PARASOFT®

RENOVO

Overcoming the Challenges of Safety & Security in the Renovo Automotive Data Platform

A Customer's Journey

WHO IS RENOVO?

We build an edge-centric data management platform. Our primary purpose at Renovo is to develop and deploy ADAS (advanced driver-assistance systems) at scale for our customers and help them manage all their data. We provide a turnkey solution that helps accelerate the development cycles, push through the validation cycles, go all the way into production, and ultimately, reduce the operational cost.

THE RENOVO AUTOMOTIVE DATA PLATFORM

Data and data management are essential to modern automobile technology. To this end, I'll introduce you to the platform, explain how it works, and then talk about some of the data requirements in today's automotive industry.

Let's drive right into the automotive data platform. We like to think about it as five key elements of the data platform.

It's **edge-centric**. I mentioned this already. We think of the vehicle itself as part of edge compute technology as well as actual edge-based devices that are either co-located on premise or relatively close to your development facilities.

It's **intelligent**. When we bring the data in, we're not just doing file management. We're managing at the channel level, at the metric level, or the object level. And we can filter that data, index that data, and enrich it.

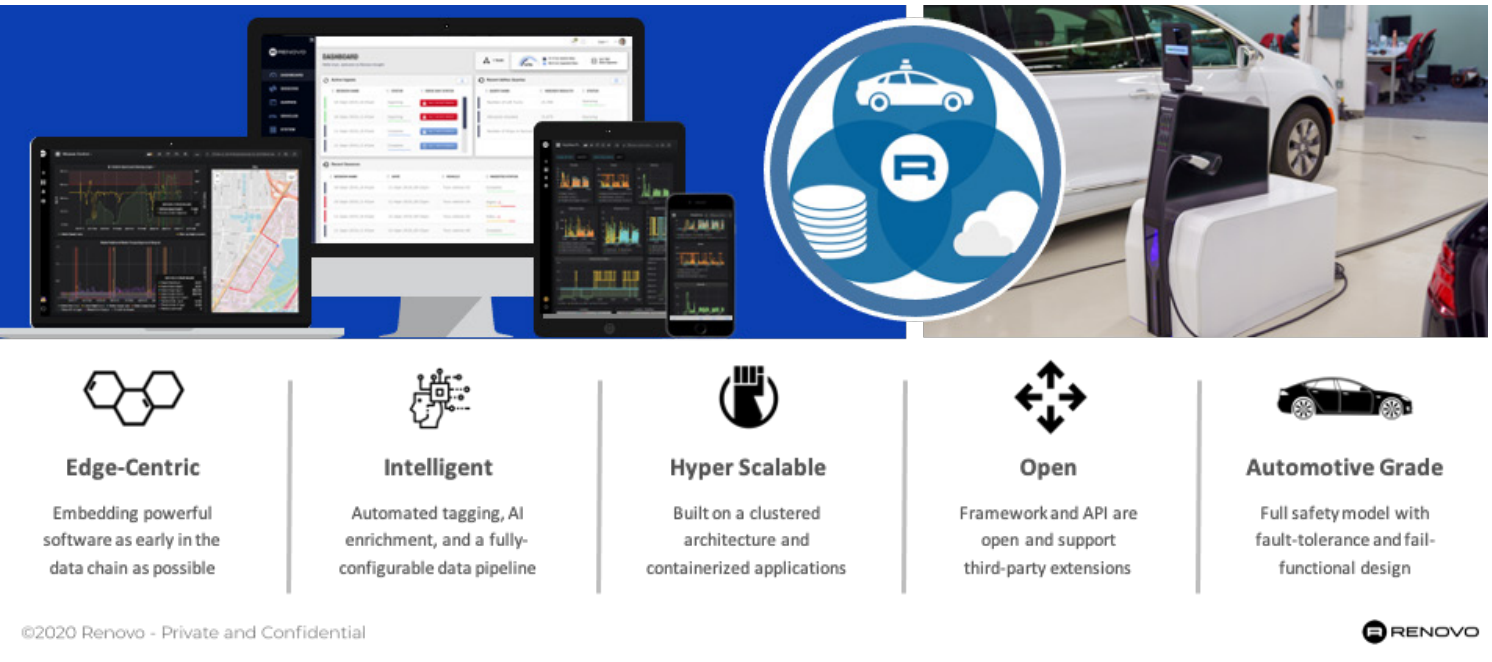


Figure 1: The Renovo automotive data platform.

