Real world cost savings with enterprise app extensions

How low-code development can reduce costs, improve processes, and increase the value of enterprise applications

The rising costs of enterprise SaaS licenses

The SaaS revolution has generally been good for businesses. Free from having to manage implementations, updates, and associated infrastructure, teams within organizations have been able to move away from traditional monolithic or legacy internal applications. They've adopted modern, highly-usable tools that are purpose-built for specific functions and processes.

These tools have improved productivity, but they've also accumulated quickly. With easy adoption, the number of SaaS products within companies has grown exponentially. Today the average SaaS portfolio has grown to 371 apps[1]. Large portfolios are difficult to manage, and they drive escalating costs. The average SaaS spend has grown 53% over the past four years to $9,643 per employee[1].

While the SaaS market continues to grow, organizations are being increasingly cautious about their spend. In their forecast of worldwide IT spend for 2024, Gartner noted that "Organizations are shifting the emphasis of IT projects towards cost control, efficiencies and automation[2]." Aberdeen Group's 2024 State of IT report found that 74% of IT leaders plan to cut software costs by reducing seats, consolidating technologies, and re-evaluating vendor relationships[3].

> *Organizations are shifting the emphasis of IT projects towards cost control, efficiencies and automation.*
>
> Gartner
> Worldwide IT Spend Forecast, 2024

Companies are looking to reduce their overall spending on SaaS products. While some tools can be removed entirely, there are a number of core systems of record within organizations like ERP, CRM, asset and project management, that store critical

[1] Productiv, State of SaaS report 2024
[2] Gartner, Worldwide IT Spend Forecast 2024
[3] Spiceworks and Aberdeen Strategy & Research, The 2024 State of IT

business information. These solutions can't be easily removed, and teams across a business need to access and interact with the information they contain as a part of their jobs.

We tend to think of the "build vs. buy" debate as a choice about how entire applications are developed. But there's a middle ground that organizations can explore. Building custom extensions to enterprise "system of record" applications can give employees access to the information they need, and reduce seat licenses without disrupting important processes.

This guide walks through two specific examples of how the Appsmith low-code application platform was used to drive significant savings on enterprise software licenses. F22 Labs, an Appsmith customer & partner, saved $14k per year on their project management platform. And, Appsmith's internal QA team saved $80k per year on their testing platform. Read on to see their stories.

What are enterprise application extensions?

Enterprise application extensions can be thought of as mini applications that use the data in an existing (usually commercial) software application. These applications either "extend" interactions with that data to users outside of the core application, or "extend" the functionality of the application, allowing users to complete related tasks that rely on the core applications data without having to switch context. Extensions can function and stand-alone applications or be embedded within the core applications they extend.

Application extensions leverage API integrations with SaaS platforms to facilitate data interactions. Because API access is often less expensive than individual seat licenses (for example Salesforce Enterprise customers get 5 complimentary API licenses), These apps can provide access that employees need without having to purchase as many licenses.

Custom extensions can bring together information across multiple systems of record to create streamlined omnichannel experiences for employees that boost productivity. Customer 360° dashboards can bridge CRM, Helpdesk, and internal issue tracking systems to simplify support agents' activities. Integrated sales and order management tools can bridge CRM and ERP systems allowing sellers to close opportunities, place, and track orders in a single interface.

The challenge is, of course, that custom extensions require custom interfaces, and many teams lack the development resources to build and maintain them. Nearly 50% of internal development teams report being understaffed, making it difficult for

companies to keep pace with their current backlog of custom development projects[4]. Here is where low-code platforms like Appsmith can help. With fast drag-and-drop UI building, JavaScript customization, and the ability to connect to any REST or GraphQL API, Appsmith allows teams to build enterprise application extensions more quickly and with fewer development resources.

Example #1: F22 Labs saves $1,200 a month by developing custom extensions to their project management platform

Specific software platforms are critical to the business success, and for services businesses the most important platform is often their project management software. Effective project management tools help companies manage resources and project delivery, track team bandwidth and capacity, improve estimations, and bill accurately.

