



Salesforce **CPQ** deployments

How to cut the complexity from
CPQ release management

Contents

| | |
|--|-----------|
| Who is this whitepaper for? | 3 |
| CPQ: a problem worth solving | 4 |
| Salesforce CPQ challenges | 5 |
| CPQ is technically complex | 5 |
| Many teams lack purpose-built tooling | 8 |
| CPQ deployments reimaged | 9 |
| External IDs handled | 10 |
| Deploying CPQ configuration like metadata | 10 |
| Full visibility into the impact of CPQ changes | 11 |
| Auditing CPQ changes alongside metadata | 12 |
| Why solving CPQ matters | 13 |
| Developer impact: frustration and inefficiency | 13 |
| Team impact: silos and poor collaboration | 13 |
| Business impact: unpredictable time and cost of delivery | 14 |
| The benefits of streamlined CPQ deployments | 15 |
| A better experience for developers | 15 |
| Effective collaboration for teams | 16 |
| Predictability and efficiency for the business | 17 |
| Next steps | 18 |
| About Gearset | 19 |

Who is this whitepaper for?

This whitepaper is for anyone wanting to untangle and eliminate the complexity of Salesforce CPQ deployments.

Perhaps you're in the early stages of CPQ adoption and want to ensure you achieve a successful implementation. Or perhaps you've had Salesforce CPQ in your organization for some time, but your existing release process is painful for both your team and the bottom line.

Getting to grips with this subject is important for anyone with an interest in rolling out or streamlining releases for Salesforce CPQ.

Technology leaders

Executives, managers, and stakeholders who are responsible for making decisions about the implementation of CPQ need to understand the practical needs of their internal technical team or consultants. Learn how to ensure a successful setup and integration of CPQ with existing business processes, maximizing its impact on efficiency and profit.

Devs, admins and CPQ specialists

Technical professionals who are responsible for implementing, configuring, and maintaining Salesforce CPQ within an organization experience the pain of deploying CPQ configuration first-hand. Dig down to the fundamental CPQ challenges, and learn how to solve them.

CPQ consultants and implementation partners

Professionals who provide Salesforce CPQ implementation services, such as system integrators, consultants, and resellers are looked to for expertise and efficient delivery, so master the subject and equip yourselves with the best solutions for CPQ.

Industry analysts and influencers

Advisors and guides across the Salesforce ecosystem provide trusted opinions and advocacy on a broad range of topics for CRM implementation and best practice. You'll gain valuable insights into the challenges of deploying a Salesforce CPQ implementation and see how leading solutions manage it more effectively.

CPQ: a problem worth solving

What is Salesforce CPQ?

Salesforce CPQ (Configure, Price, Quote) simplifies the process of creating accurate quotes for products and services. It reduces errors and ensures compliance with pricing rules and guidelines.

As part of Salesforce Revenue Cloud, CPQ helps businesses to streamline their sales processes, maximize revenue, and create better buying experiences for customers. Salesforce also offers Billing and Advanced Approvals — packages that further enhance the value of CPQ for different industries or expand its capabilities.

Integrating CPQ tools into the sales process can lead to significant improvements in performance indicators such as deal size, quote accuracy, and overall sales efficiency. CPQ also minimizes the time spent on administrative tasks, which frees up sales teams to concentrate on higher-value activities.

The state of CPQ deployments

Given these benefits, it's no surprise that there's a growing take-up of Salesforce CPQ. [The State of Salesforce DevOps 2023](#) report found that 42% of businesses use Revenue Cloud, and 29% more plan to adopt it soon.

Naturally, each organization will want to tailor the CPQ solution to meet its unique needs — configurability is what makes Salesforce so valuable. But without the right tools, deploying changes from development environments through to testing and finally to production can be time-consuming and error-prone at the best of times, and this is particularly true of CPQ configuration.

In this whitepaper, we'll outline the most common and complex challenges of deploying Salesforce CPQ, and explain how they can be addressed.

Salesforce CPQ challenges

CPQ is technically complex

While it's now part of Salesforce's Revenue Cloud offering, CPQ started life as a third-party solution. SteelBrick, acquired by Salesforce in 2015 for \$360 million, created CPQ as a managed package. So configuring and deploying CPQ feels different because it *is* different, and we'll go on to explore the various ways it's more tricky than most other kinds of Salesforce configuration.

