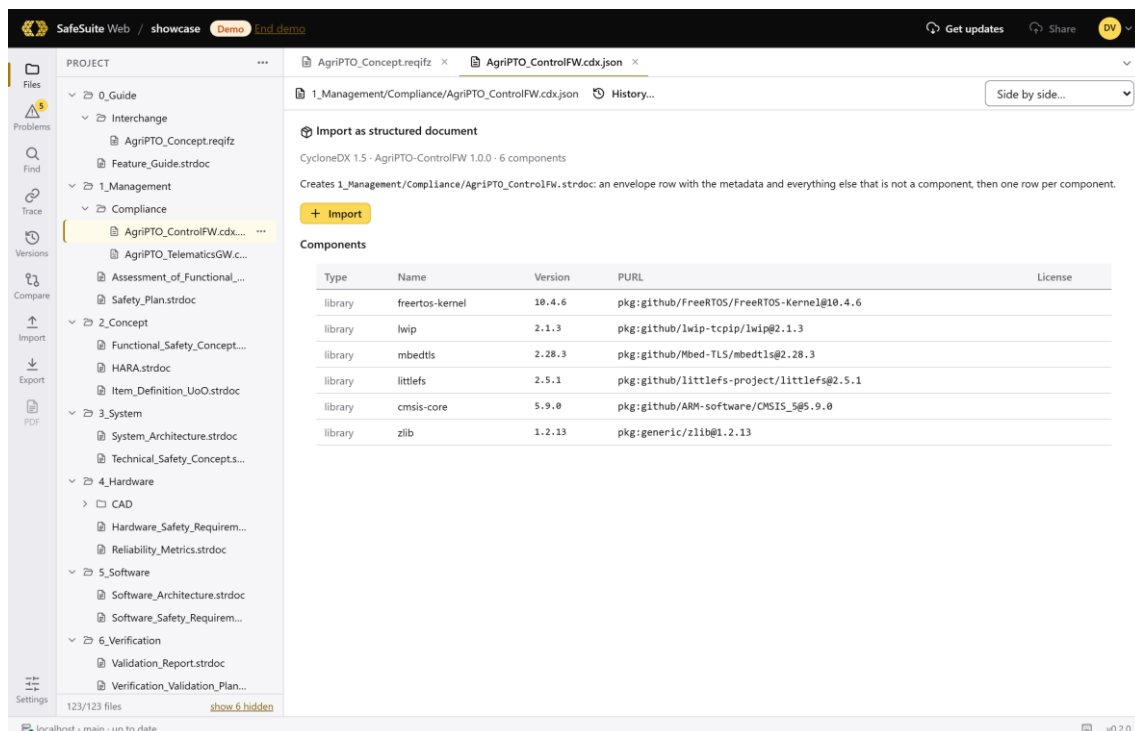
14.2 Importing Excel or CSV

Import ► **Excel or CSV...** turns a spreadsheet or a delimited text file (.xlsx, .csv, .tsv, .txt, up to 100 MB) into a new structured document:

1. Choose the file. Pick the **Worksheet**, the **Separator**, the **Encoding** and the **Decimal separator** when they are not detected.
2. Tick **The first row holds the column names** if so.
3. Give each column a type (Plain text, Rich text, Integer, Real, Date, Boolean, Selection, or **Not imported**) and check the **Preview**.
4. Choose the **Folder** and the **Document name**, and press **Import**.

14.3 Importing an SBOM

A CycloneDX software bill of materials (.cdx.json, bom.json, or an sbom file in JSON or XML) opens in the **Import as structured document** view, which turns its components into a structured document. The document's Export then offers **CycloneDX SBOM** to write it back.

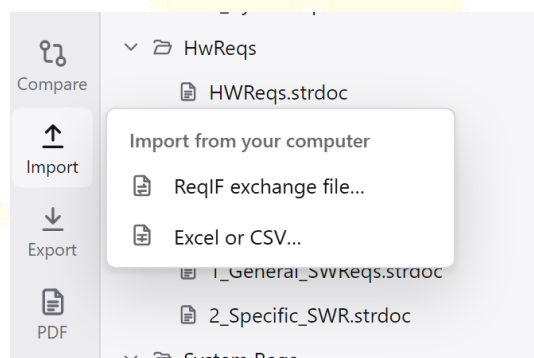


Type	Name	Version	PURL	License
library	freertos-kernel	10.4.6	pkg:github/FreRTOS/FreeRTOS-Kernel@10.4.6	
library	lwip	2.1.3	pkg:github/lwip-tcpip/lwip@2.1.3	
library	mbedtls	2.28.3	pkg:github/Mbed-TLS/mbedtls@2.28.3	
library	littlefs	2.5.1	pkg:github/littlefs-project/littlefs@2.5.1	
library	cmsis-core	5.9.0	pkg:github/ARM-software/CMSIS_5@5.9.0	
library	zlib	1.2.13	pkg:generic/zlib@1.2.13	