As with any truly mission-critical software, businesses are uncompromising in their expectations, and wary of costs — these tools get used very heavily within organizations. This was definitely the case with F22 Labs, a product design and development studio based in Chennai. Working entirely in-house, F22 consults with companies ranging from idea / early / growth stage startups to Fortune 500s to help them design and develop technology solutions.

Like a lot of companies they started with commercial off-the-shelf software as a service (SaaS), and explored the large array of solutions in the space. But they ultimately found that the right balance of cost and functionality came through the adoption of open-source, and the development of custom dashboards and interfaces.

The growing costs of project management software

With 60+ developers and designers working across multiple projects, the team at F22 Labs relies on robust project management software to keep track of in-flight work and individual team member bandwidth. The project management segment is well-served with many commercial products, but the team at F22 Labs struggled to find one that was a great fit for their business. “There are all kinds of tools in the market, and we’ve tried over half a dozen of them,” said Murtuza Kutub, Partner and Founder of F22 Labs, “but we had three consistent challenges. They were way too expensive as the team grew, they didn’t have a great user experience, or they didn’t have the kind of detailed reporting and dashboards that we needed.”

[4] Gitlab Developer Survey 2023

Believing they had no choice but to compromise, the team settled with ClickUp, a solution that paired the core backend features they needed with a good user experience. However, as their usage grew, the costs became unsustainable. “In terms of look and feel it was great,” said Murtuza, “but the challenge was to get the detailed analytics we needed would have forced us to go to a much more expensive tier.”

The project managers at F22 Labs oversee multiple, simultaneous projects, and a shared pool of development and design resources that work across them. To do their jobs effectively, they need visibility into tasks at an individual level across multiple projects. While ClickUp, Asana, and other commercial project management solutions provide this type of reporting, those features typically live in higher, enterprise-focused pricing tiers. “When you’re paying per seat, per month it gets really restrictive. Once you’re talking about 50 - 60 seats that’s nearly $2,000 per month for a project management tool. For us that just isn’t sustainable.”

