

How our suite of tools helps with your compliance journey (Automotive example)

- [Overview](#)
- [Plan](#)
 - [ISO 9001](#)
 - [Engagement of people](#)
 - [Process approach and Improvement](#)
 - [Evidence-based decision making](#)
 - [ISO 27001](#)
 - [Confidentiality and Information integrity](#)
 - [Availability of data](#)
 - [ISO 31000](#)
 - [Risk definition & mitigation](#)
- [Do, Check, Act](#)
 - [IATF 16949](#)
 - [ISO 26262](#)
 - [Requirements](#)
 - [Coverage and traceability](#)
 - [ASPICE](#)
 - [Testing techniques](#)
 - [Execution](#)

Overview

In this article, we will describe how our tools help at different stages of establishing & improving the compliant QMS, using the [PDCA cycle](#) (Plan, Do, Check, Act) as the guideline.

Plan

This stage is about **strategic** design of the QMS structure and its processes. We will focus on a few popular universal standards, namely **ISO 9001**, **27001**, and **31000**.

ISO 9001

This standard sets out the criteria for a QMS and applies to the organizations engaged in design, development, production, and servicing of goods (i.e., to most software development organizations).

ISO 9001 is based on [7 quality management principles](#):

#	Title	Summary Statement
1	Customer focus	The primary focus of quality management is to meet customer requirements and to strive to exceed customer expectations.
2	Leadership	Leaders at all levels establish unity of purpose and direction and create conditions in which people are engaged in achieving the organization's quality objectives.
3	Engagement of people	Competent, empowered, and engaged people at all levels throughout the organization are essential to enhance its capability to create and deliver value.
4	Process approach	Consistent results are achieved more effectively and efficiently when activities are managed as a coherent system.
5	Improvement	Successful organizations have an ongoing focus on improvement.
6	Evidence-based decision making	Decisions based on the analysis and evaluation of data and information are more likely to produce desired results.
7	Relationship management	For sustained success, an organization manages its relationships with interested parties, such as suppliers.

In this article (this section and beyond), we will dive deeper into principles 3-6.

Engagement of people

There are two important components to this principle: **empowerment/competence** and **collaboration**.

When it comes to competence, it is essential to have documentation covering not just the basics, but also the nuances of the processes, tools, possible configurations and extensions/integrations, etc. Xray has plenty of self-paced resources ([DC/Server](#), [Cloud](#)) to facilitate training, improve knowledge, and ensure seamless adoption. Besides the extensive user guide documentation available online, there are also many courses on [Xray Academy](#) to understand not only Xray essentials but also more advanced topics, such as test automation and optimized test design.

To facilitate collaboration and foster clarity, you can invite every team member to participate in quality-related tasks, removing friction that exists whenever different team roles use siloed tools:

- Artifacts can be shared in a variety of formats.
- Comments and evidence can be added to all Xray issue-based entities (e.g., Tests) and also to Test Runs.

Process approach and Improvement

The effectiveness of the quality management approach depends on how thoroughly it is integrated into all facets of your organization. Having the Atlassian ecosystem as the single source of truth significantly simplifies this process aspect.

Furthermore, Jira and Xray enable multiple customizations to adapt to the evolving organizational needs:

- Different deployment versions (Server/DC and Cloud) with the seamless connectivity between Jira and Xray
- Flexible ways for organizing your project related items (all-in-one vs testing entities separately)
- Different levels of objectives with traceability and quality gates
- Improved visibility into interrelated tasks with proper assignments and notifications
- Establish KPIs and quantifiably measure them
 - add custom fields to any entity (e.g. probability, severity, performance thresholds)
 - implement additional customizations (e.g., ScriptRunner, Automation for Jira)
 - integrate with other Jira apps to widen the feature set provided by our tools

Evidence-based decision making

This principle is primarily enabled by detailed reports and formats that promote easier visibility and awareness. With Jira and Xray, you can export data in compliance-focused, human-readable formats and automate data snapshots.

All core entities, including Test Runs (i.e. results), can be exported to PDF, Word, or Excel, using fully customizable documents in terms of layout. This is a way to obtain a formal, readable copy of the relevant test data, no matter whether it is related to test specification or execution.

Two options exist to achieve this: either using the built-in [Document Generator](#) capabilities, or by using the more complete and flexible [Xporter App](#).

With Xporter it is possible to automate the creation of these documents and, for example, generate them upon a transition of a workflow, or attach them to an existing Confluence page, or send them via email.

To consolidate the information from Jira and Xray, you can also use [Jira Snapshots](#):

"Easy to make reports, diagnosis is quick and straightforward. The FDA submission requires specification reports and traceability reports. Jira Snapshots compiles these reports from the Jira and Xray data, avoiding burdening the team."

[Caris Life Sciences Success Case](#)

ISO 27001

This standard establishes the requirements for an information security management system. [ISO 27001](#) focuses primarily on maintaining the **confidentiality, integrity, and availability** of information:

What are the three principles of information security in ISO/IEC 27001, also known as the CIA triad?

1. **Confidentiality**

→ *Meaning*: Only the right people can access the information held by the organization.

⚠ *Risk example*: Criminals get hold of your clients' login details and sell them on the Darknet.

2. **Information integrity**

→ *Meaning*: Data that the organization uses to pursue its business or keeps safe for others is reliably stored and not erased or damaged.

⚠ *Risk example*: A staff member accidentally deletes a row in a file during processing.

3. **Availability of data:**

→ *Meaning*: The organization and its clients can access the information whenever it is necessary so that business purposes and customer expectations are satisfied.

⚠ *Risk example*: Your enterprise database goes offline because of server problems and insufficient backup.

Confidentiality and Information integrity

To support confidentiality and information integrity principles, Jira and Xray allow you to:

- Implement access control and permissions for any entity (i.e. Project, Story, Test Plan, Test Execution, etc.)
- Use e-signatures to track explicit approval
 - Xray issue-based entities can all be digitally signed, using one of many available Jira apps for that purpose. This is a unique advantage as it provides full control over the core testing activities.