It's **hyper scalable**. Built on a clustered architecture, we can scale your small development fleets of four to 500 vehicles and all the way up to tens of thousands, or even hundreds of thousands of production vehicles.

It's an **open system**. The framework and the API are open. We support a lot of third-party providers into the ecosystem. At Renovo, edge devices are not just an appliance sitting next to the vehicle. We consider the vehicle itself as part of that edge compute. So, software must be able to co-exist and live on the vehicle with other safety critical applications.

It's **automotive grade**. The platform has a full safety model with a fault-tolerant and reliable design — and must apply all the standards that are required to go into a vehicle.

DATA DRIVES NEXT-GENERATION AUTOMOTIVE SYSTEMS

Why is data important to automotive systems? For feature development, first and foremost. All our engineers and all of our OEM engineers try to develop new ADAS features — from lane keeping to adaptive cruise control. Building and improving such systems requires test cases and scenarios. Data is vital for finding any object that was not able to be classified by a vehicle. For example, capturing camera frames off the vehicle, identifying them, and understanding them is important.

At Renovo, our view of the automotive world is a big data loop. For people that are generating or creating the data, there's a need to store it in a database, and ultimately deliver it to several different services — including back to more creation phases.

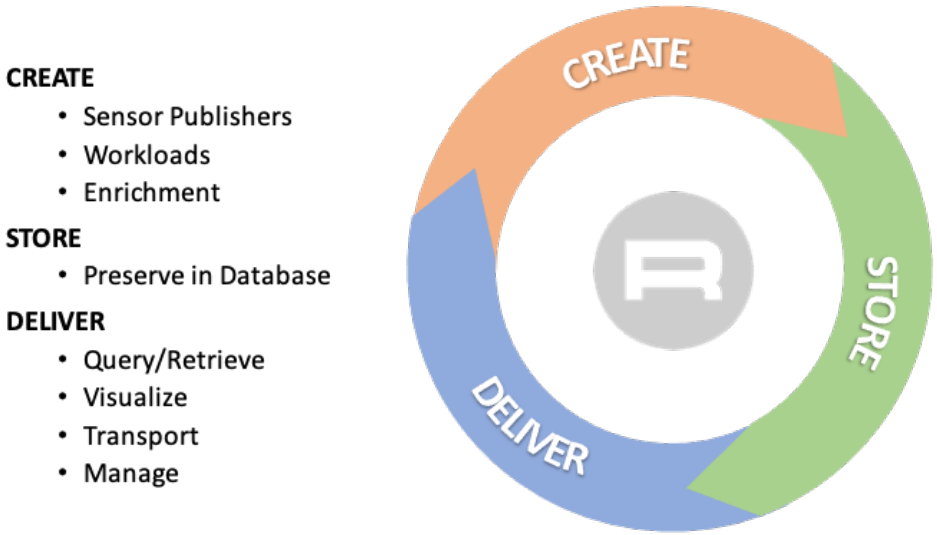


Figure 2: Renovo's approach to automotive data management.

Sensors and devices on vehicles create data that's stored into these databases on the vehicle itself. Sophisticated sensors like cameras and LiDAR are also generating very large frames or point clouds, so those are going to the blob (binary) storage. Then there are services that transport data off the vehicle. Typically, the off-vehicle storage is the cloud. It could also be an edge-based core computing device next to the vehicle that stores the data into similar databases off the vehicle and is able to generate the samples to them. The data pulled off the vehicle can be visualized, queried, and managed.

One thing of note is that this entire architecture is a microservices architecture, which means each of these boxes is its own individual application that can be instantiated — instantiated multiple times, in fact. This means that the architecture can be distributed across multiple compute nodes, services can be easily reconfigured without affecting the entire system, and the same service can be instantiated multiple times to handle different data pipeline needs in the same compute environment.

VALIDATION AND COMPLIANCE ARE CRITICAL

Validation is critical in such an important application. Being able to ensure functional behavior, and even more importantly, being able to verify adherence and compliance to requirements and standards.

Verification and validation produce data needed to document compliance with requirements and standards. Data integrity is paramount since the data used applies to many safety-critical functions. In addition, data storage and retrieval are important for dealing with liability. The data generated can rebuild a scenario — a labyrinth event almost down to the milisecond level. We've seen it done in several high-profile cases and we were able to understand exactly what happened.

What's more is that this level of data collection creates a unique position in terms of the behavior of both the vehicle and the consumers. That data is extremely valuable to a lot of different industries whether they're municipalities or commercial entities. This information is valuable and it's going to change the way that we think about automobiles and driving — and build a lot of interesting business models and revenue opportunities.