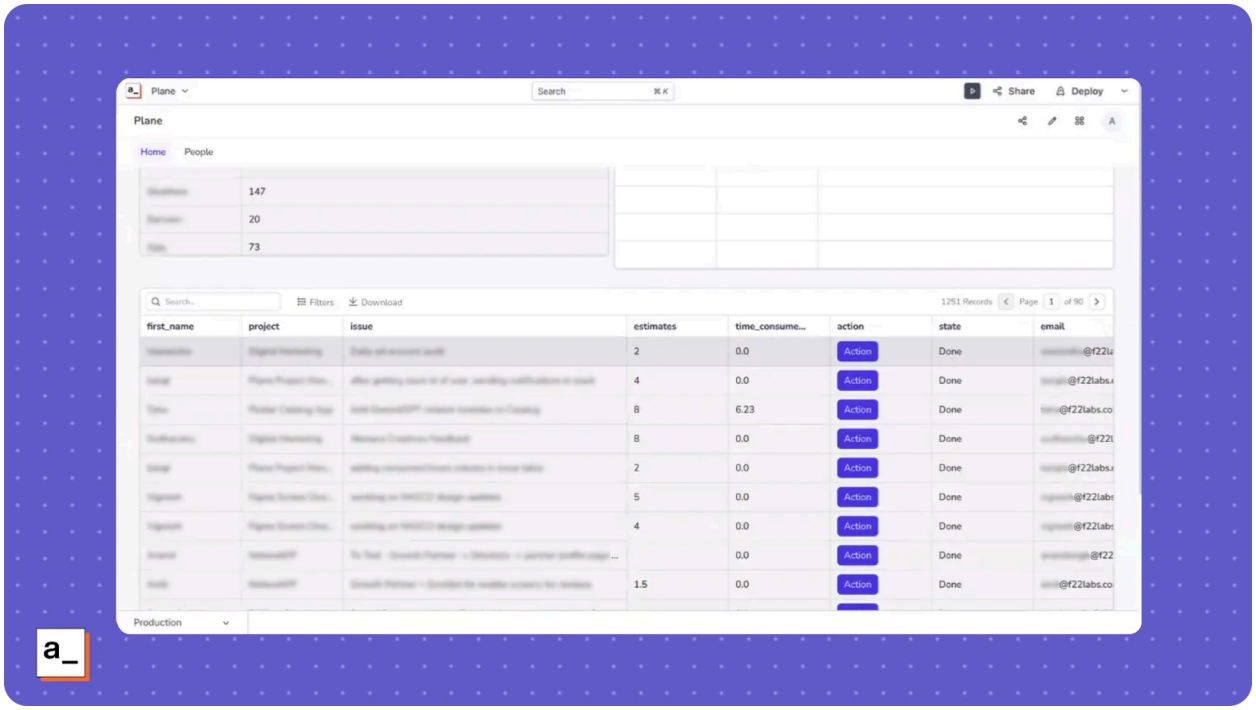
Moving beyond packaged software

The team at F22 Labs ultimately realized that in order to get the right balance of features and cost in a product management solution, they would have to look beyond commercial software products. Using a combination of open-source and custom software, they were able to build a solution that was perfectly tailored to their business.

For core project management functionality, the team selected Plane, an open-source project management platform. “Plane does the basics really well,” said Murtuza, “It gives us projects, tasks, sub-tasks, and assignees — the fundamentals of what any project management tool should have.”

What Plane didn’t have was the advanced cross-project visibility that had proved to be so expensive in commercial tools. To create this the F22 team turned to Appsmith. “We were familiar with Appsmith from a previous project. We had used it to build a customer success dashboard for an eCommerce SaaS product.” The team found that it was easy to use Appsmith to bring together customer usage data and onboarding status into a centralized dashboard that enabled customer success teams to wow customers. With this experience, they were confident that Appsmith would allow them to build the custom analytics they wanted on top of their project management platform.

Because F22 was hosting Plane on their own servers, they had direct access to the underlying database, and were able to connect to and use that data in their custom interface. “Appsmith is simple and straightforward to build with,” said Murtuza, “Within a day we were able to build out the entire dashboard we needed.”



Custom dashboards developed in Appsmith allow F22 Labs to track development and design resources across multiple projects

The Appsmith dashboard gives F22 Labs' product managers a comprehensive view across multiple projects. It allows them to quickly look across projects to see which team members are occupied and who has time left, what tasks are in-flight, how they're estimating hours, and the actual hours that are being consumed. "We've been able to do all of this with Appsmith. It provides a quick view for the project management team to look at, and assign tasks when something comes up."

Custom project management software saves F22 Labs significant time and money

Prior to having comprehensive visibility across projects, it was time-consuming and tedious for product managers to keep up with resources and bandwidth. They had to individually dive into each project to track tasks, and maintain heavy, sometimes disruptive, conversations with team members. Murturza recalls, "We tried a lot of things like creating a Slack channel for team members to post their availability, and then added a daily stand-up to share status. Everything was time consuming. Developers hated it because they were wasting up to an hour a day, and it forced project managers to keep too much information in their heads. This wasn't a system we could rely on as we grow."

With the cross-project visibility they developed using Appsmith, Murtuza estimates that they are saving their project managers 5 - 10 hours per week in time they would have spent in meetings or hunting information in individual projects. The company is also using the centralized data to improve their sales process. “Now we have all this centralized data, which we’re able to track over time. When we quote a new customer we’re working off an estimate provided by our team, but we weren’t able to go back to these estimates and track how those hours were actually consumed. With all of that data in front of us, we can make better decisions about how we’re pricing projects.”

The new project management system provided all of these capabilities at a much lower cost than commercial software. Self-hosting an open-source platform, and developing a custom UI (even using a low-code platform) adds additional overhead. However, Muturza estimates that the new setup is saving F22 Labs nearly $1,200 a month in software costs.

