

Five Features to Consider When Building an Integration Solution



The need for software integration is well-established. We discussed about how a well-unified ecosystem helps enterprises deliver phenomenal customer experience in one of our previous blogs. Every enterprise that works with multiple systems and cross-functional teams realizes at some point the need for integrating their [Application Lifecycle Management \(ALM\)](#) and DevOps systems to accelerate delivery and provide a holistic customer experience. A good [integration solution](#) (the features of which we have discussed here) not only brings collaboration but also leverages the functional richness of the individual systems to create a highly productive and collaborative ecosystem.

In case of application development ecosystem, integration within the ecosystem and with the operations teams brings the spotlight on the business priorities and helps cross-functional teams, cumulatively, deliver high quality products and brilliant customer experience.

In this blog, we will discuss the factors that an enterprise whose core focus is something other than integration should consider when they decide on building an enterprise integration solution. We will also discuss why buying an off-the-shelf [enterprise integration solution](#) can be a wise choice.

Factors Enterprises Must Consider When Building an Enterprise Integration Solution Inhouse

Creating an enterprise integration solution is a complex task. It involves the understanding of the architecture of multiple systems, the uniqueness, features, and flaws of these systems, and the challenges in making them successfully work together. When enterprises decide on building an inhouse integration solution, their vision is mostly limited to their own ecosystem. Therefore, sometimes, they fail to consider the factors which, if ignored, will create issues over long term, the cost of fixing these issues and the impact on their teams' productivity due to these issues. These factors are: scalability, complete traceability, failure recovery mechanism, system upgrades, and change in business processes.

➤ **Scalability in future: Managing more systems + people + business**

Usually, the first problem that an enterprise faces with an inhouse integration solution is the solution's inability to scale up to meet the new requirements. Scalability might be needed for multiple reasons such as:

- **Addition of more users in the ecosystem due to business expansion**
- **Addition of more projects due to growing data/clientele**
- **Addition of new systems for newer business needs**

Now, as the initial integration design was not designed with the above needs in mind, such kind of updates typically require a full-blown project again to find a new solution. It becomes a time-consuming task specially when customizations are required to add new users, projects, or systems. For example, sometimes the new system being introduced in the ecosystem has no similarity with the with the already integrated systems and requires additional backend work to integrate with the existing ecosystem. More importantly, the delay caused due to enterprise's inability to consider the future needs for scalability, sometimes, impacts the successful execution of a critical project.

Another key reason why enterprises can't ignore the need of scalability in future is because as the team sizes grow, there would a need to successfully handle concurrent updates. The integration solution must be able to handle the concurrent flow of data from multiple systems without errors and failures.

To create an [integration solution](#) that is scalable for future needs is a specialized, time-consuming process, and requires a system development mindset.

Enterprises such as [OpsHub](#) dedicate a lot of time researching and working on ways to support multiple types of systems and keep their [integration solution](#) ready for future scalability. The out-of-box integration solution offered by OpsHub also comes with multiple deployment options, such as on-premise, on cloud, or both – a feature that is difficult to create inhouse.

➤ **Complete traceability: Rich contextual data in the ecosystem + Regulatory compliance adherence**

Complete traceability is important for richer collaboration and reliable decision making. An integration solution, designed to provide complete traceability for all work and non-work items, helps stakeholders put right quality checks in place and get holistic context of the data present in the ecosystem. Traceability is constructed piecewise in multiple, disparate systems and a good integration solution should be able to consolidate all the pieces of information. Lack of this sort of traceability not only affects productivity but also leads to breach of regulatory compliance requirements in industries such as finance and aviation.

Now, a minimal viable integration workaround that enterprises create inhouse to connect two systems will never be able to support complete traceability for all work and non-work items due to its poor design approach. Approaches such as API call approach that enterprises generally use to create inhouse integration solutions or, the otherwise slightly better, file systems approach do not support the kind of traceability that's actually needed to create a [unified ecosystem](#) with 100% traceability of all items.