Issue permissions

Permission	Granted to
Archive Issues Ability to archive issues for a specific project.	
Assignable User Users with this permission may be assigned to issues.	Application access <ul style="list-style-type: none">• Any logged in user
Assign Issues Ability to assign issues to other people.	Application access <ul style="list-style-type: none">• Any logged in user
[Xray] Override Assign Test Runs Allows users to set the Test Run assignee without the Edit Issues and Assign Issues permissions.	
Close Issues Ability to close issues. Often useful where your developers resolve issues, and a QA department closes them.	Application access <ul style="list-style-type: none">• Any logged in user
Create Issues Ability to create issues.	Application access <ul style="list-style-type: none">• Any logged in user

Atlassian is ISO 27001 certified. Xray holds a SOC 2 Type 2 certification.

Availability of data

With Xray, you can ensure data persistence without tampering, maintain history visibility, and track changes.

- In order to enable auditing and facilitate diagnosis, data must be stored, and changes, whenever applicable, need to be clearly identified. In Xray, all data is persisted and can be easily tracked using a historical timeline. You can even monitor individual test step modifications - if a step is changed, the record gets into the Xray Test History tab on the Test issue screen (under Activity).

Xpand IT Admin made changes - 1 minute ago

V1

Manual Test Steps

[Action:Push the following body in the POST request
[Data:{ "operator" : "+", "value1" : "1", "value2" : "1" }]
[Expected Result:Service return a 200 ok and in the
body the following information:

{ "operator" : "+", "value1" : "1", "value2" : "1", "result":
"2" } }

[Action:Push the following body in the HTTP POST
request] [Data:{ "operator" : "+", "value1" : "1",
"value2" : "1" }] [Expected Result:Service return a 200
ok and in the body the following information:

{ "operator" : "+", "value1" : "1", "value2" : "1", "result":
"2" } }

- Historical results (i.e. Test Run details) cannot be modified. The same applies to past changes made on Jira issue-based entities.
 - If a Test changes, the recorded results don't. A warning appears in case the user is still allowed to rerun the test. If they are not allowed (because the Test Execution issue is resolved) then the execution cannot even be restarted.

CALC / Test Execution: CALC-284 / Test: CALC-225
Validate the calculator service

Dataset Return to Test Execution Execute with Exploratory App

Test definition has been modified. You can reset the execution of this test by pressing the "Reset" button. This will load the new test definition and delete the current execution data. If you wish to update the Test definition and maintain the current execution, you can click on the "Merge" button.

Reset Merge

Execution Status: PASS

Started On: 19/Apr/23 11:39 PM Finished On: 19/Apr/23 11:40 PM

Assignee: Xpand IT Admin Versions: v4.0
Executed By: Xpand IT Admin Test Version: v1
Tests environments: - Revision: -

Comment Preview Comment Execution Defects (0) Execution Evidence (0)

Test Details

Custom Fields
There are no Test Run Custom Fields defined.

Test Description

Test Issue Links

Pre-Conditions
CALC-228- Make sure the API service is available

Test Steps

1	Action	Data	Expected Result
	Push the following body in the POST request	{ "operator" : "+", "value1" : "1", "value2" : "1" }	Service return a 200 ok and in the body the following information: { "operator" : "+", "value1" : "1", "value2" : "1", "result": "2" }

Actual Result Comment Defects Evidence Step State: PASS

SOC 2 is similar in its goals & controls and includes five Trust Services Criteria: Security, Availability, Processing Integrity, Confidentiality and Privacy. The features described above also align with the sample SOC2 checklists.

ISO 31000

This standard family sets the guidelines for engaging in Enterprise Risk Management (ERM). It provides guidance on how to identify, assess, treat, and communicate risks.

ISO 31000 cannot be used for certification purposes directly. But organizations can compare their risk management practices with an internationally recognized benchmark, providing sound principles for effective management and corporate governance.

Risk definition & mitigation

ISO 31000 defines risk as the "effect of uncertainty on objectives". There are three key concepts:

- Potential event
- Probability of that event occurring
- The resulting severity of the outcome, should the event occur

In line with the improvement principle of ISO 9001, ISO 31000 puts great emphasis on the iterative nature of risk management, drawing on new knowledge and analysis for the revision of actions and controls at each stage of the process.

It would be near impossible to successfully implement and sustain the ISO 31000 risk management standard if an organization's process is heavily dependent on paper-based communication and record keeping.

In Jira and Xray, you can have the ability to:

- manage risks (e.g., create, edit, delete) and track their status
- classify risks by impact, probability, and other user-defined fields

We will dive deeper into this topic when discussing the ISO 26262 standard later on.

Do, Check, Act

The Do stage of PDCA is more about the **tactical** implementation of the QMS. Check and Act stages detail how the quantified results are processed. The goals are to determine the effectiveness and efficiency of each process toward its objectives, to communicate these findings to the stakeholders, and to develop new best practices based on the audit.

As these stages build **on top** of the strategic design one described above, we will focus more on the unique aspects and "hands-on" examples. We will review the popular standards specific to the automotive sector, namely **IATF 16949**, **ISO 26262**, and **ASPICE**, to more closely analyze the requirements to achieve compliance.



For the examples from the [medical](#) sphere, please refer to the success cases from [Caris Life Sciences](#) and [MyndTec](#) as well as [this webinar](#).

IATF 16949

This standard is an **ISO 9001 interpretation** agreed upon by major American and European automotive manufacturers. It focuses on the avoidance of errors (not on the discovery) and defines the requirements for the development, production, and installation of automotive-related products. It is more about hardware, while ASPICE covers software.



IATF 16949 is an **independent QMS standard** that is fully aligned with the structure and requirements of ISO 9001. Therefore, the IATF 16949 **cannot be implemented alone** - it must be implemented as a supplement and in conjunction with ISO 9001.