The SBOM import view

14.4 Imports into the open file

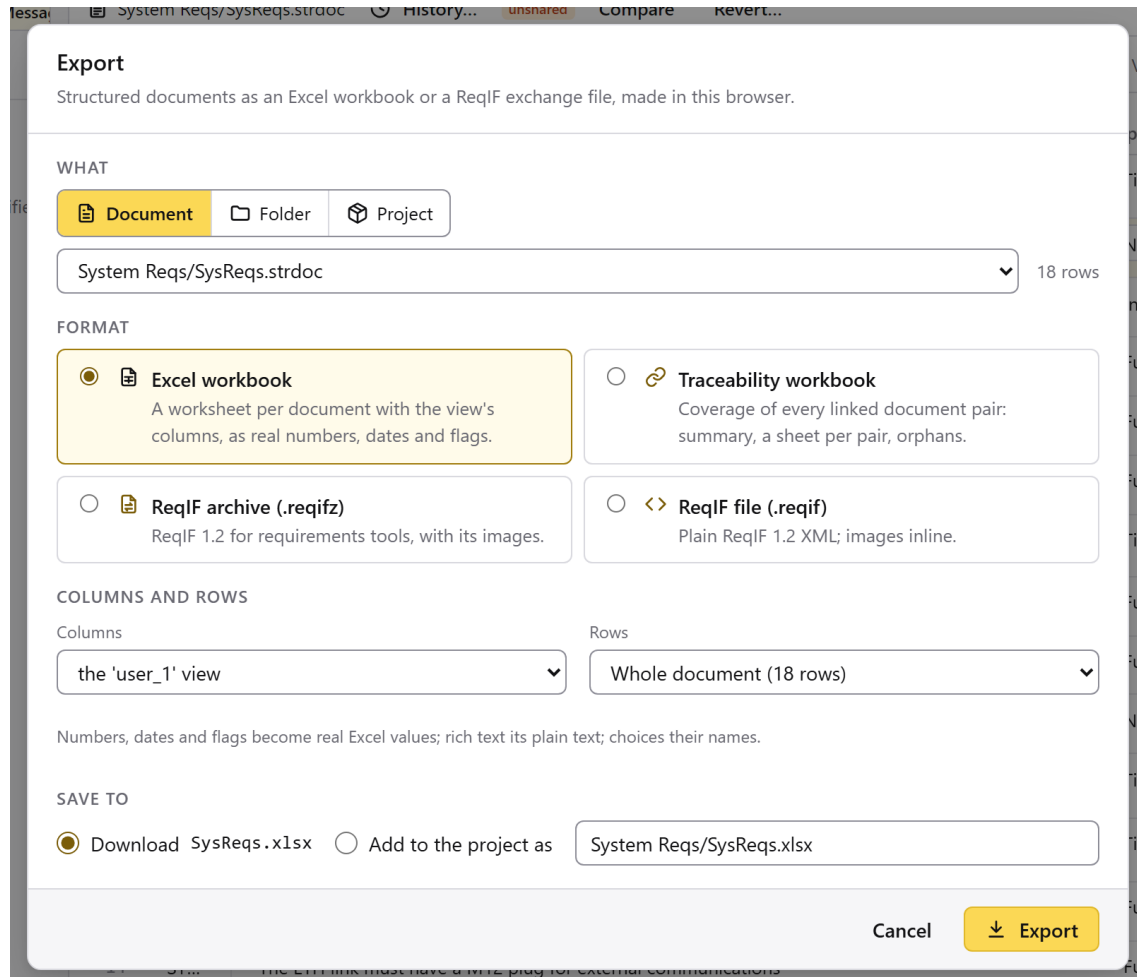
When an FMEA, a FIT prediction or a fault tree is open, **Import** adds a heading **Into <file>** with the imports of that editor (chapter 9): CSV items, functions, diagnostics, safety mechanisms and goals for an FMEA; a SafeFit CSV for a FIT prediction; FMEA items for a fault tree.



The Import menu

14.5 The Export dialog

Export in the rail (key **E**) runs the open file's only export directly, or opens a menu: the open file's formats, then **Structured documents** ▶ **Excel, ReqIF or traceability...**, which opens the **Export** dialog:



The Export dialog

Setting	Choices
What	Document, Folder or Project
Format	Excel workbook, Traceability workbook, ReqIF archive (.reqifz), ReqIF file (.reqif) (the two ReqIF formats need the licence)
Columns (one document, Excel)	The columns on screen, a saved view, or every attribute
Rows (one document, Excel)	The whole document, the filtered rows, or the selected rows
Worksheets (folder or project, Excel)	A view per document, or Not exported

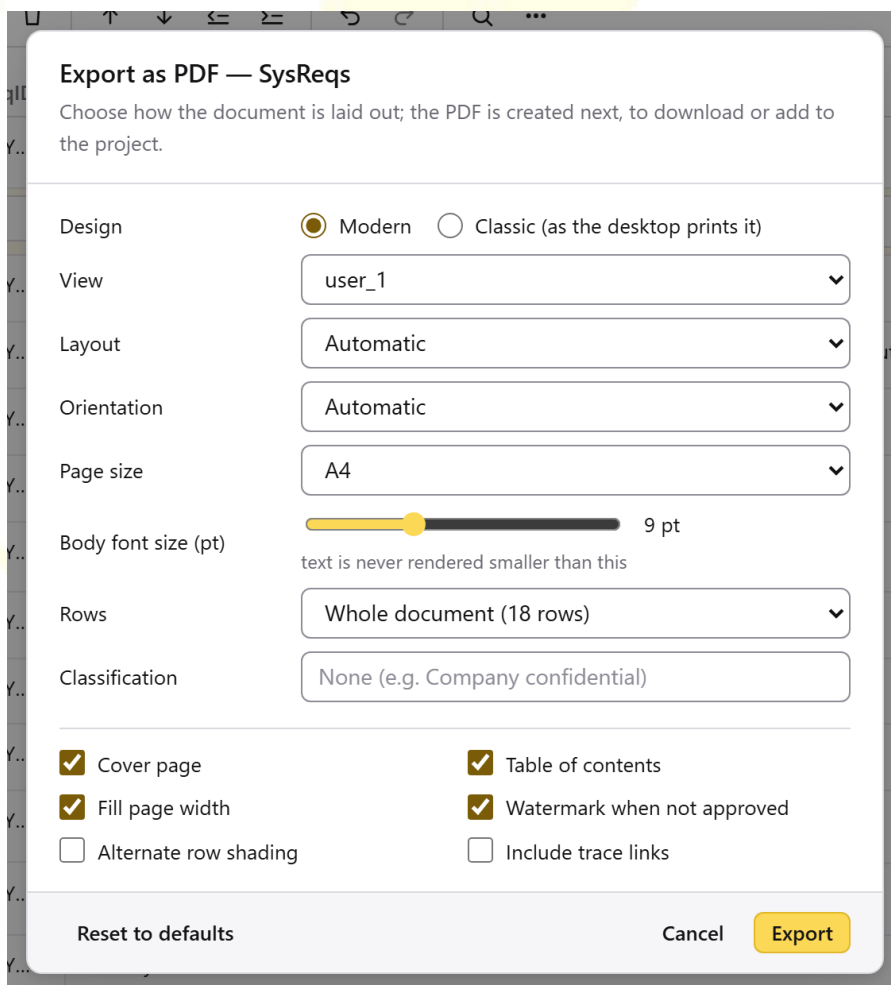
Setting	Choices
Title (ReqIF)	The exchange file's title
Save to	Download <file> , or Add to the project as <path>