“Beyond just the hours and the cost savings, Appsmith has really helped us remove a level of friction,” said Muturza, “We’re giving dashboard access to teams for the projects they’re part of. Everyone has visibility in terms of the total allocated hours for a particular project, and can see how they’re tracking. This creates a sense of responsibility in the team in terms of where the project is, how they can deliver it on time, and what they can do to improve it.”

Example #2: Appsmith QA saves $80,000 per year by developing custom extensions to their testing automation platform

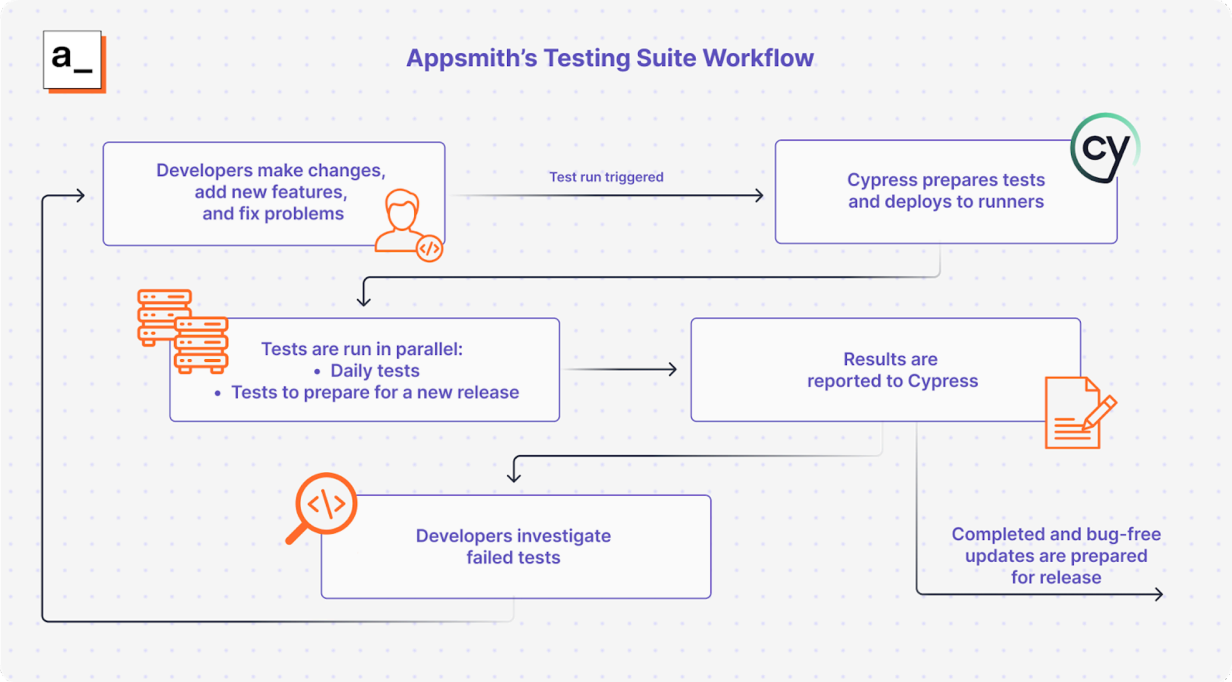
Many businesses from a huge variety of industries rely on the Appsmith low-code app platform to power their day-to-day operations — so it needs to be rock solid. Updates that add features or improve ease of use absolutely cannot break the apps our customers build. As a result, the team tests everything incredibly thoroughly.

As Appsmith has continued to grow, so do expenses for the software and infrastructure required to test each product change. Reducing the number of tests was out of the question (everything needs to be tested), so the team decided to replace some of the increasingly costly tools used in the testing process with an in-house Appsmith application.

Existing tools are expensive, don't scale well, and aren't flexible

Pivotal to the quality assurance process at Appsmith is the test suite: the software that tests each and every update we make to Appsmith. "If a bug is introduced, or someone makes a change to functionality that conflicts with another change, the test suite automatically flags it to the responsible party so that they can fix it and submit it for re-testing," said Yatin Chaubal, QA manager at Appsmith. This test suite is constantly expanding with the scope and pace of Appsmith development — more developers are contributing more features to the platform, and more contributions = more tests.