The Jira+Xray capabilities supporting ISO 9001 (described above) can also facilitate [core tools](#) from the automotive industry mentioned in IATF 16949, such as:

- Advanced Product Quality Planning (APQP)
- Failure Mode and Effects Analysis (FMEA)
- Statistical Process Control (SPC)
- Measurement Systems Analysis (MSA)
- Production Part Approval Process (PPAP)

ISO 26262

The standard covers functional safety in the automotive industry and provides the definition as “the absence of unreasonable risk due to hazards caused by malfunctioning behavior of electrical/electronic systems”. The standard covers the entire lifecycle of a safety-related system and outlines a risk-based approach to developing and implementing such systems in road vehicles.



There is some overlap between ISO 26262 and ASPICE. ISO 26262 does require that the software development is conducted in accordance with a defined process, and ASPICE can provide a framework for meeting this requirement.

There are two types of errors mentioned:

Random errors: They follow a probability distribution, so we can estimate and plan for them. Various safety analysis methods listed in ISO 26262 (FMEA, FMEDA, DFA, etc.) are the primary mechanisms to control this error type.

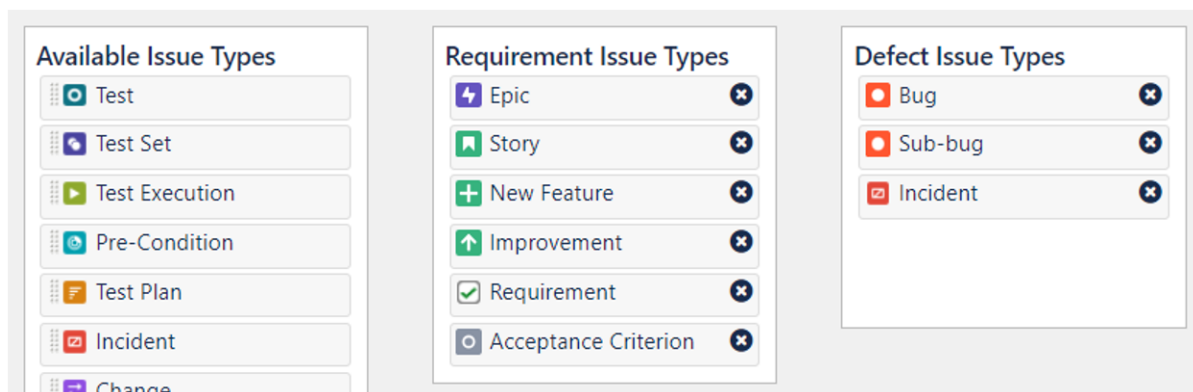
Systematic errors: Failures related in a deterministic way to a certain cause. They can only be eliminated by a change of the design or of the manufacturing process, operational procedures, documentation or other relevant factors. Process control measures in both ISO 26262 and ASPICE, which Xray facilitates, help make sure these errors are mitigated throughout the development process.

The access controls and the data integrity measures mentioned previously enable efficiency and transparency in the Verification and Validation processes covered by this standard. We will focus on the Verification level here and talk more about Validation in the ASPICE section.

Requirements

Define what a requirement is and its relevant metrics

Xray provides multiple levels of requirements and different ways to define the hierarchical relationship (e.g., using issue links, sub-tasks).



So, you can reflect the ISO 26262 and ASPICE model for documenting the system (both hardware and software) requirements in a way that **clearly reflects the scope, dependencies, status, etc.** For this example, let's assume we are working with an Automated Driving System, here is how a few requirements for it could be arranged:

Key	Summary	ASIL	Progress	Requirement Status	Σ Time Spent	Item Health
	▼ HARDWARE		<div><div></div></div>		1w 2d 4h	OK
CAR-3	▼ Sensors	B	<div><div></div></div>	<div>NOK</div>	1w 2d 4h	OK
CAR-4	▼ Sensors should withstand the temperature range -40°C to 150°C	A	<div><div></div></div>	<div>OK</div>	1w 1d	Great
CAR-10	Test 1 - Upper limit		<div><div></div></div>			Great
CAR-18	Test 2 - Lower limit		<div><div></div></div>			Great
CAR-2	▼ Sensors should support a horizontal view angle of 270°	B	<div><div></div></div>	<div>NOK</div>	1d 4h 1d	At Risk
CAR-8	▼ Rotational test		<div><div></div></div>		4h	OK
CAR-25	Rotation left is limited		<div><div></div></div>			At Risk
CAR-1	▼ Platform and infrastructure	A	<div><div></div></div>	<div>NOTRUN</div>	2d	OK
CAR-22	▼ The infrastructure should support automotive-specific communications pr	A	<div><div></div></div>	<div>NOTRUN</div>	2d	OK
CAR-12	▶ Test Plan for infrastructure		<div><div></div></div>		2d	OK
	▼ SOFTWARE		<div><div></div></div>			At Risk
	▼ Safety		<div><div></div></div>			At Risk
CAR-5	▼ Driver Assistance requirements	C	<div><div></div></div>	<div>NOTRUN</div>		At Risk
CAR-6	▶ Adaptive cruise control requirements	C	<div><div></div></div>	<div>NOTRUN</div>		At Risk
CAR-15	Highway pilot requirements	D	<div><div></div></div>	<div>UNCOVERED</div>		At Risk
	▼ Object Detection		<div><div></div></div>			At Risk
CAR-17	Object Detection	B	<div><div></div></div>	<div>UNCOVERED</div>		At Risk

The hierarchical view is achieved with the Structure add-on for Jira

We can **capture and track key pieces of information**, both in structured (custom fields) and unstructured (description, attachments, mockup links, etc.) formats. Specifically, we can track industry-specific Components and compliance-related aspects like Automotive Safety Integrity Levels (ASILs) and their parts:



ASIL = Impact x (Probability x Controllability). Although the definition of risk in ISO 31000 is slightly different, these 3 key concepts are still largely applicable for that standard as well.