THE PROBLEMS RENOVO FACED

Development of an automotive data platform of this scale is a large task. It isn't like the data systems of the past, whether a simple telemetry system, what you see on an OBD bus, or an infotainment contact list. This next generation data platform is vitally important information for automotive software developers and legal and business teams. It requires a higher level of quality, reliability, and security.

Internally at Renovo, we use the analogy of an aircraft black box. We're at the point where data systems are as important and require the same hardened systems that are safe, reliable, and secure. We decided to ensure our automotive application platform met the requirements of ISO 26262. There were challenges.

ISO 26262 COMPLIANCE LOOKED LIKE A BIG HILL TO CLIMB

Our data platform requires ISO 26262 compliant development. Verification and validation are required to make sure our software works correctly and to ensure data integrity. We needed to apply the same quality, security, and safety concepts to the data collection, storage, and transfer as any other safety critical software in automobiles. Our software is used to monitor safety related events and responses and to drive the development of other automotive systems.

FITTING COMPLIANCE INTO OUR DAY-TO-DAY WORKFLOW

Modern software development requires Agile techniques with continuous processes, integration, and deployment. We needed to be able to build safety and security into our process without it detracting from functional requirements and slowing the team down. It was important for us to integrate the needed tools and techniques into our developers' day-to-day work with the current tools in use and processes in place. Compliance requires adherence to sound software engineering practices, strict traceability, and documentation of critical decisions and results. So, we needed the ability to create and maintain traceability and an audit trail.

LACK OF VISIBILITY INTO QUALITY, SECURITY, AND COMPLIANCE

A constant struggle in managing software development was deciding where best to assign resources. It was also difficult to understand progress toward our compliance and quality goals. We're under strict guidelines in terms of coding standards, code coverage, and verification and validation traceability. Data helps us understand where we stand on a weekly or even daily basis. We need visibility into the state of our projects so that we can decide where to concentrate resources. It's also essential to understand the status of validation and ISO 26262 compliance.

FINDING A PARTNER

With these requirements, we looked to a partner that could help us solve the issues at hand and reach our goals to deliver on schedule and meet ISO 26262 compliance. In this case, our partner is Parasoft and their tools are critical to Renovo to develop platforms in a way that is safe, reliable, and high quality. We start validation early in the development phase and avoid bolting on safety and reliability at the end of the cycle. With Parasoft, we build it in throughout the entire development cycle.

The dashboards and the traceability that their tools provide are amazing. It helps us zero in quickly without having to search and figure out what the static analysis tool is trying to tell us about what is happening. This level of information, in an easy-to-navigate form, is super valuable to me as a manager to understand how the team is doing and anticipate what is coming down the road

HOW PARASOFT HELPED RENOVO OVERCOME CHALLENGES

Having experience with other automotive manufacturers and ISO 26262 compliance, Parasoft brought their experience helping other companies adopt a quality (and security) first approach to development. They helped "shift left" Renovo's testing while creating and maintaining the traceability and code coverage audit trail needed for compliance. The following outlines the general approach Parasoft took to help us achieve success on our automotive data platform.

ADOPT A PREVENTION STRATEGY: SHIFT-LEFT QUALITY AND SECURITY

Parasoft provides technology that helps development teams at each stage of the software development process build safety and security into the application. One of the things to remember is that if your application is unreliable, it's insecure by definition. Unreliability within a system is the exploit that somebody will take to compromise your system, compromise the data, or compromise the integrity of its operation. There are a couple of different strategies that you can take when it comes to identity and security issues within an application.

Parasoft helped Renovo improve our quality and safety initiative by adopting a prevention strategy. In other words, stopping the software bugs (weaknesses or future vulnerabilities) from being injected into the process or the code in the first place via detection and prevention, along with following good design and coding practices to avoid possible reliability, security, and safety issues.

Static analysis is the first part of building code that's safe, reliable, and compliant with ISO 26262. Parasoft static analysis engines for a C, C++, Java, and .NET code, bring detection and prevention together. However, these tools need to do more than create reports. And although important, it's more critical to successfully adopt these types of quality practices into a team-centric workflow in your development process.

IMPLEMENT A TEAM-CENTRIC ISO 26262 WORKFLOW

Parasoft encouraged us to assign a person (or small team) to oversee overall compliance within our organization. The responsible person or team defines the set of checkers (static analysis rules) and configurations (sets of checkers) to validate against the appropriate standard.