Appsmith uses the Cypress testing tool for this task. "We currently have over 2,500 tests in our suite, each of which automatically runs every time we make a change to the Appsmith code. We use 60 GitHub runners to execute these tests in parallel," said Yatin. These runners — short-lived containers that execute computing tasks — come at a cost, but are necessary to make sure that the tests are run as quickly and efficiently as possible.



The core of Cypress is open source and free to use, while additional features like test parallelization and report generation (both of which the QA team used) are paid enterprise functionality. The pricing model for accessing these features is usage-based, so as the quantity of our tests increased so did costs. "Our Cypress enterprise bill had eclipsed $80,000 per year," said Yatin, "We needed to find an alternative, and fast."

Using Appsmith to manage the development of Appsmith

As existing open-source and commercial alternatives did not meet the team's requirements, they decided to build our own solution. They used Cypress plugins to implement their own test parallelization and export the results to a SQL database. An Appsmith app provided a custom reporting dashboard to interact with this data.

Cypress organizes its tests into files called spec files. The first plugin that the team built for this project allocates eligible specs for the GitHub runners before Cypress begins its execution, splitting them equally in terms of time required so that they finish at roughly the same time. At the current load of 2,500 tests, run on 60 parallel runners, every run takes about 50–60 minutes.

The second plugin takes the results of these tests and writes them to a database. "Even though our purpose was to reduce our reliance on Cypress's enterprise features," said Yatin, "Cypress itself is still the best tool for the job, especially as we can extend it using plugins and inspect its output using hooks." The team validates each test as it completes. On a failure, Cypress automatically records the test artifacts such as screenshots and videos of the problem that was encountered and uploads them.

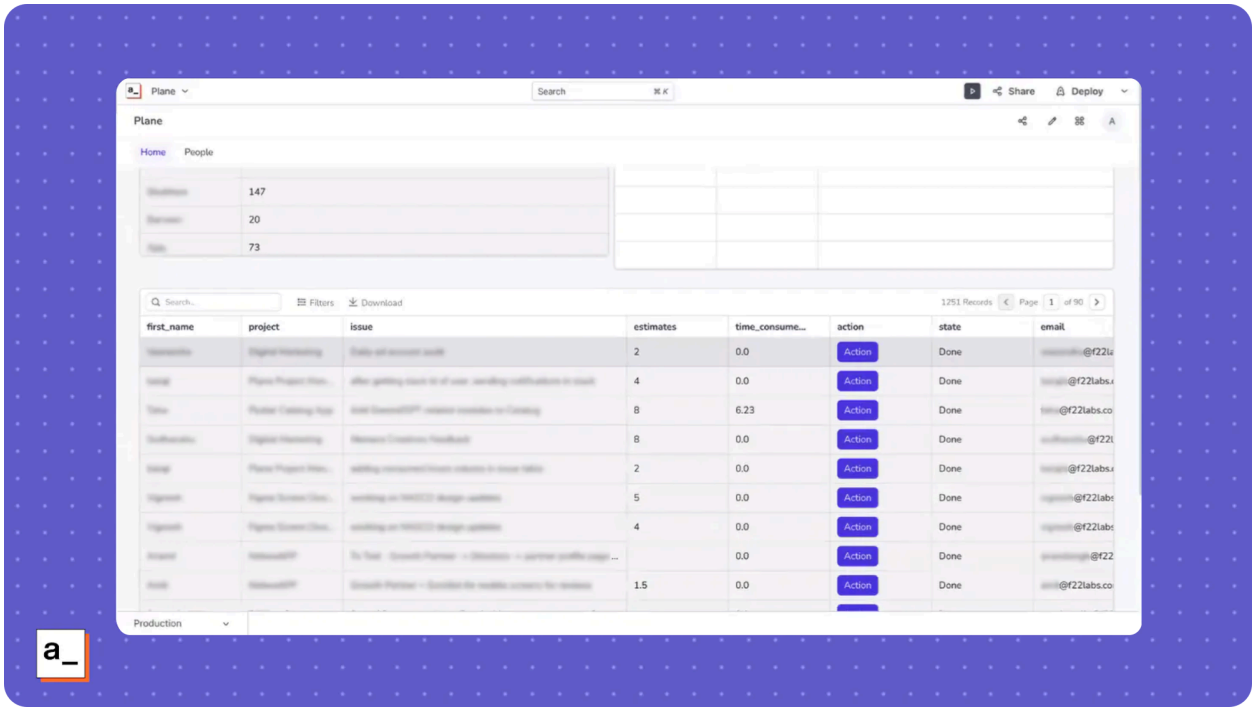
The final component that ties this all together is the dashboard — a graphical user interface created using Appsmith. "We used list, button group and chart widgets to quickly build out the user interface and implemented any remaining bespoke functionality using JS Objects," said Yatin. The team implemented access control using Appsmith's built in access control features.