CPQ is inherently complicated

Before we consider how configuration is represented and deployed, it's worth recognizing that CPQ is inherently complex.

There are many **factors that make it hard to generate an accurate quote manually:**

- Businesses have numerous products and prices to keep track of.
- Some products are unit based; others are subscription based.
- Products may be sold as bundles, or with different feature sets.
- Discounts might be offered, and there may be rules around discount limits.
- Selling into different regions has implications for currency and tax.
- Prices may change depending on quantities.

CPQ exists to automate these processes. It removes the challenge for sales reps, but shifts the complexity to the developers building and customizing CPQ. All of these interrelated factors need to be represented in CPQ, with dependencies between them.

Deploying CPQ involves data and metadata

The inherent complexity of CPQ makes deployments difficult. And to make matters even more challenging, CPQ configuration is represented as data records, not metadata. And yet it still relates to data and metadata objects across your org.

For example, Products, Pricebooks, Quotes, and Orders are all CPQ configurations structured as data. CPQ-specific metadata types for Apex, Permissions, Page Layouts, Customer Fields and Custom Objects are also used. And other regular metadata types, such as a Custom Field, might still be referenced.

Deploying data on Salesforce is typically harder than deploying metadata — partly because data often references metadata anyway, but also because there are usually more dependencies and relationships between data records than metadata types.

All this means deploying CPQ is a real challenge. Teams need to ensure data integrity, manage data mapping, handle duplicate or missing data, and make sure all dependencies are accounted for.

Record IDs are inconsistent between environments

Salesforce automatically creates a unique ID for each record. But, as anyone who's handled data migrations will know, when deploying the same record through different Salesforce orgs the Record ID changes. Without matching IDs, it's all too easy to get duplicate records in orgs. To make sure records are correctly mapped and deployed between environments, these records need to be matched up using a unique identifier.

A common approach used by teams deploying CPQ is to assign each record an “external” ID.

External IDs help you:

- Cross reference records between environments and even multiple systems
- Accurately deploy CPQ configuration data for products, prices, price rules, etc.

- Simplify integrations with external systems such as ERP or financial platforms
- Maintain consistency across all environments and systems
- Ensure data integrity — preventing duplicates and data corruption
- Speed up CPQ deployments and improve confidence in the release process
- Drastically reduce the likelihood of human error

All of those benefits underline the importance of external IDs, but creating and managing external IDs is also a huge amount of work. Teams deploying CPQ will typically sink lots of time on spreadsheets and scripts, trying to maintain a source of truth for CPQ configuration records and external IDs.

Many teams lack purpose-built tooling

It's critical for teams to find tools that will help them overcome the obstacles of CPQ deployments, but **until recently the options have been very limited.** Most teams continue to follow a fairly manual process, even though it's painstaking, tedious and error-prone.

Most deployment solutions aren't built for both data and metadata

As we've seen, deploying CPQ changes isn't like deploying regular metadata, because it isn't metadata. This means tools built solely for deploying metadata can't be used for CPQ, whether that's change sets, DX, or most third-party solutions.

On the other hand, data deployment tools aren't built for CPQ either. Usually they're built for loading records into an org from a CSV file, or for migrating data between orgs to seed sandboxes with test data. CPQ configuration can be deployed using these tools — some teams use Salesforce's Data Loader, for example — but it's hardly an intuitive process because they're the wrong tools for the job. It's a bit like opening a tin with a knife: doable, but hard work and pretty risky. Teams taking this approach will often rely on templates to help them configure the data deployments.

Many teams simply choose not to bother with any tools. Instead, they document all changes and recreate them in the target org — a highly laborious and error-prone process that reflects just how painful those teams have found CPQ deployments.

CPQ is often isolated from DevOps processes

As a result of these tooling challenges, Salesforce development and releases are very often split. One approach is used for regular metadata deployments; another for CPQ. Audit trails are either incomplete or divided. It's a challenge to plan or integrate work across the different workstreams.

In short, teams aren't able to get everyone together in one process to collaborate on release management. Building, testing, deploying, monitoring and planning all happen in silos. This compromises DevOps and violates the Agile manifesto, preventing teams from seeing the benefits of these approaches when deploying CPQ.