The tool configurations map to specific standards (such as AUTOSAR C++14 and MISRA C/C++ for ISO 26262, SEI CERT C for security) and any organization rules that apply. They're shared across the development team in the build process and with the developers across their desktops directly in their IDE. This prevention-oriented workflow is illustrated below.

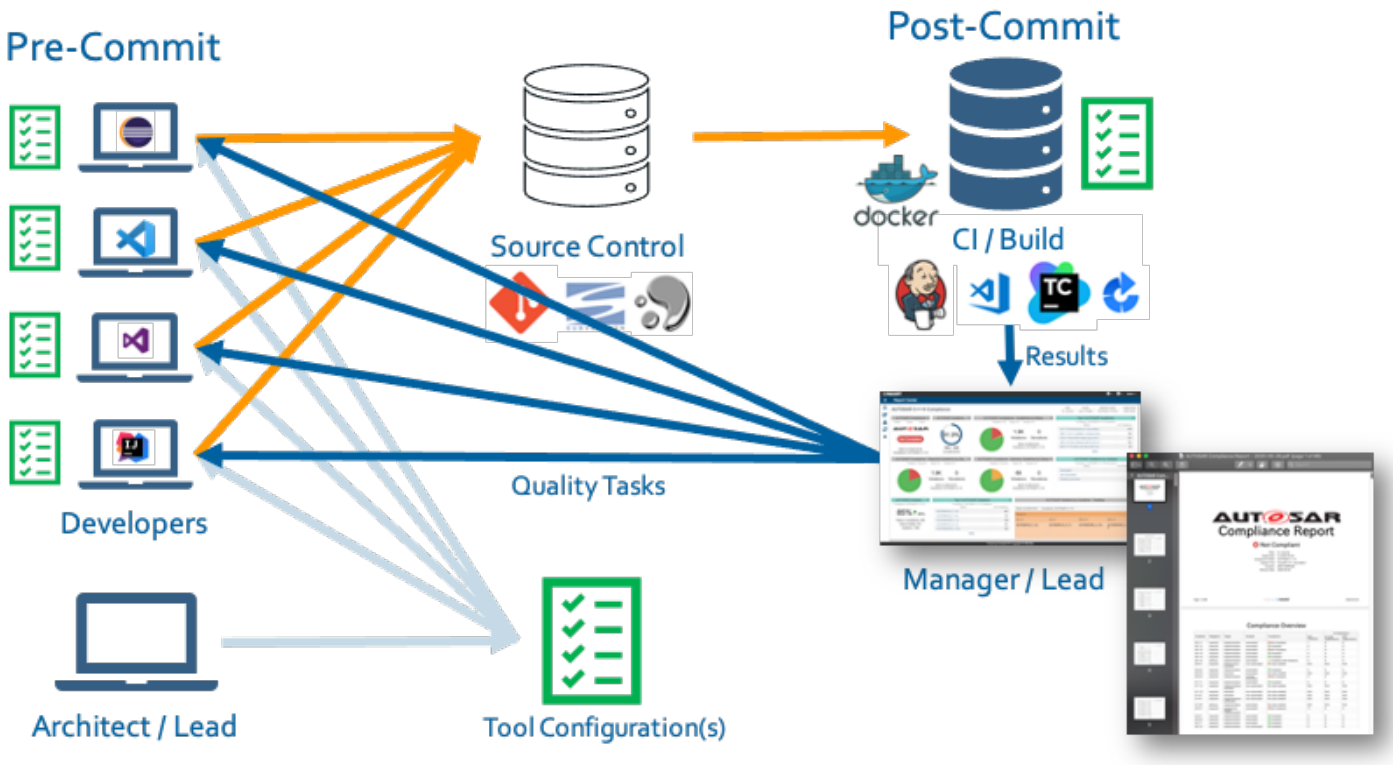


Figure 3: Parasoft team-centric workflow for compliance.

Moving to a focus on prevention means better quality before code is checked into source control. This was one of the key mechanisms to ensure that Renovo was building quality into the application rather than just trying to find the problems later in development and testing. This approach helps us develop behaviors that instill best practices, like coding patterns, in compliance with the standards.

We adopted a workflow, like the one illustrated above, with a pre-commit step that requires a static analysis check before check-in and a post-commit step that acts as a safety net. It's the full analysis built into the CI/CD process. This deeper analysis discovers complex issues that cross file, class, or module boundaries. Further along the post-commit phase, the results from the CI/CD process are used for the dashboards and reports for a quick view of what's going on within the quality of codebase and standards compliance.