Creating this reporting tool — from conception, to design, to building the plugins and dashboard, and testing it — took two developers three weeks. "For those not familiar with software development, this is a very fast turnaround for a custom application of this scope," said Yatin, "This was possible because we were building on the foundations of the Appsmith platform rather than coding everything from scratch."

The result: a scalable testing platform that matches the team's workflows and eliminates costs

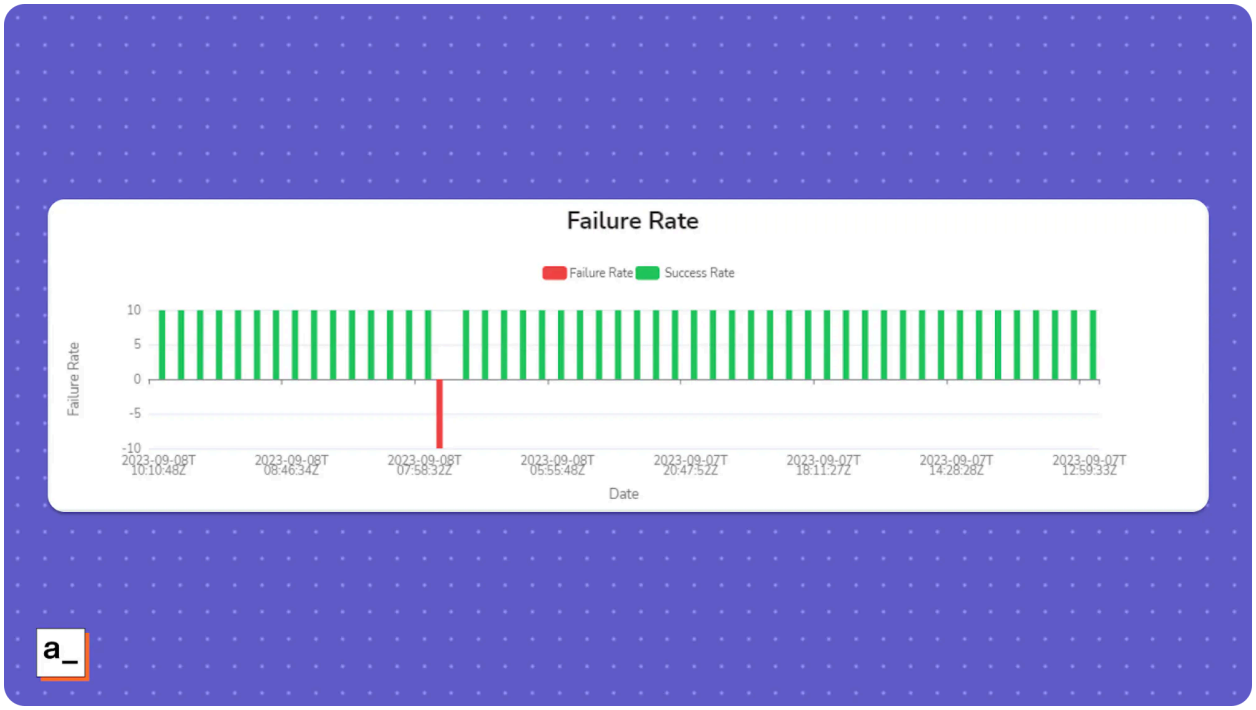
Alongside the significant cost savings, this was a highly rewarding project for the team as it really shows off how impactful Appsmith can be for an organization. Businesses in any category can look at these results and investigate which of their own expensive software tools could be replaced to improve productivity and reduce costs.