The explorer's **Export ReqIF...** and **Export as Excel...** open the same dialog for a document, a folder or the project. Documents not downloaded are skipped and listed, with **Download them**; the result has **Copy log**.

Other exports: **SafeFMEA CSV**, **SafeFit CSV...**, a fault tree's **Diagram as SVG**, **Diagram as PNG** and **Trees, events and cut sets as Excel**, and an SBOM document's **CycloneDX SBOM**.

14.6 PDF

PDF in the rail makes the open document's PDF. For a structured document it opens **Export as PDF**:



Export as PDF — SysReqs

Choose how the document is laid out; the PDF is created next, to download or add to the project.

Design ☒ Modern ☐ Classic (as the desktop prints it)

View

Layout

Orientation

Page size

Body font size (pt) 9 pt
text is never rendered smaller than this

Rows

Classification

☒ Cover page ☒ Table of contents

☒ Fill page width ☒ Watermark when not approved

☐ Alternate row shading ☐ Include trace links

The PDF options of a structured document

Setting	Choices
Design	Modern , or Classic (as the desktop prints it)
View	Current layout or a saved view
Layout	Automatic, Table or Report
Orientation, Page size, Body font size	Portrait or landscape; A4 or Letter; 8 to 11 points
Rows	All, filtered or selected
Classification (Modern)	A text printed on every page, for example "Company confidential"
Checkboxes	Cover page , Table of contents , Fill page width , Watermark when not approved , Alternate row shading , Include trace links

Reset to defaults restores the options. **Export** makes the PDF and shows "PDF ready." with **Download**, **Print** and **Add to the project as <name>.pdf**. The Modern cover records the document, the view, the rows, the lifecycle state and its approvals, the baseline, the version ("main @ a3f9c21", plus "+ local changes" when the file differs from the shared version), who exported it and the classification. A document with a lifecycle record that is not approved carries a DRAFT or IN REVIEW watermark when the option is on. Every PDF made in a demo carries DEMO across its pages.

For an FMEA, a FIT prediction or a fault tree, **PDF** opens the report dialog of that editor (chapter 9).

14.7 Printing

Print... (Ctrl+P) in a structured document makes the PDF with the same options and opens it in the browser's PDF viewer to print. A fault tree prints its diagram with the diagram's **Print**. Any other file prints as the page.

14.8 Files to and from your computer

- **Save to computer** in a file's menu downloads it; on a folder or the project, **Save to computer as .zip** downloads an archive (files not downloaded are left out, and the message says so).
- **Upload a new version...** replaces a file with one from your computer; the replaced version is backed up in this browser.
- **Upload files here...** adds files or a folder (chapter 7).
- **Compare** compares two files from your computer without adding them to the project; a file picked there can be saved back in place in Chrome and Edge.

15 Governance

Governance is the part of SafeSuite that turns a project's history into evidence: who reviewed and approved a document, which versions were sealed as baselines, who may do what, and a tamper-evident log of it all. It is optional, and off by default. Teams that do not need it work with links, versions, history and compare, and nothing of governance is shown or recorded. This chapter describes governance in the web application; the full toolset, including signing at the higher levels, is in the desktop application.

15.1 Switching governance on

Settings ► **Experimental** ► **Governance**. The switch is kept per browser. It is on by default only in a browser that already used governance (one that holds a signing key); everywhere else it is off. Acting in governance also needs the Governance licence (chapter 4); reading never does.

With governance on, the workspace shows:

- **Govern** (key **G**) and **Reviews** (key **R**) in the tool rail, the second with the count of records waiting for you;
- **Identity and signing key...** in the account menu, and an identity item in the status bar;
- a lifecycle chip on each structured document (**Draft**, **In review**, **Approved**), and a signature page at its foot;
- the **Activity log** tab of the Versions pane, and the seal actions in **Baselines**;
- **Create baseline from folder...**, **Show effective policy here...** and **Show change log** in the explorer's menus.

Turning it off takes effect at once. Any governance action attempted while it is off is refused with "Governance is off in this browser: turn it on in Settings (Experimental) to review, approve, sign or change the policy."

Note: The governance records live in the project's `.safesuite` folder, hidden in the explorer, in the same format the desktop application writes. The two applications read each other's records, and a record made in the web application is a change in this browser until you share it. Every governance outcome ends with "Recorded in this browser; share it to make it count for others."

15.2 Identity and signing key

Governance records say who did what, signed with your own key. The **Identity** dialog (account menu) has these sections:

- **Repository account:** whom the repository knows you as, with **Sign in** or **Sign out**, and the switch **Stay signed in on this device**.