CPQ deployments reimaged

The many challenges of Salesforce CPQ are clear. This clarity makes it easier for us to identify what a proper solution for CPQ deployments should look like.

It needs to:

- Manage external IDs without the need for spreadsheets and scripts
- Bring CPQ into the same DevOps process used for metadata development
- Support deployments of Salesforce data and metadata together
- Give teams visibility into the complex configuration changes they're deploying
- Help teams identify dependencies and deploy them intact

These were the things Gearset had in mind when we set out to build [Gearset for CPQ](#). Our “compare and deploy” solution was already helping thousands of teams to deploy Salesforce metadata with dramatically improved success rates, and we wanted to make CPQ deployments just as easy.

External IDs handled

To save teams from the headache of managing external IDs, Gearset handles those after a one-time setup. **Our quick and intuitive setup wizard guides teams through the process** for each org. The external ID doesn't interfere with any existing external ID fields.

Before deploying CPQ configuration in Gearset, you need to run its external ID setup wizard on both your source and target orgs. You only need to go through this process once, unless new objects are added.

Adding the external ID is a simple, automated process, which prevents duplicate records from being created. It also enables rollbacks of changed CPQ items. The external ID doesn't interfere with any existing external ID fields you may already have.

You can enter the setup wizard from the list of your connected Salesforce orgs. Or, when you first try to compare an org that has CPQ enabled, Gearset will detect this automatically and prompt you to begin the setup process.

Deploying CPQ configuration like metadata

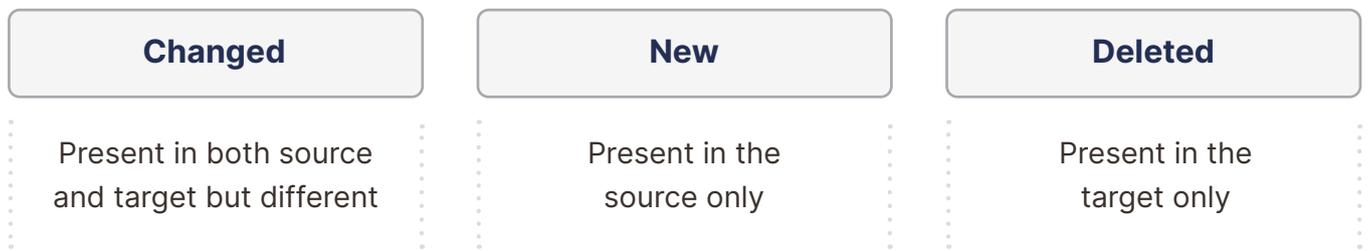
With Gearset for CPQ, teams can deploy CPQ config just like metadata. Using two different processes — one for deploying metadata changes and another for CPQ config changes — is time-consuming, inefficient and stressful.

Gearset cuts the workload in half.

To give complete visibility over changes, Gearset compares both metadata and CPQ configuration between source and target environments for the deployment. Gearset's comparison filter allows you to pull in the right CPQ and metadata types to compare and deploy — an improvement on deployment templates. There's a "standard CPQ filter" which preselects the CPQ and metadata types most likely to be needed. Teams can also create, save, and share other filters with the metadata and CPQ types they typically need to deploy together. CPQ, Advanced Approvals and Billing types are all supported.

Full visibility into the impact of CPQ changes

The comparison view in Gearset shows all differences that can be deployed. Items are categorized as:



Clicking on any item will show the exact line-by-line differences in Gearset's diff viewer, and we'll also highlight all dependencies between objects. If teams do miss any dependencies, Gearset will catch that during pre-deployment checks and offer to include the missing dependency. This is thanks to one of Gearset's many "problem analyzers" that find and fix common causes of deployment failure.

