



Eli Lilly Accelerates RIM Deployment with Agile Approach

HIGHLIGHTS

Adopted agile approach to increase value and speed

Expanded Vault user base from 300 to ~5,000 in two years

Increased submissions output by 50% in one year

Eli Lilly implemented Veeva Vault RIM with an initial focus on streamlining CMC post-approval submissions. The top pharma company used the traditional waterfall method to manage its first two releases but quickly realized the need to deliver more capabilities and business value at greater speed.

Agile Practices and Processes

Lilly pivoted to an agile project management approach, replacing one major release a year with three incremental ones. The faster release cadence allowed them to address new user needs sooner while sticking to set milestone dates. In parallel, Lilly established Discovery Action Teams to define and refine the capabilities in each release cycle. These teams were trained on Vault RIM and had access to a sandbox with appropriate data so they could accurately document business use cases and requirements.

In addition, Lilly enabled an accelerated timeline by improving multiple underlying processes, including requirements definition, release management, configuration, testing, and end user communication. They set up weekly reviews and dashboards to raise and quickly resolve any technical, resource, or scope issues. They also instituted end-to-end walkthroughs ahead of configuration lock to minimize rework delays.

Roadmap Acceleration

Since adopting an agile approach, Lilly dramatically reduced the time required to deploy Vault RIM capabilities. In a two-year period, the company delivered six releases, added support for more roles and submission types, and expanded the number of Vault RIM users from 300 to 5,000. Paula Hudson, RIM Program Leader, says, "The agile approach was instrumental in implementing the full scope of our capabilities in Vault RIM and achieving that value in our end-to-end regulatory submission management processes."

In 2023, Eli Lilly added Vault Submissions Publishing to improve collaboration, validation, and visibility for faster submissions publishing. Adding Vault Submissions Publishing to the existing Vault RIM platform eliminated manual data entry of each content plan into an eCTD tool. This not only enabled a faster, more efficient process, but also reduced risks and errors associated with manual data entry.

ELI LILLY

Company size: 34,000+
Headquarters: Indianapolis, IN

VEEVA SOLUTIONS

Veeva Vault Registrations
Veeva Vault Submissions
Veeva Vault Submissions Archive
Veeva Vault Submissions Publishing

Authors and Publishers Collaborate to Shorten Timelines and Improve Quality

Continuous publishing with Vault RIM enables earlier review and validation in the publishing cycle, says Tina Fuson, senior manager of regulatory affairs, Eli Lilly. Previously, publishers could not validate and review documents until authoring of all submission documents were completed. With continuous publishing, each authored document created during the submission process can be reviewed and revised by publishers as soon as it is completed. By working more closely together with submission teams, publishers can reduce multiple rounds of revisions to publish a submission.

Fuson also sees a positive change in the mindset of the submission team. "In the past, content authors didn't have to worry about [submission] lifecycle states. Now they reflect on those issues, and [authors] do some of the prework for publishing instead of merely telling the publishing team what text to replace or what changes to make." Greater collaboration created by parallel publishing and validation provides better overall visibility of the publishing timeline, leading to faster approvals and higher quality submissions.

With a more efficient publishing process, Eli Lilly has increased submissions output by 50%, from 400 submissions per month to 600 by the end of 2023. "As we get more comfortable with continuous publishing, efficiency is only going to [continue to] improve over time," Fuson predicts.