- **In this project's records:** whom you act as, and what your signatures prove. A suggestion "Your account looks like Elena Vidal in these records." with **I am Elena Vidal** matches you to the people the records name; **Act as** lists them.
- **Signing keys in this browser: Create a signing key...** (your name, e-mail and a passphrase of at least 8 characters; the key file is downloaded, keep it safe), **Use a key file from this computer...** (the desktop's own key file works), or a key file committed in the project. Each key offers **download**, **forget**, **enrol now** and **enrol here....**
- **Session: Unlock** a session of 1, 4, 8 or 24 hours, so that your save receipts are signed without a prompt; **Lock now** ends it.
- **Your access in the open project: Delegate my access....**

The web application signs at **Level 0 (claimed)**, with your repository account only, or **Level 1 (local-passphrase)**, with a key file protected by your passphrase, as the desktop makes one. The higher levels (Windows account, TPM, hardware token, certificate) belong to the desktop application. A policy can demand a level: the refusal says which one you sign at, and offers a waiver. A key is enrolled in a project automatically at its first signature there.

15.3 The lifecycle of a document

The lifecycle chip in a structured document's header shows **Draft**, **In review**, **Approved** or "In review (1 of 2 signatures)". Its menu:

Action	What it asks	What happens
Send for review... (a draft)	"Send ... for review? It stays editable, but reviewers see it as awaiting their decision, and an edit sends it back to draft."	The document is In review ; reviewers find it under Reviews
Approve... (in review)	The Approve document dialog: what is signed, as whom, the policy's rule ("Requires 2 signatures. 0 of 2 collected."), your passphrase	Your signature is recorded; the document is Approved when enough signatures are collected
Reopen for editing...	"Reopen ... for editing? Any approval it already has stops applying."	A draft again

Before acting, the application checks the licence and the switch, who you are ("Sign in, or say who you are (Identity), first."), and the policy. A policy refusal explains itself and, where a waiver could help, offers **Request a waiver....**

Editing an approved document voids its approval: it returns to Draft, the save message says so, and the signature page at the foot of the document reads "Seal broken — Changed after signing — the signatures below no longer apply." An approval needs the document saved ("Save ... first: an approval signs the saved content."), and the policy may require that the approver is not the last editor.

The signature page lists each signature with its role, name, date, intent and method, under a badge **Not signed**, **Seal intact** or **Seal broken**. A row's history offers **Changes since last approved**.

15.4 Reviews

Reviews (key **R**) lists what waits for you: cards labelled **Approve**, **Sign baseline** or **Waiver request**, each with its date and the buttons **Open**, **Changes** (a comparison with the last approved version), **Approve...**, **Sign...** or **Decide...** (reason, passphrase, **Grant** or **Deny**). "Nothing waits for you" means just that. On a phone, a document waiting for your approval shows **Waiting for your approval** at its foot, with **Changes** and **Approve...**

Below the records, **Sent for review** lists the work branches of colleagues who work without governance (see below).

15.5 Sealed baselines

With governance on, **Create...** in the Versions pane's **Baselines** (or **Govern** ▶ **Records** ▶ **Create baseline...**, or the explorer's **Create baseline from folder...**) opens a wizard in five steps:

1. **What should this baseline cover?** The whole project, or a single folder.
2. **Name this baseline**, for example "v1.0 — Type Approval", with an optional note to the record.
3. **What will be sealed**: the count of files hashed into the seal.
4. **Sign this baseline**: tick **Record my signature on this baseline now**, state what you are signing for, enter your passphrase. A baseline can also be sealed now and signed later from the list.
5. **Seal**. The outcome: "Sealed 'v1.0 — Type Approval'. 123 files sealed and hashed · the whole project."

From the baseline list: **Sign...**, **Verify integrity...** (a report, with advice for every file that changed since the seal), **Export evidence pack...** (a zip with a README, a certificate PDF, the manifests, the activity log and the trust records, for an assessor), **Compare with current**, **Mark for compare** and **Compare with marked**. A seal that reports changes is not a failure: it tells you the documents moved on after the baseline was taken.

Note: The desktop application also makes a Git tag for every sealed baseline. The web application makes none, so a baseline sealed here is named in the records, not in the branch's tags, until the desktop adds it.

15.6 The Governance dialog

Govern (key **G**) opens the dialog with its tabs:

Tab	What it holds
Records	A summary ("Seals: 3 intact, 0 broken. Audit log: 41 entries, chain intact. Governed documents: 12."), then the seals with their state and advice,

Tab	What it holds
	the governed documents, the people and policy, the trust records, and the waivers and delegations
Waivers	Requests and decisions: Request a waiver... (capability, scope, reason, single use or until a date); holders of Grant waivers decide with Grant, Deny or Revoke
Delegations	Rights lent to someone for a time, with Delegate and Revoke... ; "Edit the policy" is never delegable
Policy	The policy editor: People (roles, capabilities, folder scope, validity, minimum level), Roles (Reader, Author, Reviewer, Approver, Auditor, Owner), Requirements (per capability: minimum level, number of signers, not the author, a required role, pending timeout; signed commits; the delegation limit), Keys (revoke, vet, make trust root) and History . Save and sign puts it in force
Effective policy	Who holds what at a place of the project, also from the explorer's Show effective policy here...
Pending signatures	Records waiting for more signatures, with their timeout, Withdraw... and Withdraw lapsed records
Activity log	The chained log, with filters and Export... ; "Chain verified" or "Chain broken" at its head
Enforcement	How the repository enforces the policy: a required check on a hosted service (Add the CI check writes the GitLab or GitHub check file), the pre-receive hook on a self-hosted server (Download hook kit), or "detective only" on a shared folder; Check enforcement asks the host

Verify all seals checks every baseline, governed document, the history and the activity log against your copy and reports its findings, each with "What to do". **Record in the activity log** writes the verification into the log. Certificate signatures and seals of files with mixed line endings are confirmed by the desktop application.

15.7 Sharing in a governed project

With governance on, the Share dialog applies the policy: a refusal of **Share records** names who holds it; a policy that requires signed commits warns that "The browser cannot sign the version it shares"

and suggests the desktop application for a signed version; governance records left unticked are flagged. Single-use waivers are spent by the share they allow.

