exalate

How a Construction Tech Company Implemented Exalate to Maximize Workflow Efficiency

Table of contents

Nevaris software has been in construction technology solutions for over a decade and covers all aspects like financial accounting, ERP, extensive construction wages, etc.

One of their products, Nevaris Build, is used for construction management and supports all functions and data for your entire construction project in a single software.

The Struggle

The fundamental issue Nevaris Build faced was unclear and ineffective communication between different teams working on the software.

For instance, the dev team was using Azure DevOps, but the Quality Assurance (QA) team wanted to keep working in Jira for test cases and release management.

So they wanted to look for a solution that allowed the dev team to work in Azure DevOps but still have the important data reflected in Jira for the QA team.

We reached out to David Müehlenhoff, the development team lead at Nevaris Build, to talk about his experience with Exalate.

Why Exalate?

- Streamlined collaboration between teams
- Enhanced customization capability through scripting engine
- Flexibility to adapt different workflows
- Cost efficiency and reasonable price
- Adapters for other systems like Salesforce

> *“We evaluated different products and Exalate was the one. It was also the most cost effective overall.”*
> *David Müehlenhoff, development team lead at Nevaris Build*

Challenges

- Inefficient collaboration between different departments
- Siloed Information flow across teams

Initially, most of the teams worked in Jira and each department had its own issue types, tasks, rules, and boards. But there wasn’t a single interface that could give an overview of where the entire team, as a whole, stood.

Further on, the tools changed and the development team switched to Azure DevOps. This increased manual work of copying and sending information between Jira (QA team) and Azure DevOps.There were almost 30 tickets that were copied on a daily basis.

The dev team needed visibility on the test cases and releases that were planned and executed by the QA team, whereas the QA team needed to see the dev team’s progress.

The biggest issue Nevaris Build faced was information flow. Every team wanted visibility on each other’s issue types, rules, and workflows. They also didn’t want to leave the tool they were currently using.

> *“For us, flexibility with a large margin is the most important feature of Exalate.”*
> *David Müehlenhoff, development team lead at Nevaris Build*

That's why they started looking for a solution to help them bridge the information gap between applications.

Solutions

- An In-house tool
- 3-rd party synchronization solutions

Nevaris Build started connecting the 2 systems (Jira and Azure DevOps) with a tool they owned, which was just a once-off replication.

The tool allowed pushing issues from Jira to Azure DevOps. They had visibility on the issues that were being pushed, but the synchronization wasn't real-time. Plus, comments or changing specifications never reached the other system without manual communication or making the change manually in both systems.

They also decided to not pursue building a synchronization solution, because it required a lot of development effort on their part.

At this point, they started exploring other solutions in the market. They chose Exalate to integrate Jira and Azure DevOps.

The reason they chose Exalate was its scripting capabilities. They could easily customize their processes and implement their own logic to transfer data.

> "A lot of our peers liked the scripting feature of Exalate because they rightfully think they can solve complex mappings with it, although the other solutions in the market might be a bit easier."
> *David Müehlenhoff, development team lead at Nevaris Build*

Results

- Decentralized control of the integration (avoiding a single point of failure)

- Decreased issue-handling times thanks to automatic and accurate information exchange between teams
- Optimized costs as developers for Exalate were assigned and workflows ran smoothly

Initially, teams working on Nevaris Build had to spend a lot of time and effort manually transferring information or syncing information in batches.

Using Exalate, the admins could orchestrate the information exchange without the teams even realizing what goes on behind the scenes.

> *“It is also nice to have this distributed architecture that Exalate supports so failure on one side can’t affect the other side.”*
> *David Müehlenhoff, development team lead at Nevaris Build*

They now continue getting the information they want in Jira and Azure DevOps since Exalate continues working in the background whenever needed.

Earlier the developers had to spend a lot of their working time sending information back and forth between systems and keeping track of all the synced information and its related processes. And of course, issue-handling times increased because of this.

> *“The company doesn’t realize we’re using Exalate, which is a good thing. We have 3 people trained in Exalate, so they administer the solution whenever there’s a sync requirement or an error. The rest of the team doesn’t even notice because it just works.”*
> *David Müehlenhoff, development team lead at Nevaris Build*

With Exalate in place, the issue-handling time is reduced. Additionally, they get to spend time on tasks that really matter.

The teams have actually become more productive in managing their time and issues.

Use Case

Nevaris used Exalate for an internal use case where it wanted to connect 3 departments. The dev team, the product management team, and the QA team.

Every department has its own processes, workflows, and tools that it uses, and they were looking for a way so that all of them could work as one and have a centralized view of user stories, issues, bugs, etc.

> *“Exalate works pretty smoothly. And With the scripting engine, we could implement our own logic and transfer the data however we wanted.”*
> *David Müehlenhoff, development team lead at Nevaris Build*

Certain reports within Jira used by the QA team needed to be worked on in Azure DevOps used by the dev team.

How to Set up a Jira Azure DevOps integration

So they used Exalate to implement a synchronization that helped pass the required information between Jira and Azure DevOps. They also wanted to orchestrate workflows between the Dev, the QA, and other teams.

They used Exalate for its configurability, specifically for user mentions in comments.

Future

Having connected the development and the QA team with a Jira Azure DevOps synchronization, Nevaris Build aims to expand further.

It wants to connect its support team using Salesforce with Azure DevOps to streamline communication between them.

> *“Exalate is a winner there because it supports Salesforce Integrations. In our case, Salesforce is used by our support team and we want to set up synchronization between Salesforce, Jira, and Azure DevOps.”*
> *David Müehlenhoff, development team lead at Nevaris Build*

Their vision is to smoothen information flow and increase transparency between teams using disparate applications so that internal and external workflows are processed automatically.