Components

<input type="text" value="Q"/>	QUICK FILTERS:	Active	Archived
<input type="text" value="Component name"/>	<input type="text" value="Lead (optional)"/>		
Component	Status	Issues	
detection	ACTIVE	1 Issue	
interfaces	ACTIVE	3 Issues	
safety	ACTIVE	4 Issues	
sensors	ACTIVE	6 Issues	



CAR / CAR-3

Sensors

[Edit](#) [Comment](#) [Assign](#) [More](#) [To Do](#)

Details

Type: [Epic](#)
Priority: [Critical](#)
Component/s: [sensors](#)
Labels: [None](#)
Epic Name: [Sensors](#)
Requirement Status: [NOTRUN](#)

Probability: [High](#)
Impact: [Severe Injuries](#)
Controllability: [Normal](#)
ASIL: [B](#)
[None](#)
[A](#)
[B](#)
[C](#)
[D](#)

Description
[Click to add description](#)

We could then classify the requirement entities by ASIL directly, or just impact, or any other user-defined field.

Proper risk management

Handling risks is an intrinsic part of good testing and is essential to organizations that work in highly regulated environments. Xray supports [Risk-Based Testing \(RBT\)](#) and allows you to define risks at different levels: project, requirement, or test.

We have already set up the ASIL and related values to match ISO 26262 **at the requirement/story level**. We can choose to maintain the same level of detail **at the test level** or do a more customized risk assessment - e.g., combining ASIL with complexity for an aggregate Risk Score metric:

Add Tests to Test Execution CAR-27

Assignee Issue Assignee

Select **Search** JQL

Filter(s) +

Project
CAR (CAR) x

Test Type
Choose the Test Type

Contains text




Folder ⓘ
All
☐ Include sub-folders

Risk Score between
4
and
5
Complexity
High x Medium x

Clear Search

☒ Hide Tests in non-executable Statuses

Add selected Add all (3) Cancel

Issue Type	Key	Summary	Test Type	Complexity	Risk Score
<input type="checkbox"/> 	CAR-16	Test 1 for Adaptive cruise control	Manual	High	5
<input type="checkbox"/> 	CAR-10	Test 1 - Upper limit	Manual	High	4
<input type="checkbox"/> 	CAR-8	Rotational test	Manual	High	5

Showing 1 to 3 of 3 entries

First Previous **1** Next Last

You can not only add tests to the execution based on risk values in filters, but also prioritize the execution order accordingly.



If you decide to prioritize the same metrics between requirements and tests, you can set up the rules for inheriting using [Jira Automation](#), so that you don't need to set the values manually for each entity.

Implement workflows to have explicit control over the process

Ensure issues get done by assigning them to users and tracking them using workflows. ISO 26262 requires different verification and approval activities depending on the testing type, so the ability to customize is critical. To elaborate on the list we mentioned in the Process section of ISO 9001, with Jira and Xray it is possible to:

- implement an approval mechanism, commonly having one approver
- Implement quality gates based on field status (see the nuances below)
- make items "read-only" when transitioned to a certain workflow status by setting a Jira property ("jira.issue.editable")
- restrict usage of Tests in a certain workflow status ([see how](#))
- disallow executions for Test Executions in a specific status ([see how](#))

Learn more in [Using Jira workflows for testing purpose](#).

To be clear, Xray doesn't have an *explicit* feature for "quality gates". However, Xray does have mechanisms to implement quality-related checks, mostly acting at the workflow status.


Examples [leveraging ScriptRunner](#) include:

- validating a requirement/story based on their coverage status, before allowing to make a transition
- reopening/transitioning linked Tests to a requirement
 - whenever you change the specification of a requirement, you most probably will need to review the Tests that you have already specified.
- inhibiting transition of the Test Plan workflow status, unless all tests or a certain % are passing
 - can be accomplished with the combination of ScriptRunner and Xray REST API or custom field

Coverage and traceability

Guarantee requirements coverage

Epic and Story issues, common in Agile environments, can be covered by creating tests and tracking coverage directly from Jira. From the user story issue screen you can immediately create a Test. It is also easy to go back to the user story, using the issue link that was created automatically between the Test and the user story.

 CAR / CAR-6

Adaptive cruise control requirements

EditCommentAssignMoreTo DoIn ProgressDoneAdmin

Details

Type:Requirement

Priority:Major

Component/s:safety

Labels:None

Requirement Status:NOTRUN

Probability:High

Impact:Severe Injuries

Controllability:Uncontrollable

ASIL:C

Status:TO DO (View Workflow)

Resolution:Unresolved

Description

Click to add description

Test Coverage

Add TestsExecute...

New Test

Existing Tests

Tests from Test Sets

LOWING ANALYSIS SCOPE

on: None - latest execution; Environment: All Environments

P

Status

Resolution

Key

Summary

Test Type

Test Runs

Test Status

TO DO

Unresolved

CAR-16

Test 1 for Adaptive cruise control

Manual

TODO

Showing 1 to 1 of 1 entries

First

Previous

1

Next

Last

Attachments

...