AUTOMATE TESTING TO ACHIEVE COVERAGE AND TRACEABILITY

Renovo needed a test automation solution that did more than functional testing. We needed a solution that tracked coverage and requirements traceability while helping improve both. Parasoft provided testing and validation of the software to meet the defined requirements for functionality, safety, and security — and enabled us to automate acceptance testing where appropriate.

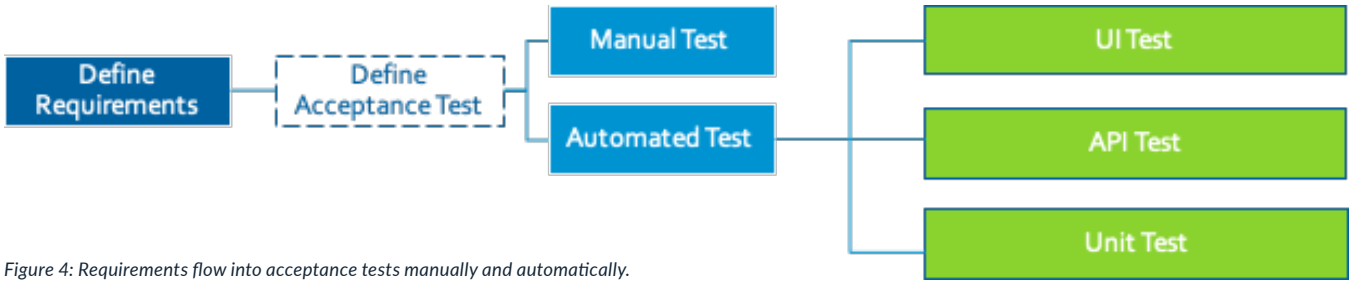


Figure 4: Requirements flow into acceptance tests manually and automatically. Automated testing applies at different levels of complexity starting at unit testing.

With Parasoft, we had the option to generate work items in requirements management and agile planning systems like codeBeamer, Polarion, and JIRA. These work items require tests to verify the functionality, validate that they are functionally correct — and keep moving up the higher levels of the V-model from a development perspective.