15.8 A governed project with governance off

A project that has a policy still applies it to what you save and share, even in a browser where governance is off. The status bar shows **Governed project**, and **Share** becomes **Send for review**:

- your changes go to a branch of your own, `work/<login>/<branch>`, replaced at each sending; the governed branch is never written by you;
- the outcome: "Sent 3 file(s) for review on work/evidal/main (version ...). Someone with governance brings them into main; until then they stay your unshared changes here, and Get updates settles them once they are in.";
- governance records and the repository's check files are never sent.

On the reviewer's side, with governance on, **Reviews** ► **Sent for review** lists each work branch with **Compare** (what it changed since the version it was sent from), **Bring in...** and **Delete...** **Bring in...** makes the colleague's work your unshared changes, each governed document saved with your receipt and one line in the activity log ("Work brought in"); the question lists each file, including those merged with your copy and those left out. You then review and share. **Delete...** removes the work branch; what it held and was not brought in stays with its author as unshared changes.

15.9 Copy without governance

Copy without governance..., in the account menu of a governed project, makes a new Git repository of the project as it is in its shared version, without the governance records and the SafeSuite checks of its CI, with a `PROVENANCE.md` naming the source. It downloads as a zip: unzip it and open the folder in the desktop application, or push it to a new repository on your host. Nothing from the copy ever comes back into the governed project; to propose a change there, open the governed project with governance off and use **Send for review**.

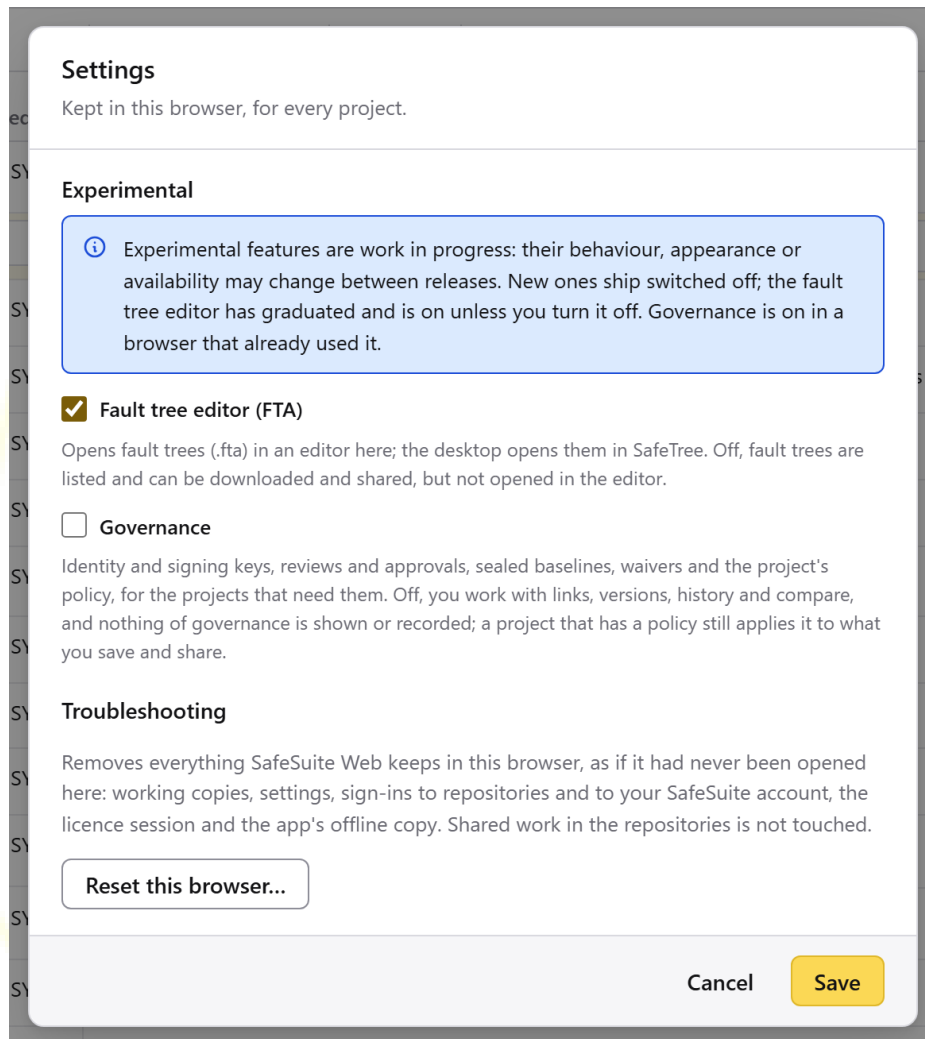
15.10 What the desktop application adds

The desktop application signs every version it shares, makes a tag for each sealed baseline, signs at Levels 2 to 5, confirms certificate signatures and mixed-line-ending seals, withdraws lapsed signatures on its own during **Verify all seals**, records every save in any repository, and attests a second key for an existing identity. The records are compatible in both directions.

16 Settings, shortcuts and troubleshooting

16.1 Settings

Settings in the rail (key „), or the **Settings** link on the start page, opens the dialog. Its subtitle says where the choices live: "Kept in this browser, for every project."



The screenshot shows a 'Settings' dialog box with the subtitle 'Kept in this browser, for every project.' The dialog is divided into sections: 'Experimental', 'Fault tree editor (FTA)', 'Governance', and 'Troubleshooting'. The 'Experimental' section contains an information icon and a text box explaining that experimental features are work in progress. The 'Fault tree editor (FTA)' section has a checked checkbox and a description. The 'Governance' section has an unchecked checkbox and a description. The 'Troubleshooting' section has a description and a 'Reset this browser...' button. At the bottom right are 'Cancel' and 'Save' buttons.

Settings
Kept in this browser, for every project.