Issue Links

+

satisfies

✓

CAR-5 Driver Assistance requirements

−

TO DO

tested by

⊞

CAR-16 Test 1 for Adaptive cruise control

○

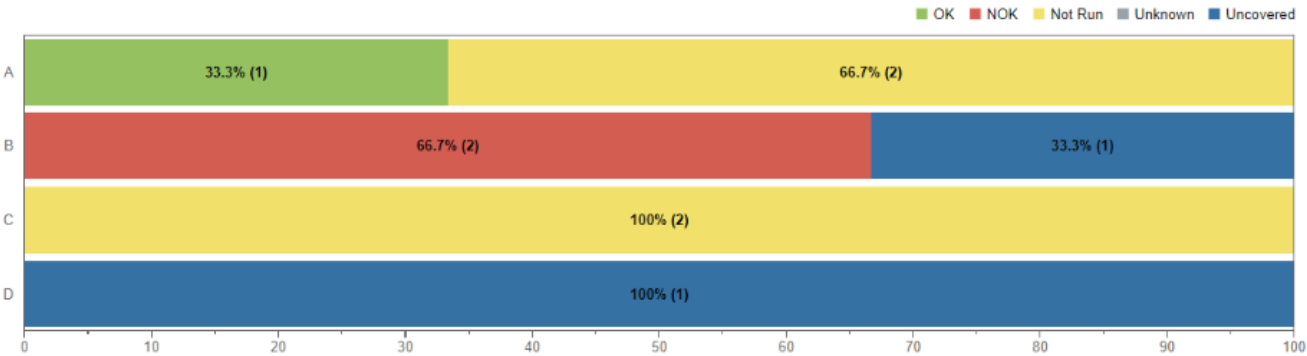
TO DO

Within the Test Coverage Report, you can have a birds-eye view of quality status of your requirements, based on the tests that cover them and their results. To ease your analysis, you can group stories by specific metrics; for example, by risk level. That way you can have a better understanding of whether a higher priority issue is ready to be released or not.

Coverage information is updated in real-time and is multidimensional. In other words, we can analyze requirements from different perspectives: risk level, priority, version, environment, etc.

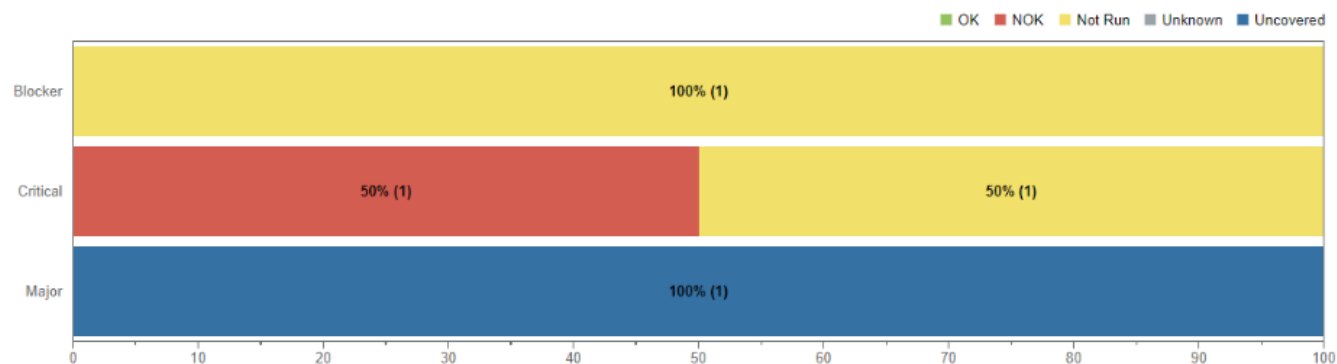
Overall Requirement Coverage Report [Switch report](#)

Scope: Version; Version: None - latest execution; Environment: All Environments ASIL Project: CAR (CAR) [How to read this report](#)



Overall Requirement Coverage Report [Switch report](#)

Scope: Version; Version: None - latest execution; Environment: All Environments Priority Project: CAR (CAR) [How to read this report](#)

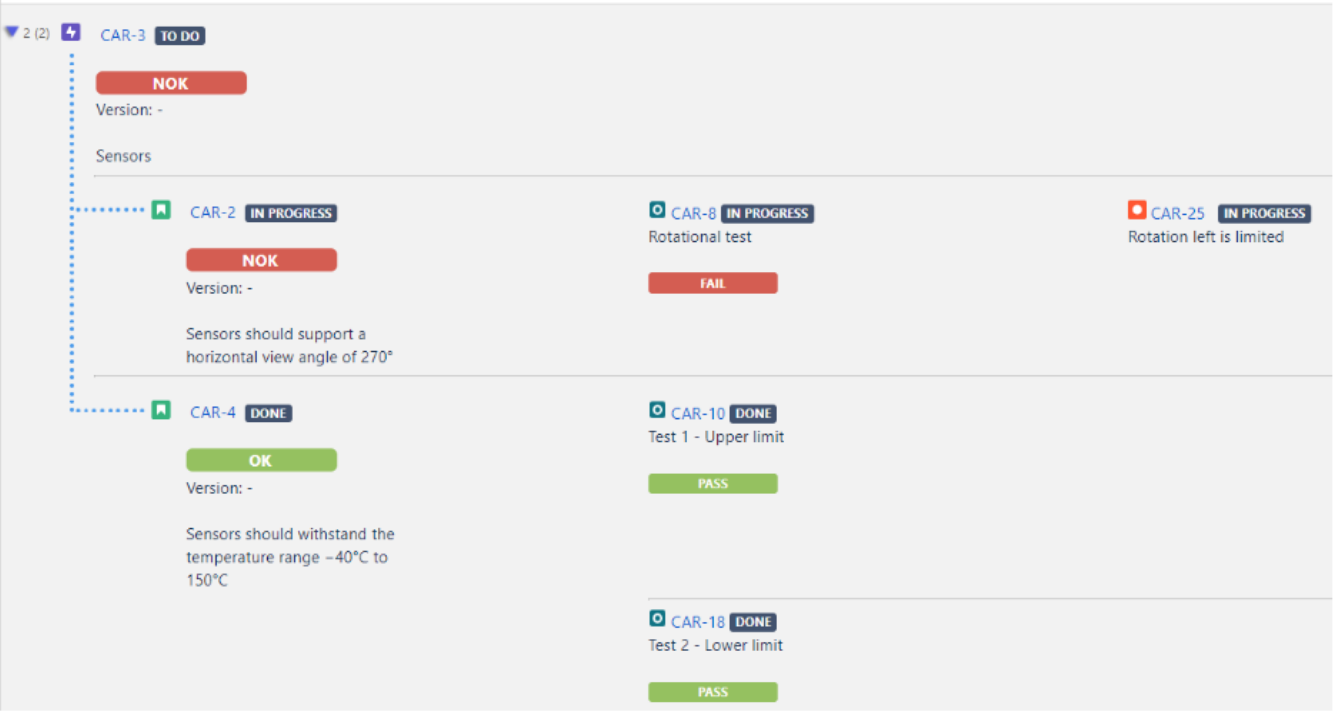


Enable audit visibility with full traceability

Xray helps answer the compliance questions by providing an overview of the status of deliverables. It provides full traceability between requirements, tests, their runs, and reported defects, including not just the relationship, but also the status of each entity.