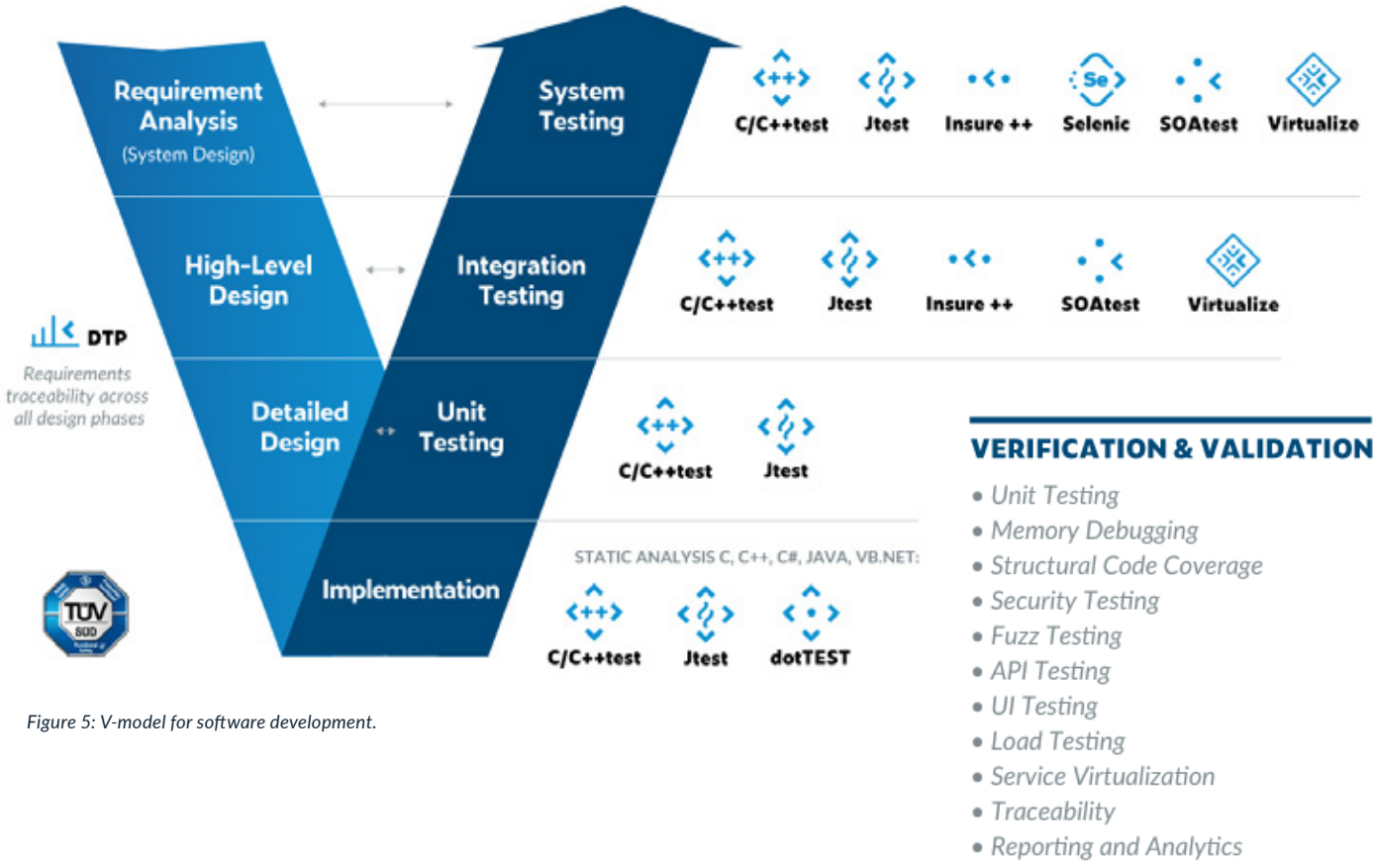


Figure 5: V-model for software development.

At the lowest testing level are unit tests, which validate that the code complies to the low-level module requirements. Testing moves up a complexity scale — from unit to integration testing. This is demonstrated on the right side of the "V" in the V-model for software development.

Automated API testing is a great way to be able to validate those functional modules at the integration level. This approach is especially important with the type of a cloud-based architecture like Renovo's microservices-based platform.

INCREASE CODE COVERAGE WITH TEST AUTOMATION

A critical aspect of API — and integration testing in general — is creating a test environment with enough test variation to maximize test code coverage (amount of code tested) and functional test coverage (number of tests completed.) This requires satisfying dependencies for systems unavailable or impractical to use for integration testing.

Service virtualization emulates the backend system dependencies that allow for modeling different scenarios and performance characteristics. This approach will enable Renovo to shift left our integration testing. It makes it possible for us to imitate missing or incomplete components and simulate scenarios that are otherwise difficult to achieve in a production environment.

Being able to automate unit and integration testing with variations of test data and scenarios is important for achieving the testing and code coverage required for rigorous processes like those needed for ISO 26262 compliance.

ISO 26262 calls out different types of structural code coverage. At the unit level (depending on criticality) it requires statement, branch, and modified condition/ decision coverage (MC/DC), which is an intensive type of coverage reserved for ISO 26262 ASIL D.

As testing moves toward the integration and system, tracking this granular level of code coverage is difficult without tool support. As different forms of testing are performing, it's essential to aggregate the coverage data across these different testing practices. The true coverage information is only available with a centralized aggregate view, which the Parasoft reporting and analytics dashboard provides.

AUTOMATE BIDIRECTIONAL REQUIREMENTS TRACEABILITY

Requirements in safety-critical software are the key driver for product design and development. These requirements include functional and non-functional requirements that are used to fully define the product.

This reliance on documented requirements is a mixed blessing because poor requirements are one of the critical causes of safety incidents in software. In other words, the implementation wasn't at fault, but poor or missing requirements were. It's important to realize that many requirements in safety-critical software are derived from safety analysis and risk management.

The system must perform its intended functions, of course, but it must also mitigate risks to greatly reduce the possibility of injury. It's also important to document and prove that these safety functions are implemented and tested fully and correctly. Traceability is critical.

Renovo needed a way to maintain traceability at scale. Although application lifecycle management (ALM) tools are the hub for traceability, Parasoft provides the verification and validation of requirements with an automated bidirectional traceability to the executable test case. This includes the pass or fail test results and the traceability down to the source code that implements the requirement.

As shown in the image below, each of Parasoft's test automation tools (C/C++test, Jtest, dotTEST, SOAtest, and Selenic) support the association of tests with work items defined in these systems (such as requirements, stories, defects, test case definitions). Traceability is captured and presented through Parasoft's central reporting and analytics dashboard (Parasoft DTP).