Experimental

Experimental features are work in progress: their behaviour, appearance or availability may change between releases. New ones ship switched off; the fault tree editor has graduated and is on unless you turn it off. Governance is on in a browser that already used it.

☒ **Fault tree editor (FTA)**
Opens fault trees (.fta) in an editor here; the desktop opens them in SafeTree. Off, fault trees are listed and can be downloaded and shared, but not opened in the editor.

☐ **Governance**
Identity and signing keys, reviews and approvals, sealed baselines, waivers and the project's policy, for the projects that need them. Off, you work with links, versions, history and compare, and nothing of governance is shown or recorded; a project that has a policy still applies it to what you save and share.

Troubleshooting
Removes everything SafeSuite Web keeps in this browser, as if it had never been opened here: working copies, settings, sign-ins to repositories and to your SafeSuite account, the licence session and the app's offline copy. Shared work in the repositories is not touched.

[Reset this browser...](#)

Cancel Save

The Settings dialog

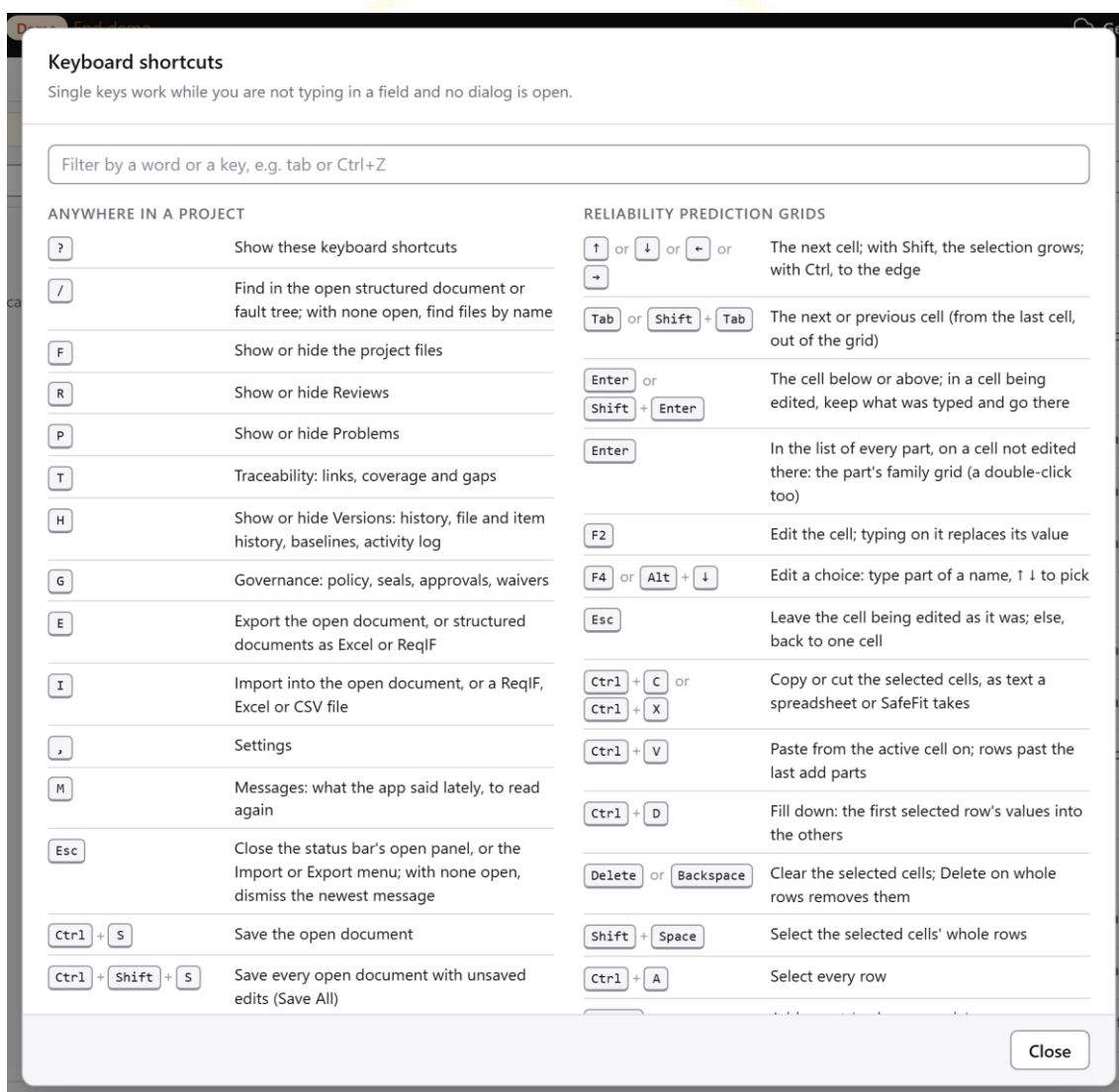
Section	Setting
Experimental	Fault tree editor (FTA): opens fault trees in the editor here; off, a fault tree can only be downloaded. Governance: identity and signing keys, reviews and approvals, sealed baselines, waivers and policies (chapter 15)
Licence (licensed sites)	The account line, each feature with its state, Sign in... or Sign out of the licence

Section	Setting
Troubleshooting	Reset this browser...

Save keeps the choices; **Cancel** leaves them.

16.2 Keyboard shortcuts

The legend, **Keyboard shortcuts**, opens with the ? key, the keyboard icon in the status bar, or **More** on a phone. It has a filter box, and the note that single keys work while you are not typing in a field and no dialog is open. On a Mac, $\mathbb{C}$ works where Ctrl is shown.



Keyboard shortcuts
Single keys work while you are not typing in a field and no dialog is open.

