

The ultimate guide to
**Salesforce
Release
Management**

*Why release management support is essential
for low-code app success, no matter which
methodology you choose*



The low-code revolution


The Salesforce environment is no longer the sole domain of one person or one team. The clicks, not code movement has expanded the responsibility of platform configurations, app development plans, and releases of new features or capabilities across the business.

In today's rapid-fire, quickly changing business environment, IT leaders and Salesforce administrators must work in concert with business units to ensure the apps they run in Salesforce perform as expected and can evolve and adapt with the demands of the business.

They're embracing low-code app development tools and technologies as part of this drive towards greater business agility and focusing on delivering faster time-to-value of new CRM tools and features. The low-code software approach allows business users and Salesforce administrators to configure changes using point-and-click tools instead of writing code.

The low-code approach allows organizations to deliver customized applications at scale especially when facing a shortage of skilled developers who had previously been charged with building and maintaining business applications. But many mid-commercial and large enterprise teams are also taking shortcuts in their delivery of low-code apps, opting to make release updates directly within the production environment in the name of speed and agility, while exposing their entire organization to the possibility of unintended consequences from poorly managed releases.

Instead, IT and Salesforce leaders should seize the opportunity to elevate their strategic roles in the organization, combining their efforts to ensure changes to apps and systems are safe, secure, and properly managed as business units in their organizations continue to take more hands-on responsibility for various apps. That means that they must reconsider their release management practices and tools to help them realize the full benefits of low-code development while minimizing bugs, errors, or other missteps that can wreak havoc on an organization in the name of speed and velocity.



Low-Code Apps Are Hot

- The Low-code market is expected to reach \$21.2 billion by 2022¹
- By 2024, more than 65% of application development activity will be low-code²
- Approximately 70% of enterprises will have policies in place for citizen developers by the end of 2020³

Changes in production don't go according to plan

While low-code app development greatly empowers business users, it also opens the release and maintenance process to errors—especially if release updates are made directly within the Salesforce production environment.

Salesforce administrators and IT teams understand that customizing Salesforce in production is a no-no, but teams don't always apply that logic to low-code apps because the complex data schemas make it too time consuming to methodically and systematically manage these apps through the release process.

Worse, they may have bought the low-code app with the intention of a business user taking on some of the administration, only to find out that business users often don't understand release management methodologies or that making changes in production is a bad practice.

In particular, they might like the idea that changes in production take place immediately, but don't realize that skipping the testing phase before making the changes can potentially have a significant negative impact to the user experience and quality of the data. For example, a workflow rule could accidentally create an infinite processing loop or a simple change in a field's type might modify data in ways that can't be undone, necessitating IT intervention with costly and time-consuming workarounds to fix the issue.

Changes in the production environment also limit the scalability of projects. They're typically completed following a first-in, first-out model, or based on whoever shouts the loudest for attention. This lack of planning makes it difficult to prioritize delivery of key features or functions in a methodical, strategic way.

Reasons organizations may release low-code apps directly into a production environment:

- Business users live in production and want to see the changes in a familiar setting
- Business users have limited understanding of release methodologies, org management strategies, and best practices
- Managing low-code configuration changes through a release path often means time-consuming and error-prone manual work and re-work

The Change Management Conundrum⁴

Just **45%** of organizations automate release to production

The change failure rate of lower performing companies is **7X greater** than high performers

Deployment frequency of high performers is approximately **4% higher** than low performers

The average cost of a critical application failure for a F1000 company is **\$500,000 to \$1 million**

A multitude of better release practices

Salesforce admins and others tasked with managing new feature releases and updates have a litany of better alternatives to deploying changes in production.

Like their software development colleagues, administrators and other business users can employ proven methodologies like making one-off changes, taking a waterfall approach, or implementing an agile development methodology to help speed the time-to-production while averting errors.

Regardless of the preferred method, the result is an organization's increased ability to continuously deliver app innovation, catch issues before they're introduced in production, and ultimately achieve true business agility. To accomplish this, IT needs to build release management processes with low-code app managers, administrators, analysts, and project managers in mind, thus becoming a strategic partner to the business.

One-off changes

How It Works: Team members build changes in a sandbox, then promote to production in change sets.

Pros:

- Team members can catch mistakes before they impact users
- Team members can test changes with users before they go into production

Cons:

- Sandboxes can quickly get out of sync with production, slowing down new development efforts and degrading testing ability
- High risk of overwriting work as multiple teammates work in the same sandbox
- Change sets are limited and do not support the reference data that underpins low-code apps

Waterfall

How It Works: Often referred to as “org to org deployment,” waterfall methodology breaks releases up into linear steps for planning, implementation, and testing. In a waterfall methodology, companies [define an org management strategy for Salesforce](#) that defines how work will flow from lower level orgs to production.

