

INTLAND
SOFTWARE

navya



Navya Customer Case Study

About Navya

Navya specializes in the development of autonomous driving systems. Founded in 2014 and headquartered in Lyon, France, it employs over 280 employees (140 of which are engineers and technology experts) across sites in France and the US. Navya was the first company in the world to release a commercially available self-driving shuttle. The company's first fully autonomous, driverless, and electric AUTONOM® SHUTTLE was launched in September 2015 as a first and last-mile passenger transport solution.

Today, Navya has over 160 self-driving shuttles operating in more than 20 countries. The company's goal is to become the leading player supplying autonomous driving systems for passenger and goods transport.

In 2018, Navya launched a project to find development tools that could help the company manage the complexity of delivering high-tech autonomous driving solutions. Jules Garbé, Navya's Head of Product Development (responsible for system specification but also development process improvement) joined the company just as Navya was wrapping up the ALM selection process that led to the rollout of codeBeamer ALM. Jules' team is responsible for delivering a single interface for all hardware, embedded software, and cloud-based software development activities in the company, and for ensuring the coherence of the overall product architecture.



The problem: Agile & ISO 26262 compliance

In early 2018, Navya's primary challenge was handling product development complexity in a fast-growing organization. They needed a way to balance the flexibility of Agile and the rigidity of the V-model while ensuring compliance with safety obligations.

This was the company's startup era: the dev team didn't have a system in place to manage product specifications, and requirements weren't frozen while development was ongoing. Software teams were using Agile, with each team managing its own backlog. There was insufficient synchronization between teams, no procedures to follow, and no visibility into the process of development. All this was bound to conflict with the stringent regulatory requirements that govern functional safety in the mobility industry.

Before making the transition to codeBeamer ALM, Navya's dev team was using a mix of tools similar to the toolsets of so many other organizations. They used Git for versioning code and relied on a combination of manually updated spreadsheets, Jira, and a constantly growing document repository to stay on top of development activities.

As Navya's teams were growing and product complexity was on the verge of starting to spiral out of control, the team saw risks that could have had a negative impact on development efficiency. The team didn't wait for the situation to become so bad that it would affect operations. It was obvious that a methodological change was needed.



Jules Garbé
Head of Product
Development, Navya

Jules heads the product team at Navya, responsible for ensuring technical coherence between hardware and software development for the company's products.

As a technology expert with genuine curiosity, he has worked in several industries. After spending 6 years in the aerospace sector, where Agility has been a banned word for decades, Jules transitioned to the fast-moving robotics industry. He helped design humanoid robots from scratch to solve the problems of an aging population in Japan. Jules then joined the e-mobility revolution by starting a company to develop the electrical charging infrastructure that enables the transition to electromobility.

Jules joined Navya in 2018 and was involved in the deployment of codeBeamer ALM in this Agile organization. As the manager responsible for the overall product architecture, he oversees process improvement in the entire product development lifecycle including demand and requirements management, development management, and validation management. Jules was the primary contact providing input for this case study.

“As a matter of fact, transformation is our day to day job. It's part of our DNA, it's what defines our organization and products, and ultimately it is what helps us stay ahead of the market.”

– Jules Garbé, Head of Product
Development, Navya

“We needed to tool up, formalize our processes, and made the decision to support our development efforts with an Application Lifecycle Management tool. We had to quit using just Git, spreadsheets, and email to follow up. It was time for us to industrialize our processes.”

By introducing ALM, Navya’s dev team was able to reconcile the Waterfall model that is still, in effect, required in automotive development, and scale Agile through the Scaled Agile Framework (SAFe®).

Safety & compliance in autonomous development

Navya develops a self-driving product with multiple safety-critical subsystems. As governing bodies are lagging behind product developers in delivering regulations for autonomous systems, ISO 26262 is still the most important regulatory requirement that Navya has to comply with in order to ensure functional safety.

From a process perspective, the dev team runs high-level risk analysis based on product use cases. Those are cascaded down into system requirements which are propagated to the required subsystems and components. To validate functional and dysfunctional requirements, the team conducts testing at the function or simulation level, stressing the software system in a controlled environment and then at the system level in a representative environment.

Once all these tests are passed, the product is validated and may be deployed – yet even after all this, the Navya team starts with a smaller, more controlled deployment with a limited number of customers. If that’s successful, the newly developed solution is deployed to the rest of the fleet.

“Process is one of the keys to ensuring that our software respects safety regulations. Obviously, it is our main goal to ensure ISO 26262 compliance. For that, codeBeamer contains the right set of features that provides us with the required traceability between different levels of specifications.”

Evaluating ALM tools with flexibility in mind

In July 2018, Navya initiated a sourcing process to select an Application Lifecycle Management solution. The search was fairly rapid due to a growing need to formalize development processes. In order to make an informed decision, the team analyzed the capabilities of several tools and started a discussion with multiple vendors.