Xray implements its test management capabilities using Jira issues. This means that all the activity logs and history are actually managed by Jira. We can even look at the code of the requirements to facilitate analysis as Jira can [connect to development tools](#). Traceability can be evaluated for a specific context (version, environment, etc.) which helps with more precise auditing and diagnosing the impact.

Another useful report, right in Jira, is the Traceability Report. We can analyze our Epics, child Story issues, show the Tests that cover them, along with the reported results and, hopefully not many, defects that may have been found. If the user story is part of an Epic, the corresponding panel from the Traceability Report will automatically show all the Tests that validate the child stories and will provide a quick overview of the Epic status, making it easier to decide whether it can be released or not.

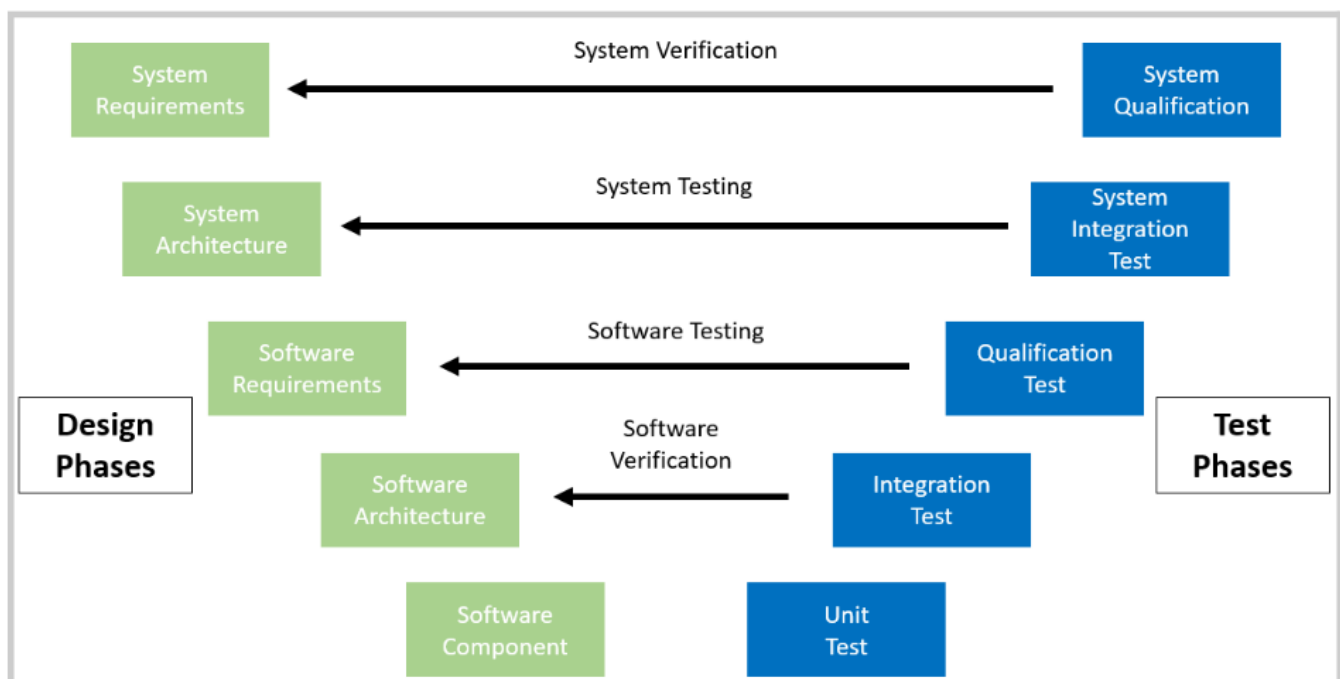


You can also handle reports focused on custom metrics. For example, one of the items for ISO 26262 compliance is estimating random failure rates over the lifetime of the device ($\text{Failure Rate} = R/T$, where R is the number of failures and T is total time). You can start by customizing a template (from scratch or from the [template store](#)) to fit your compliance requirements, with all the data to be embedded, logos, and other information (e.g., legal disclaimers). Then Xporter can use a [JQL function](#) to get a list of defect-type issues and perform the count on it (or you can consider [this snippet option](#) which is a bit trickier).

ASPICE

The abbreviation stands for “Automotive Software Process Improvement and Capability dEtermination”. It is a domain-specific variant of the international standard ISO/IEC 15504. The underlying SPICE framework is based on the ISO/IEC 12207 software development standard and is tailored specifically to the needs of the automotive industry. In general, ASPICE builds on the V-Model of software development.

The ASPICE framework consists of a process **reference** model, a process **assessment** model, and a process **capability** model. The process reference model defines a set of processes that are relevant to automotive software development, while the process assessment model provides a method for evaluating the maturity of these processes within an organization. The process capability model provides a way to measure and compare the capability of organizations in the automotive supply chain.



Given the overlap mentioned, we are going to focus on the aspects we didn't cover in the ISO 26262 section, specifically the process reference model and its validation/testing aspects.