➤ **A robust failure recovery mechanism: Safe data + Secure transactions**

Inflight failures when parallel streams of concurrent updates are coming from disparate sources (which is a common phenomenon in an ALM ecosystem) can be a nightmare if the integration solution doesn't have a reliable failure recovery mechanism in place. It can lead to loss or corruption of critical customer data. Therefore, it is critical for the [integration solution](#) to support automatic recovery in case of a failure and ensure eventual data consistency across the ecosystem.

However, this kind of robust failure mechanism is impossible for an integration solution if it doesn't keep a record of the transient states of all transactions happening across in the ecosystem in an audit log.

Now, handling failure efficiently is certainly not the top-of-the-mind feature when an integration solution is built inhouse. Therefore, it is common for these solutions to run into issues such as data corruption due to temporary system failure, connection loss between the two systems, or unexpected changes in mapping.

Robust integration solution such as [OpsHub Integration Manager](#) have robust failure recovery mechanism that queues up failures for resolution even when an end system is unavailable/down for a while. In the failure recovery mechanism designed by [OpsHub](#), failed events are processed automatically and could also be processed manually by administrators.



System upgrades: Readiness to handle API changes + Feature enhancements

Unlike a decade back, enterprises, today, don't follow bi-annual or annual cycles to release the upgraded versions of their software. Upgradation can be as frequent as weekly or monthly. Therefore, enterprises must always be prepared to ensure that the inhouse solution is always compatible with the new version(s). The preparedness could be: a) being ready to manage the upcoming changes in the software solutions or b) being familiar with the pre-release version of the software solutions. Advance planning not only makes it easier to adopt new features/releases but also significantly improve time to value. The reaction time to changes becomes even more critical when enterprises are dealing with Software as a Service (SaaS) products and they don't have the luxury to wait before adopting the latest release.



Change in business process: Re-evaluation of the existing systemic integration

Evolving business priorities, changing business needs, or change in the business model can impact the way two teams within an enterprise and their corresponding systems interact with each other. This, in turn, calls for a re-evaluation of the existing systemic integration in the ecosystem. For example, an integration solution designed on direct API call-out approach will need additional development to support even minor changes in the business processes. The process of development, testing, and deployment will not only add to the maintenance cost but also disrupt the business, sometimes for long periods.

The trigger-based integration approach that OpsHub follows is thoughtfully designed to manage such high-level business changes. The [robust integration](#) approach that OpsHub follows allows enterprises to make minor mapping updates to keep the integration up and running even in case of major business process changes.

In short, designing a robust integration solution is a complex task. When an enterprise realizes the need for [integrating its ecosystem](#) or integrating two or more systems, their default reaction is, 'Yes! Let's just create one inhouse'.

Prima facie, the process of connecting two or more systems looks very simple. Checking the APIs of the source and the target systems, and using the APIs to integrate the systems can be done by an enterprise's inhouse team – if not perfectly, at least to an extent where the immediate needs for integration can be addressed. The idea of developing an integration solution inhouse sounds faster, cheaper, and easier to handle as compared to going all over looking for a vendor that understands the enterprise's exclusive needs until enterprises consider incorporating the five critical factors we discussed in this article in their inhouse solution.

Even after putting all the money and effort, an enterprise not specialized in creating an integration solution can't be sure that they have built an integration solution better than an integration solution available off the shelf with a vendor.

Apart from niche skills and efforts required to design an integration solution, the collaboration between integration solution vendors and software providers to keep the integration solution ready latest for all latest releases give the vendors an edge over inhouse integration solutions. Therefore, if the core focus of an enterprise is not to create an integration solution – it is not worth investing years to develop an expertise in creating integration solutions.

The cost of building a robust integration such as [OpsHub Integration Manager](#), the effort required from the inhouse resources to reach that level of expertise, the time required to get the integration solution ready for use can rather be invested in enhancing the enterprise's core business application. This is the reason big companies such as [CA Technologies \(Rally Software\)](#), [Atlassian](#), [IBM](#), [Salesforce](#), [Jama](#), [Digital.ai Agility](#), and many more partner with OpsHub to provide integration solutions for their customers instead of building one inhouse.

Schedule a **complimentary consultation** session with our integration expert.



1000 Elwell Ct, #217, Palo Alto,
CA 94303

Web - www.opshub.com

Email - sales@opshub.com

Phone - +1.650.701.1800