Figure 6: Parasoft provides bidirectional traceability from work items to test cases and test results - both displaying traceability reports with Parasoft DTP as well as reporting results back to the requirements management system.

Parasoft DTP correlates the unique requirement identifiers from the originating ALM system with static analysis findings, code coverage, and test results from unit, integration, and functional tests. Results are displayed within traceability reports and sent back to the requirements ALM system.

These systems provide full bidirectional traceability, a traceability matrix, and reporting as part of their core functionality. Maintaining this bidirectional correlation between requirements, tests, and the artifacts that implement them is an essential component of traceability.

VIEW DATA QUALITY IN ONE PLACE

Renovo didn't have sufficient information to make management decisions on our compliance and quality goals. We also lacked visibility of progress towards those goals. Parasoft's reporting and analytics dashboard aggregates data from across all the different testing practices including static analysis (with AUTOSAR or MISRA coding standard compliance) and coverage and traceability from unit testing, API testing, and system level functional validation.

Parasoft's dashboard gives us a centralized view of quality offering us a real-time audit on the process and visibility into the status of compliance with ISO 26262. This analysis also generates the documentation needed to demonstrate compliance to an auditor.

Figure 7: Parasoft DTP centralized view of quality aggregated across all the tools and phases of development.

We can customize dashboards to provide different views of collected data in one centralized place and gain insight into the quality. One view is static analysis results. It shows the level of compliance to a selected coding standard.

Clicking on high-level graphs gives us in-depth information about the identified violations in files and methods. The graphs reveal important details about the source code, stack traces, and test results. A truly multi-dimensional risk metric, the dashboard shows how risky or vulnerable different classes, functions, and files are within our codebase. Everything is determined by runtime analysis, code coverage, and static analysis results.

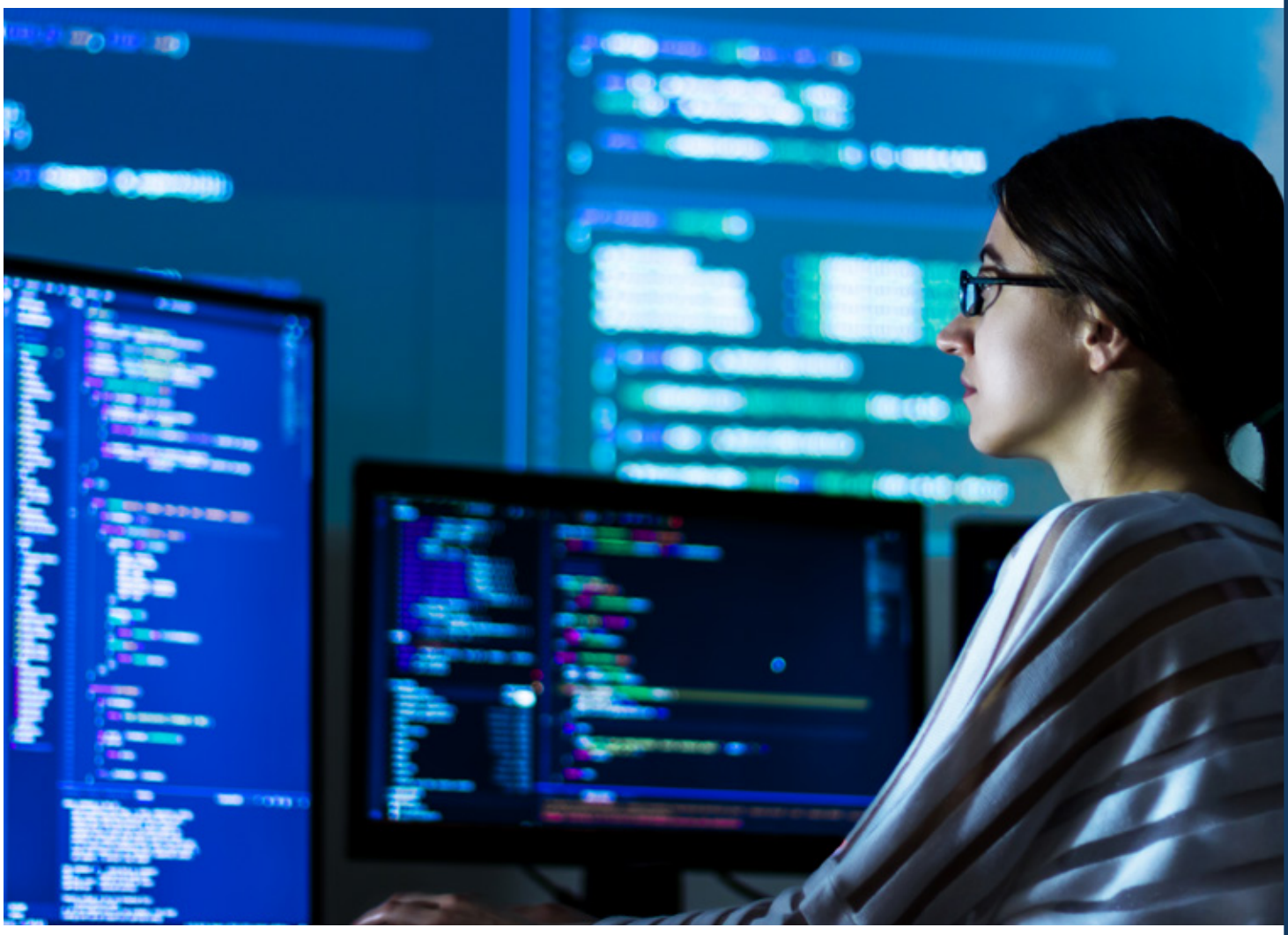
This rich centralized view provides our Renovo management team with the insight to assign and prioritize work and effort based on visible analysis results. We're not guessing where our high-risk issues exist. The dashboard clearly lays it all out. We can see the progress over time as our team corrects the issues. This makes project planning and scheduling activities much easier and aids us in time-to-market projections.