Testing techniques

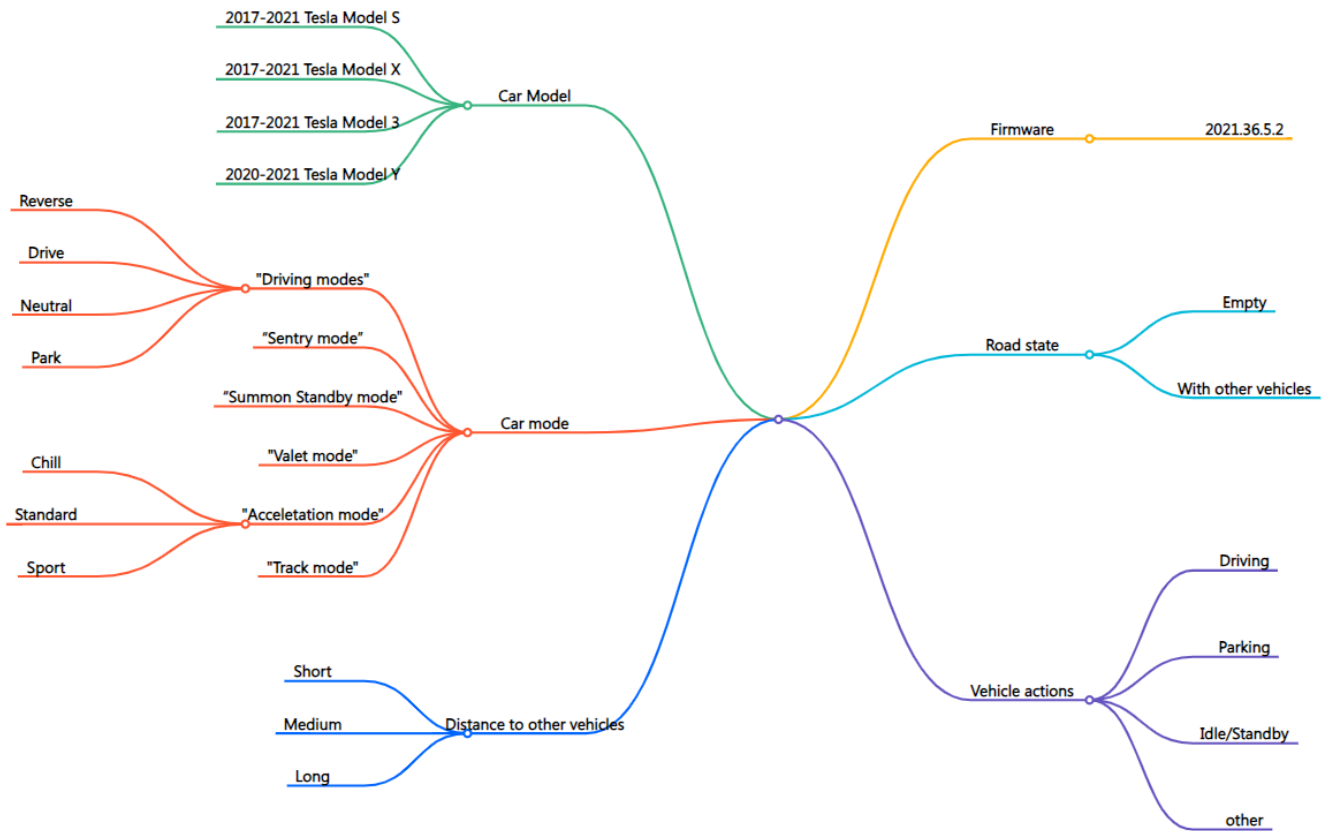
For the testing side of the ASPICE model, given the complexity of modern enterprises, all types of testing are necessary, with efficient coordination between them. No matter which approaches teams choose or the tools they prefer for test automation and CI/CD goals, it is essential to enable all of them in a frictionless way. And our suite of tools is flexible enough to do that.

Xray supports the full arsenal of techniques:

- Manual tests, including parameterized ones aligned with the dataset.
- Automated tests (Cucumber, including data-driven outline, and others)
- Exploratory testing using a desktop [Xray Exploratory App](#) (XEA).

So far, we have talked about mitigating risk by evaluating the criticality of each requirement and guaranteeing its coverage with tests. But you can also “fortify” another angle - the coverage of important data interactions within your systems.

That can be achieved with the optimized scenario generation facilitated by [Test Case Designer](#) (TCD; part of [Xray Enterprise](#)). Its **combinatorial, model-based methodology** is based on the research results about the causes of defects in production (e.g. such studies as “Estimating t-Way Fault Profile Evolution During Testing” and “Practical Combinatorial Testing”, presented by the National Institute of Standards and Technology in 2017 and 2010 respectively). It is one of the suggested techniques in [ISO/IEC/IEEE 29119:4](#) and also allows [risk-based fine tuning](#) at the parameter level.



Firmware - 2021.36.5.2	Car mode - "Driving modes"	Car mode - "Sentry mode"	Car mode - "Summon Standby"	Car mode - "Valet mode"	Car mode - "Acceleration m	Car mode - "Track mode"	Road state - Empty	Road state - With other vehi	Distance to oth - Short	Distance to oth - Medium	Distance to oth - Long	Vehicle actions - Driving	Vehicle actions - Parking	Vehicle actions - Idle/Standby	Vehicle actions - other	
																Car Model - 2017-2021 Tesla
																Car Model - 2017-2021 Tesla
																Car Model - 2017-2021 Tesla
																Car Model - 2020-2021 Tesla
																Firmware - 2021.36.5.2
																Car mode - "Driving modes"
																Car mode - "Sentry mode"
																Car mode - "Summon Standby"
																Car mode - "Valet mode"
																Car mode - "Acceleration m
																Car mode - "Track mode"
																Road state - Empty
																Road state - With other vehi
																Distance to oth - Short
																Distance to oth - Medium
																Distance to oth - Long



"Challenging Autonomy with Combinatorial Testing ([CACTus](#)) may be used by practitioners to exercise systems as part of efforts to obtain compliance with standards like ISO 21448 or UL 4600."