Below is the final result — the Cypress plugin has run the tests in parallel and reported the results to the Appsmith dashboard. For each test the team can see what it was, what triggered it, and who is responsible for it, as well as its final status using the color-coded button group.



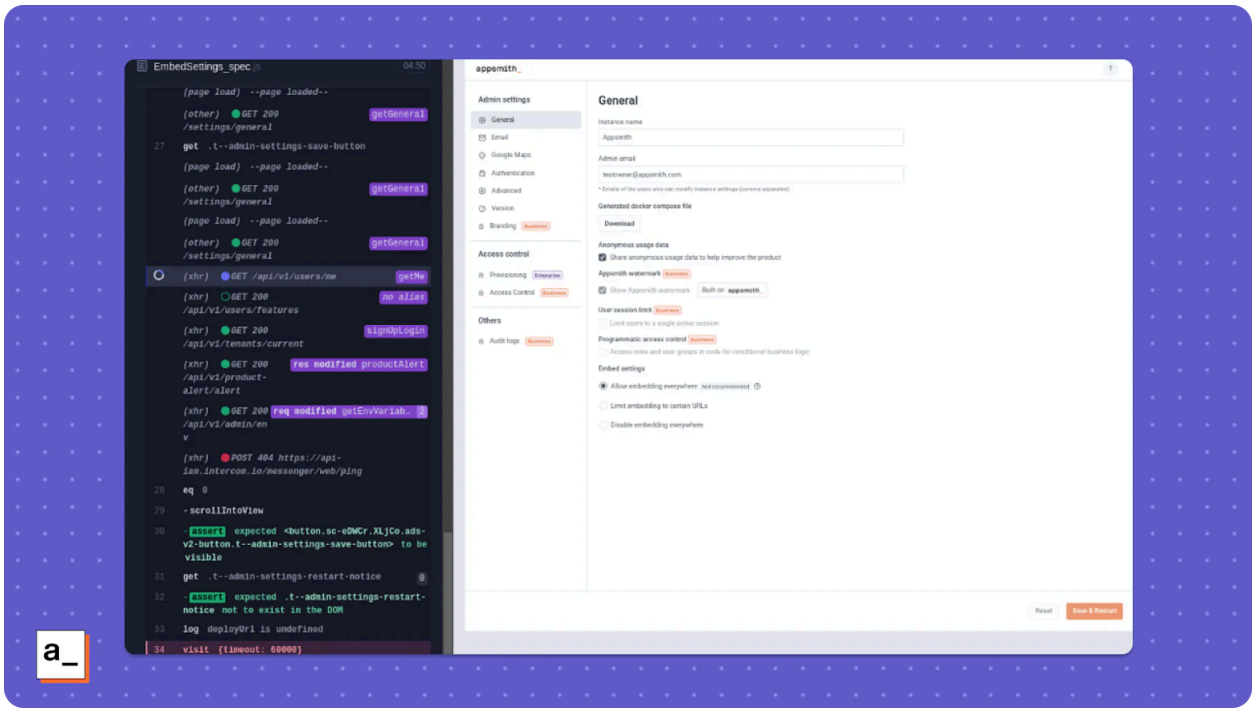
Each test run shown in the dashboard corresponds to a change made to the Appsmith codebase and summarizes: the developer responsible (1), the repository affected (2), what triggered the test run (3), and its current status (4). For the trigger, "push" happens when a developer merges their changed code into the Appsmith development code repository and "repository_dispatch" happens when the developer wants to merge their code into the main codebase. The status is color-coded, with green indicating that a test run passed, red that it failed, dark gray that it was skipped, and light gray showing pending test runs.