While the exact release path will vary company to company, the common thread in waterfall is each team member starts new work in their own independent Developer org.

Pros:

- Eliminates risk of overwriting each other's work by isolating the changes each teammate is completing
- Enables prioritization based on business priorities
- Allows for a predictable release schedule
- Faster release of new functionalities

Cons:

- Creates monolithic change management that discourages risk taking
- Testing occurs at end of a release cycle, meaning issues may not be identified until it's too late
- Team members must keep track of changes to ensure the final release to production is correct

Agile

How It Works: An incremental approach to changes, agile focuses work on smaller components that can be delivered in a specific, short timeframe, called a sprint. Unlike waterfall, agile approaches insist that implementing and testing changes happens concurrently.

Pros:

- Detects issues early in project
- Allows rollback to earlier versions if needed
- Reduces level of effort of releases
- Encourages experimentation and more collaborative culture
- Results in higher quality releases

Cons:

- Requires mindset shift from “production org is source of truth” to “version control is source of truth”
- Requires ability to quickly create reproducible orgs you'll use for a specific change request then dispose of

Change management for every release methodology

Regardless of which methodology you use, you'll need to manage application operations to ensure you release new functionality and updates in a fast, error-free manner.

Effective change management:

- Achieves faster and more frequent releases with higher reliability and minimal interruption to end users
- Engages admins and other non-technical users in the configure, test, and release process
- Removes bottlenecks in app development
- Applies the governance structure, process, and security seen with traditional software development to low-code configurations
- Enables admins to find and fix bugs before they're introduced in production
- Continuously delivers application innovation
- Uses templates to manage even the most complex, relational datasets

The bottom line is low-code application development requires a strategy and new way of organizing your delivery cycles. You'll need a way to manage rollouts and updates, to ensure they are rolling out continuously and ultimately reducing time-to-value.

AppOps: Modern release management at the speed of business

Prodly is leading the charge to revolutionize the low-code movement. The integrated platform gives admins a simple way to manage configuration changes with the same change management rigor developers apply to code changes while automatically maintaining an audit trail of every change. Admins can seamlessly deploy data across different orgs to continuously implement, test, train, and release apps.

With Prodly, Salesforce admins and IT teams can quickly check in/out changes to data between multiple Salesforce organizations to better control changes. AppOps Release integrates with popular version control systems, allowing admins to track every change, compare changes, and resolve conflicts if multiple users are working on the same data.

It features pre-built data sets and deployment plans, along with specially designed deployment templates for common low-code apps so you can hit the ground running. Prodly's automatic data deployments make it easy for your team to:

- Automatically deploy entire relational data schemas at once
- Schedule future releases via Salesforce, APIs, command lines or third-party scheduling tools
- Deactivate and reactivate workflow and validation rules to prevent erroneous user alerts during a deployment
- Roll back deployments to restore the data in a Salesforce org to any prior version.

And its Salesforce DX plugin allows you to integrate low-code application change management into established release cycles to quickly seed a scratch org with data for testing, prompt app changes from a scratch org to a sandbox, and add AppOps deployments to Developer Hub processes.

Business agility in the new world of low-code apps

Business demands can change seemingly by the minute. In response, many organizations are adopting low-code Salesforce apps to empower business users to create functionality and configurations in support of rapidly evolving business demands without the need for extensive or expensive traditional coding.

The most successful enterprises have adopted best practices for software development to their low-code practices and are increasingly turning to a new breed of low-code release management tools to help them reap the full benefits of low-code development in their organization, while minimizing the negative impact of data errors and irreversible changes.

They know that the right approach, combined with the right technologies and expertise is key to aligning Salesforce and IT teams to deliver the apps and functionality the business needs, when they're needed most to remain relevant, competitive, and ahead of the curve.



Visit www.prodly.co to learn more about optimizing your Salesforce release practices or schedule a [free personalized demo](#).

About Prodly

Prodly helps companies build and continuously improve business applications faster, more reliably, and more frequently. We automate the full lifecycle of low-code development, empower more non-developers to configure applications, remove bottlenecks in the development process, and provide IT governance to mitigate risk of agile development.

For more information visit prodly.co.

Sources

1. Rymer, John. "Why You Need To Know About Low-Code, Even If You're Not Responsible For Software Delivery." Forrester.com. August 8, 2018.
2. Gartner Magic Quadrant for Enterprise Low-Code Application Platforms, August 2019
3. Forrest, Connor. "Citizen developers' are ready to fill the gaps in enterprise applications." TechRepublic.com. Accessed September 24, 2020.
4. <https://go.forrester.com/blogs/release-velocity-is-abysmal-and-needs-continuous-deployment/>
4. <https://devops.com/real-cost-downtime/>