Execution

Once the test suite is established, you can launch the execution directly from Jira (for both manual and automated types). Regardless of the testing approach or execution type, Xray can provide visibility of testing results, including evidence, in one place for faster feedback loops.

Whenever manually executing scripted test cases, we can report results at the step level, and attach screenshots as evidence which accelerates collaboration & defect triaging process.

XEA can integrate with Xray to bring the best of both worlds: have exploratory testing evidence tracked in Jira, be accessible by the team, and reflect on the related requirements.

Hello XET App / Test Execution: HX-194 / Test: HX-189

my test to edit

Execute with Exploratory AppImport Execution Results

Return to Test ExecutionNext

Execution Status **FAILED**

Started On: 4/Dec/2019 04:30 PM

Finished On: 4/Dec/2019 06:43 PM

Assignee: **Unassigned**

Executed By: Nelson Pereira

Test Environments: -

Versions: -

Revision: -

Comment

Session Started • 18h36 | 04-12-2019
Duration 00:01:02

00:00:52 • Note [Idea]
A great note about this

Execution Defects (1)

HX-204 my defect documented

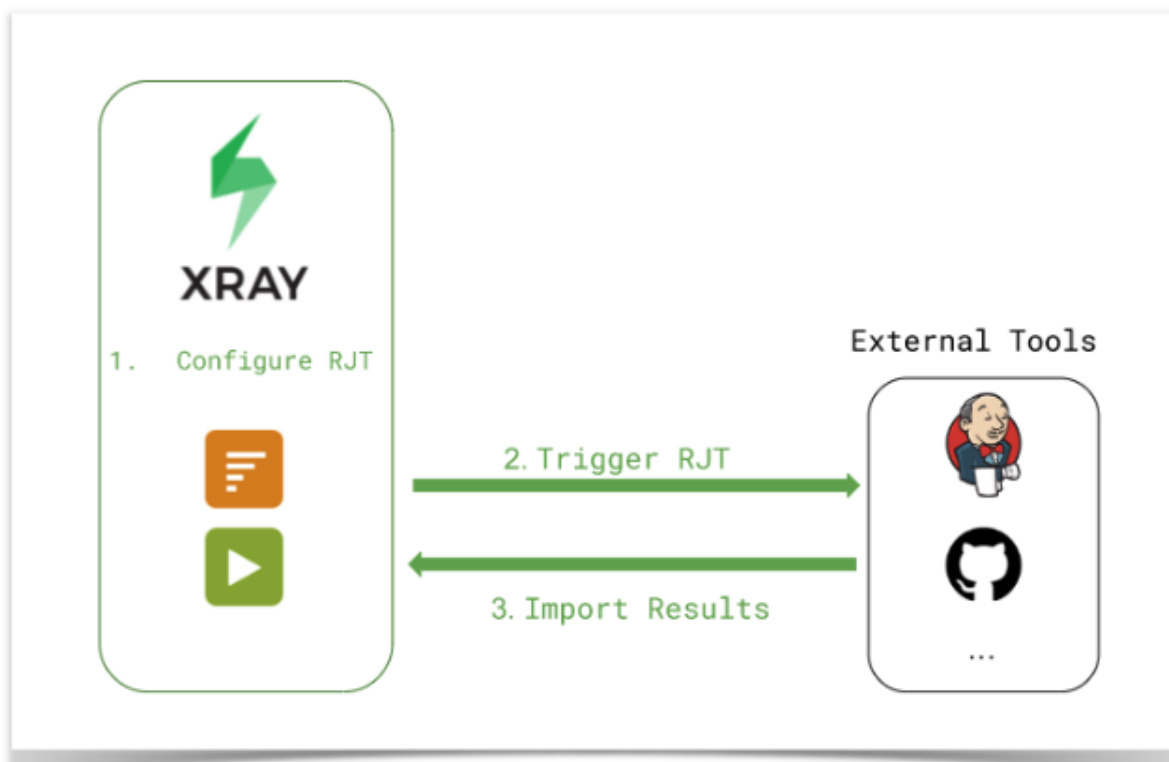
Execution Evidence (2)

xrayVideo_2019-... 266.7 kB 4/Dec/2019 06:41 PM

Test Session-D-... 2.1 MB 4/Dec/2019 06:43 PM

Execution Details

In Agile and DevOps teams, it is also especially important to enable adoption of any test automation tool/library and CI/CD tool. With the new feature - [Remote Jobs Triggering](#) - you can launch the pipeline action directly from the Test Execution issue. Then, detailed results from the automation framework can be imported to Xray and tracked consistently with other tests. You can take it a step further and link test code to requirements.



Lastly, in Xray it is possible to define several different issue types to be handled as "defects." That makes them easier to distinguish and manage - they can be classified (similar to other entities) by priority, severity, impact, and user-defined fields. They can also be covered explicitly with tests.

Issue Type Mapping

Requirements and Defects are two concepts commonly related to Tests. A Requirement represents a singular documented fault in a computer program or system that produces an incorrect or unexpected result, or causes it to behave in unintended and a Defect.

The screenshot displays a user interface for mapping issue types. It is organized into three main columns:

- Available Issue Types:** A list of ten items: Test, Test Set, Test Execution, Pre-Condition, Test Plan, Incident, Change, Sub-task, and Technical task.
- Requirement Issue Types:** A list of five items: Epic, New Feature, Story, Improvement, and Incident. The 'Incident' item is highlighted with a red box.
- Defect Issue Types:** A list of three items: Bug, Incident, and Problem. This entire column is highlighted with a red box.



A few other relevant automotive standards that we can help with are:

- SOTIF (ISO/PAS 21448)
- ISO 21434
- UL 4600
- IEEE 29119