AUTOMATE TOOL QUALIFICATION

Developing safety-critical software like Renovo's data platform requires that tools used during development meet the same levels of quality as the software being developed. Parasoft makes this easy for us because Parasoft C/C++test is certified by TÜV SÜD as suitable for use when developing for ASIL A to ASIL D safety-critical automotive software. This saves us concern, time, and effort.

If you happen to need it, there's an option to perform a qualification by validation using Parasoft's automated tool qualification kit. The Qualification Kit automates the tool qualification process according to the method, "Validation of the software tool".

A dedicated qualification support tool guides you through all the steps required to qualify the tool, automating most of the tedious manual work. That includes executing the test cases from the exhaustive test suite provided with the Qualification Kit and generating the final reports required to document the qualification process.



SUMMARY

Renovo approached Parasoft to seek help to reach our quality, security, and ISO 26262 compliance goals. Delivering our next generation product while also complying with the rigorous processes required for safety-critical software seemed a daunting task. We needed a test automation solution that did more than help us run tests. We needed a solution that automated compliance requirements and fit in nicely with our current workflow, tools, and processes. We also needed tools to keep and maintain traceability and code coverage with an audit-compliant documentation trail.

Parasoft helped Renovo move toward a prevention approach. We used static analysis early during development to improve the quality and security of our code. It also helped us ensure coding standard compliance required by ISO 26262.

Preventing bugs and vulnerabilities earlier in the lifecycle saved Renovo time and money. Parasoft integrated this approach into our workflow at the develop desktop level and in our CI/CD pipeline. Once the solution was in place, the rest of Parasoft testing tools were made available to the team and became part of the day-to-day activity of developers and testers.

It was important to Renovo to have test automation tools that maintained requirements traceability and code coverage information across all our testing activities. More than that, we wanted tools to help increase our test and code coverage over time.

Parasoft provided a testing platform to aggregate test results and code coverage, and the ability to create unit and integration test environments to increase coverage. This aggregation of traceability, test coverage results, and static analysis offered the kind of data Renovo required to understand just where their development stood in real time.

Parasoft's dashboard management tools continue to empower Renovo to make the right decisions and determine where to concentrate their development and testing efforts.

TAKE THE NEXT STEP

Build quality, reliability, and security into your software development process from the beginning. Talk to a Parasoft expert today.

If you're interested to know more about Parasoft automotive software testing and its benefits, reach out to Parasoft and Renovo.

ABOUT PARASOFT

For over 30 years, Parasoft has been providing innovative tools that reduce the time, effort, and cost of delivering secure, reliable, and compliant software by integrating static and runtime analysis; unit, functional, and API testing; and service virtualization. We help automotive organizations succeed in today's most complex development ecosystems and initiatives — real-time, safety-critical, secure, Agile, continuous testing, and DevOps.