Filter by a word or a key, e.g. tab or Ctrl+Z

ANYWHERE IN A PROJECT		RELIABILITY PREDICTION GRIDS	
[?]	Show these keyboard shortcuts	[↑] or [↓] or [←] or [→]	The next cell; with Shift, the selection grows; with Ctrl, to the edge
[/]	Find in the open structured document or fault tree; with none open, find files by name	[Tab] or [Shift] + [Tab]	The next or previous cell (from the last cell, out of the grid)
[F]	Show or hide the project files	[Enter] or [Shift] + [Enter]	The cell below or above; in a cell being edited, keep what was typed and go there
[R]	Show or hide Reviews	[Enter]	In the list of every part, on a cell not edited there: the part's family grid (a double-click too)
[P]	Show or hide Problems	[F2]	Edit the cell; typing on it replaces its value
[T]	Traceability: links, coverage and gaps	[F4] or [Alt] + [↓]	Edit a choice: type part of a name, ↑ ↓ to pick
[H]	Show or hide Versions: history, file and item history, baselines, activity log	[Esc]	Leave the cell being edited as it was; else, back to one cell
[G]	Governance: policy, seals, approvals, waivers	[Ctrl] + [C] or [Ctrl] + [X]	Copy or cut the selected cells, as text a spreadsheet or SafeFit takes
[E]	Export the open document, or structured documents as Excel or ReqlF	[Ctrl] + [V]	Paste from the active cell on; rows past the last add parts
[I]	Import into the open document, or a ReqlF, Excel or CSV file	[Ctrl] + [D]	Fill down: the first selected row's values into the others
[,]	Settings	[Delete] or [Backspace]	Clear the selected cells; Delete on whole rows removes them
[M]	Messages: what the app said lately, to read again	[Shift] + [Space]	Select the selected cells' whole rows
[Esc]	Close the status bar's open panel, or the Import or Export menu; with none open, dismiss the newest message	[Ctrl] + [A]	Select every row
[Ctrl] + [S]	Save the open document		
[Ctrl] + [Shift] + [S]	Save every open document with unsaved edits (Save All)		

Close

The keyboard shortcuts legend

Anywhere in a project

Key	Action
?	Keyboard shortcuts
/	Find in the open structured document or fault tree; otherwise files by name
F, R, P, T, H, G, E, I, ,	Files, Reviews, Problems, Traceability, Versions, Governance, Export, Import, Settings
M	Messages
Esc	Close the status-bar panel or a rail menu; otherwise dismiss the newest message
Ctrl+S, Ctrl+Shift+S	Save, Save All
Ctrl+P	Print
[and]	Previous and next document tab

Explorer

Key	Action
↑ ↓ Home End → ←	Move, expand, collapse
Enter or Space	Open; expand or collapse
F2, Delete	Rename, delete
Ctrl+X, Ctrl+C, Ctrl+V, Ctrl+D	Cut, copy, paste, duplicate
Ctrl+Z, Ctrl+Y	Undo, redo
Ctrl+F	Find files and folders
Shift+F10	The item's menu

Structured document editor

Key	Action
Ctrl+F, F3, Shift+F3	Find and filter; next, previous
↑ ↓ PageUp PageDown Home End (with Shift)	Move between rows, extend the selection
Ctrl+A	Select all rows
Ctrl+C, Ctrl+X, Ctrl+V, Ctrl+D	Copy, cut, paste, duplicate rows
Ctrl+L	Link to the copied rows

Key	Action
Insert, Shift+Insert	New row below, above
Delete	Delete rows
Ctrl+Z, Ctrl+Y	Undo, redo
Alt+Shift+↑ ↓ ← →	Move up, move down, promote, demote
Tab, Shift+Tab	Next, previous cell
Esc or Ctrl+Enter	Leave a rich-text cell

The legend also lists the keys of the FIT grids, the fault tree editor, the FMEA worksheet and the dialogs (Enter runs the main action, Esc closes, Alt+← and Alt+→ copy a row across in a comparison).

16.3 Messages and new versions

The bell in the status bar keeps the latest messages (chapter 6). When a new version of the application is ready, the message "A new version of SafeSuite Web is ready. Save your open edits, then reload to use it." offers **Reload**; nothing changes until you do.

16.4 Troubleshooting

What you see	What to do
"The browser removed your local copy of ..."	The browser deleted the site's storage, usually because the disk was nearly full. Press Check out again ; if your unshared changes were backed up to the repository, the checkout offers Restore it into this checkout . Installing the app (account menu ► Install app) prevents this
"The browser asks again whether SafeSuite may open the folder ..."	Chrome and Edge forget a folder's permission after a reload. Press Allow access
"Your session has ended. Please sign in again."	The GitLab session expired. Sign in again from the Connect page or the account menu
"GitHub.com no longer accepts the token (it has expired or was revoked)."	Make a new fine-grained token and sign in with it
"Could not reach the repository (...)"	Check the network; the project keeps working locally meanwhile
"The repository refused the change: ..."	The branch is protected, or the folder's branch is checked out. Share to a branch you may write to, or ask your administrator
"N file(s) changed on both sides need your decision."	Open Get updates or the Team server panel's Resolve decisions... and decide (chapter 12)

What you see	What to do
"Not saved. The document contains errors."	Show the errors and fix the configuration before saving
A document refuses to open	Open Check & Repair... from the file's menu or the Problems panel (chapter 8)
The licence note says a feature is off	Chapter 4 lists every licence state and its remedy
"An unexpected error occurred." with Reload	Reload the page. Your working copy and saved changes are kept; unsaved edits in an open editor are not
Everything looks wrong and nothing helps	Settings ▶ Troubleshooting ▶ Reset this browser... removes everything the application keeps in this browser: working copies, settings, sign-ins, the licence session and the offline copy. The dialog counts the working copies and unshared changes that would be lost; share them first. Shared work in the repositories is never touched

17 Limits, and what the web application does not do

SafeSuite Web covers the everyday work on a project. It deliberately leaves some tasks to the desktop application, and the browser imposes a few limits of its own.