Users can filter and search the results, and they can click on a test to see its test failure rate graph. From there, clicking on a failure (red bar) reveals more detail about the failure and its cause.



Failure rate graph for a test.

That detail includes the testing artifacts uploaded by the Cypress plugin, including screenshots and recordings of the failure occurring, as well as technical details and log output.



Details, logs, and artifacts from a failed test.

Everything the Appsmith development teams need to quickly react to failed tests is available in the dashboard. This greatly reduces the amount of effort required to track and fix bugs and the amount of friction between remote teams collaborating on tasks.

Additional benefits of owning the toolchain

There have been some additional benefits that the QA team realized from the project. “Not only have we saved money and streamlined our testing process,” said Yatin, “Now that we're in full control of our tools in this area, we are able to add functionality that didn't exist in the proprietary solutions we previously relied on.” The team is also free of the effects of vendor lock-in such as unexpected price increases and functionality changes that counter their preferred workflows.

“If we want a custom report, or we see a feature in another product that we think will improve our workflow, we can simply implement it in our own dashboard. Our tools can match our desired processes exactly.”

The power of custom enterprise application extensions

A project management system for a digital services company and a testing platform for a QA team are both core systems of record. They are software products that are central to the work teams do, and they store mission-critical information. And, as we’ve seen in both examples, they can become burdensome expenses. Essential features are locked in expensive “enterprise” tiers, and individual seat licenses are required to view and interact with system data.

Custom application extensions provide a way for teams to break free from these constraints. Both the F22 Labs and Appsmith QA teams used the Appsmith low-code platform to build custom dashboards that replicated expensive enterprise-tier features in the underlying platform. The extensions leveraged API integrations with the underlying platforms to surface data — allowing teams to view and interact with information without having to purchase additional seat licenses.

Why Appsmith for custom extensions?

Most major SaaS platforms offer SDKs and low-code/no-code builders that are designed to facilitate extension development. On the surface these would seem to be an ideal solution. They offer deep, native connections to system data, and pre-built processes and components for common use cases.

However, they often use custom scripts and complex proprietary devops processes with steep learning curves. Residing entirely within the platform environment, they don't easily facilitate connections across multiple systems. And drag-and drop UI builders, geared towards "citizen developers" are often slow and difficult to customize. Companies looking to build within SaaS platforms' environment often have to rely on expensive external consultants to find the relevant expertise for their projects.

Business intelligence tools can also extend information from SaaS applications and consolidate information from multiple sources. However, they can be complicated to implement, sometimes require external data warehouses, and lack a critical functionality: the ability to update data.

The reality is that JavaScript development skills are much more readily available than custom Salesforce development or other application specialities. Low-code application platforms that leverage standardized languages and modern dev-ops processes can provide a simpler and more cost-effective solution to building enterprise application extensions.

Appsmith's unique approach to low-code development allows developers to easily connect to a wide range of APIs and SaaS applications, assemble responsive UIs with drag-and-drop modules, and write custom business logic in JavaScript. It's easy for teams to develop CRUD (create, read, update, delete) functionality, giving end-users full access to the information stored within enterprise applications (with built in security guardrails, and access controls).

Cost is always a consideration when it comes to business software, and companies often have to trade-off the value of expanding access to valuable systems with the significant incremental costs of seat licenses. The team at F22 Labs realized significant savings by building the custom dashboards and interfaces their team needed in Appsmith. The QA team at Appsmith used Appsmith to solve a significant cost problem, and wound up improving their internal processes as a result.

Learn more

Appsmith is free and easy to get started with. You can get up and running with Appsmith in minutes for free on our cloud-hosted platform. If you have custom needs for your business, contact our team to discuss how Appsmith can meet your unique requirements.