Navya Customer Case Study

The main factors that the Navya team was looking for as part of the evaluation included:

- The ability to **custom-configure various types of objects** and to establish links between them for **end-to-end traceability**
- **Ease of use** in a fast-growing, innovative, Agile environment
- **Collaboration** support and an ability for users to interact within the tool
- Flexibility and the **ability to evolve and adapt the system** to support continuous improvement
- **Ease of configuration** as the team's needs would be constantly changing as processes were improved
- **Pricing** model for optimal ROI

Like in software development itself, Navya's development team took an Agile approach in building out their ALM environment. They wanted to proceed incrementally, following fundamental Agile patterns in the process to make sure the solution suited their evolving needs. The team sought maximum efficiency by continuously improving their working environment. Therefore, flexibility was a key requirement for Navya: their tool of choice had to be able to adjust to and support a fully Agile mindset.

Having conducted the assessment process, Navya narrowed the search down to a shortlist of finalists that included Siemens Polarion and other tools. Eventually, it was this crucial need for flexibility that tipped the scale in favor of codeBeamer ALM.

“codeBeamer's flexibility in setting up the environment and around evolving objects were the main arguments in the final stage decision.”

Navya purchased a handful of licenses in August 2018 and continued to progressively expand their number of seats. In 2019, there were 120 users of codeBeamer ALM at Navya, with plans to reach 200 the next year.



Self-governing rollout & training

Navya had no previous experience in the scaled rollout of a development tool, and according to Jules, they hadn't anticipated the success and growing demand they encountered with codeBeamer ALM. Overall, Navya reported fast adoption of the tool, and very little effort was required to train users.

First, the goal was to use the platform for software bug management, which was shortly extended to user story and epic management for the software team. One surprising aspect of Navya's experiences with the system came when the tool's use was adopted by the hardware team.

Having used codeBeamer ALM for some time, it felt natural for the product team to expand the tool's use to hardware development – but management faced some resistance as the platform was perceived by hardware engineers as a software-only tool. However, it soon became very clear that codeBeamer ALM was able to manage hardware problems and to reduce complexity, and so the hardware team's buy-in was secured.

In terms of training, Navya took a hybrid approach: they did have formal training in place but saw more value in just letting engineers “play” with the tool. Once they learned how easy to configure codeBeamer ALM was, Navya decided to help spread the use of the tool organically by just handing out admin rights to many teams.

This approach proved to be a double-edged sword: on one hand, engineers did become adept at using and configuring the platform – but Navya's dev team ended up with tons of unmaintained and heterogeneous objects that just couldn't work together. To resolve this, they set up a team tasked with maintaining a systemic approach to using the platform, cutting objects, and reintegrating them under a shared framework.

“We conducted several training sessions prior to deployment and had multiple presentations and all sorts of communication about the upcoming changes. But to be honest, we saw a lot more success just letting users organically train each other.”



Benefits of integrated ALM

By switching to codeBeamer ALM, Navya primarily benefited from transparency, clarity, and consistency in their processes.

“Providing our company with the right tooling which helps us manage complexity is definitely an enabler that helps take our innovation to the next step. That’s where having the right tool can be a real differentiator.”

Before implementing codeBeamer ALM, Navya’s teams were working in silos on various functional aspects of the product. Visibility into their activities was limited, and it was difficult to create a system-level view that would allow the team to handle complexity.

With codeBeamer, they are able to have an end-to-end view of development activities with links between all kinds of artifacts from requirements through to validation. Dependencies are easily tracked, and common-ground policies help standardize the quality of coding. The team can easily monitor progress and various high-level metrics to measure areas of focus. This helps spot problem areas to fuel continuous improvement.

“codeBeamer has provided us with a single interface to link and manage all sorts of objects, all of which are required to develop a highly complex product. The platform enables us to do this in a user-friendly and very flexible environment, linking all the functions of the company in a coherent manner.”

ALM helped smoothen the compliance aspect of Navya’s development efforts, with traceable links going from high-level requirements through the organic architecture and back to final testing. This supports Navya’s team in conducting the necessary validation steps and compliance checks. Prior to making the switch to codeBeamer, these processes were not tooled, and Navya reported that using the platform had a beneficial impact on the team’s efficiency.

Considering the flexibility of the platform and all the benefits it provides, it’s no surprise that Navya is now investigating novel ways of adopting codeBeamer ALM in the organization:

“We haven’t yet finished exploring the true potential of codeBeamer as we are in continuous improvement in our ways of working. We’re expanding the tool’s use to other areas of the company. While this project was started to deal with development complexity, focus is now shifting toward managing all activities in our company such as ticketing for development, risk management, corporate management, project management, product roadmap management...”

About Intland Software

Intland Software is the developer and vendor of industry-leading software tools for the delivery of safety-critical technology. The company's flagship product codeBeamer ALM is a fully integrated end-to-end Application Lifecycle Management software platform with regulatory compliance support. The recently released Intland Retina provides all-in-one requirements, risk, and test management capabilities with digital quality management and audit control support for regulated industries. Intland Software's tools help reduce the time, risks, and costs of product innovation and regulatory compliance. Follow @intland on Twitter, [LinkedIn](#), or find out more at <https://intland.com/>.



Start a free trial of codeBeamer ALM:
<https://intland.com/codebeamer/download/>

To request a free 1-on-1 online demo, contact us:
contact@intland.com