17.1 Left to the desktop application

Task	In the web application
The AI assistant: chat about the project, document generation, tabular imports, summaries, coherence checks, repair with the assistant	Not included. The Team server panel states "AI assistance is off." Nothing leaves the browser for an AI service
Jira integration: binding a project, refreshing statuses, creating and importing issues	Not included. Jira columns are shown read-only; links to issues can be recorded and opened
Creating a new fault tree or a new FMEA	Duplicate an existing one, or create it in the desktop
Safety assessments, cybersecurity and CRA reporting, the vulnerability feed, SBOM export of the project	Not included. An SBOM file can be imported as a structured document
Signing the versions you share, and tagging sealed baselines	Versions shared from the web are unsigned; the desktop signs them
Signing at Levels 2 to 5 (Windows account, TPM, hardware token, certificate); verifying certificate signatures	The web signs at Level 0 or 1 and asks the desktop to confirm the rest
Maintaining the project file (.ssproject), moving or renaming a project	Read-only here
Shared datatypes, unknown attribute types and some rich-text formatting	Shown, kept read-only, edited in the desktop
Editing a file in place on the disk, "Show in Explorer"	Save to computer and Upload a new version... instead
ReqIF on the demo	Not available in a demo seat

17.2 Browser and device limits

Limit	Value
Repositories in a folder	Chrome and Edge on a computer only
Phone layout (at most 760 px wide)	Structured documents as cards; FMEA, FIT and fault trees read-only; configuration, level styles,

Limit	Value
	columns, PDF options and side by side on a larger screen
Browser storage	May be deleted by the browser when the disk is nearly full; the backup branch and Install app protect against it. Safari may wipe a site's storage after 7 days without use
Text editor	Files over 1 MiB, binary or UTF-16 files and invalid UTF-8 are shown, not edited
Upload	50 files at a time, a folder of at most 200 files, 100 MB per file
ReqIF import	200 MB
Excel or CSV import	100 MB
FMEA and FIT CSV imports	64 MB
Fault-tree FMEA import	16 MB
Files picked for Compare	50 MB
Pictures in rich text	10 MB
Open tabs restored	10
Recent projects in the account menu	9
Item history and "to get" counting	The latest 100 versions

17.3 Other known limits

- Problems, the Traceability dialog and PDFs leave out documents that are not downloaded, and say so.
- Undo of a deletion lasts only until the page is closed when no backup could be taken.
- FMEA failure modes paste between FMEAs in the same page, not between browser tabs; structured-document rows paste only within the same document.
- Renumbering designators in a FIT prediction does not update the FMEAs that use them.
- In a repository that holds several SafeSuite projects, the start page's name, the Problems scan and Traceability use the top-most project.
- Without governance, a plain baseline needs a repository that supports tags; a project with a governance policy needs governance on to create a baseline.

18 Glossary

Term	Meaning
Backup branch	A branch of your own in the repository, safesuite-wip/<you>/<branch>, where your unshared changes are copied in the background while you are signed in
Backups in this browser	The versions of your files that a merge, Revert, restore, repair, delete or upload replaced, kept inside the working copy until you delete them
Base version	The repository version your working copy was checked out at, or last updated to. Every Share and Get updates compares it with yours and the repository's
Baseline	A named version of the project, for example "v1.0 — Type Approval". A Git tag; with governance, also sealed and signed
Branch	A line of versions in a repository. Teams usually work on the default branch, often main
Check out	Download the files of a branch into this browser, making a working copy
Conflict	A file, field or set of lines changed on both sides since the base version. The application asks you what to keep
Demo	A seat lent for a visit: the demo project opens in memory; nothing is kept
Get updates	Bring the versions others shared into your working copy, merging them with your own changes
Git	The version-control technology underneath the repository. SafeSuite uses it without asking you to learn it
Governance	The optional records of reviews, approvals, seals, waivers, delegations and policy, signed and logged
Host	The service or place that holds the repository: GitLab, GitHub, a Git server, or a folder
Item	A row of a structured document

Term	Meaning
Link	A traceability relation between two items, usually in different documents
Merge	The combination of two versions of a file changed on both sides: item by item for structured documents, line by line for text
Presence	The notice "... is editing since 10:42" that others see while you edit a file
Problems	Review markers, invalid values, broken links and unreadable documents, listed for the project
Repository	The shared, versioned home of a project: every version of every file, with who changed it, when and why
Reservation (lock)	The exclusive right to edit a file that cannot be merged (FMEA, FIT, fault tree), taken at your first edit and honoured by the desktop
Revert	Discard your unshared change of a file and return to the shared version, with a backup first
Share	Send your saved changes to the repository as one new version with a summary
Structured document	A .strdoc file: a table of typed items with attributes, the format of requirements and analyses in SafeSuite
Unshared change	A saved change that is in this browser only, not yet shared
Version	A snapshot of the branch after a share, identified by a short code such as a3f9c21
View	A set of visible columns of a structured document, with their order and widths
Working copy	The project's files in this browser, with the base version, your unshared changes and the backups

Appendix A: Technical requirements (informative)

Requirement	Details
Browser	A current Chrome, Edge, Firefox or Safari. Chrome or Edge on a computer for repositories in a folder and for installing the app
Network	Access to the SafeSuite Web site and to your repository host. Once loaded, the application starts without a network; sharing and getting updates need the repository
Storage	A few hundred megabytes of browser storage for the working copies, more for large projects. Installing the app asks the browser to keep the storage
Repository	A GitLab or GitHub project, a Git server reachable over HTTPS, or a Git repository in a folder (a clone or a bare repository)
Licence	A SafeSuite account on a licensed site. One computer at a time per licence

Appendix B: Support

- Documentation, videos and news: www.cynoss.net.
- Licences, accounts and questions: support@cynoss.net.
- The application's **About SafeSuite Web** shows the version and the build to quote in a request.