Comparison finished! All items (7236) New items (2537) **Changed items (284)** Deleted items (1793) Selected items (15) Filter...

| Name | Metadata type | Difference type | Changed on | Changed by | Record active status |
|---|---------------|-----------------|-----------------------|--------------------|----------------------|
| > @ ZFAUSKEY - 2 Factor Authentication USB Key | CPQ product | Different | Oct 20, 2021 1:26 PM | Jonathan Plumridge | Active |
| <input checked="" type="checkbox"/> @ HOMESecurityCONSOLE - Home Security Console <ul style="list-style-type: none"> <input type="checkbox"/> Depends on <input type="checkbox"/> Used by <input checked="" type="checkbox"/> Related <ul style="list-style-type: none"> <input checked="" type="checkbox"/> @ CPQ product option <input checked="" type="checkbox"/> @ CPQ product feature <input type="checkbox"/> @ CPQ pricebook entry <input checked="" type="checkbox"/> @ CPQ configuration rule <input checked="" type="checkbox"/> @ CPQ cost | CPQ product | Different | Nov 29, 2022 5:02 PM | Jonathan Plumridge | Active |
| > @ INTERIORKEYPAD - Interior Keypad | CPQ product | Different | Jan 19, 2022 11:49 AM | Jonathan Plumridge | Active |
| > @ SRINTERFACERLETTER - Printer, Daniel Letter | CPQ product | Different | Jan 19, 2022 11:49 AM | Jonathan Plumridge | Active |

Source: jonathan-cpq-trailhead@gearset.com
Item changed: Nov 29, 2022 5:02 PM

Target: jonathan-test@gearset.com
Item changed: Nov 29, 2022 5:02 PM

| Line | Source JSON | Target JSON |
|------|---|---|
| 57 | "\$BQQ__PriceEditable__c": "false", | "\$BQQ__PriceEditable__c": "false", |
| 58 | "\$BQQ__PricingMethodEditable__c": "false", | "\$BQQ__PricingMethodEditable__c": "false", |
| 59 | "\$BQQ__PricingMethod__c": "List", | "\$BQQ__PricingMethod__c": "List", |
| 60 | "\$BQQ__ProductPictureID__c": "", | "\$BQQ__ProductPictureID__c": "", |
| 61 | "\$BQQ__QuantityEditable__c": "false", | "\$BQQ__QuantityEditable__c": "false", |
| 62 | "\$BQQ__QuantityScale__c": "", | "\$BQQ__QuantityScale__c": "", |
| 63 | "\$BQQ__ReconfigurationDisabled__c": "false", | "\$BQQ__ReconfigurationDisabled__c": "false", |
| 64 | "\$BQQ__SortOrder__c": "1.0", | "\$BQQ__SortOrder__c": "3.0", |
| 65 | "\$BQQ__Specifications__c": "", | "\$BQQ__Specifications__c": "", |
| 66 | "\$BQQ__SubscriptionBase__c": "List", | "\$BQQ__SubscriptionBase__c": "List", |
| 67 | "\$BQQ__SubscriptionCategory__c": "", | "\$BQQ__SubscriptionCategory__c": "", |
| 68 | "\$BQQ__SubscriptionPercent__c": "", | "\$BQQ__SubscriptionPercent__c": "", |

New comparison Refresh comparison... Save draft deployment... Export all... NEXT

Auditing CPQ changes alongside metadata

Once you've selected items to deploy and passed the problem analysis stage, Gearset will prompt you to give each deployment a name and type in deployment notes, allowing your team to build up a deployment history that makes auditing and debugging much easier. The audit trail covers all metadata and CPQ changes alongside each other. You can roll back changes from any deployment in the history.

To help you monitor any changes, including any unwanted ones made outside of a DevOps process, Gearset's change monitoring automatically checks every day for differences to an org's CPQ configuration and metadata, and notifies you of any changes found. You can then decide whether these changes should be removed or pulled upstream to keep all environments in sync.

Why solving CPQ matters

Developer impact: frustration and inefficiency

Due to the complexity of CPQ, it's common for teams to have dedicated specialists whose sole job is to develop and deploy CPQ configuration. At the implementation stage, businesses may also choose to bring in consultants with expertise in CPQ.

CPQ specialists have valuable expertise in business systems and the CPQ model. They've learned how to handle configuration, migration, dependency management, testing, and change management. But, all too often, they're left working on repetitive and manual tasks.

Unsurprisingly, CPQ specialists often find their work stressful and difficult to manage. And regardless of expertise, CPQ specialists struggle without the right tools. A simple typo in a manual process will cause deployments to fail and waste many hours on troubleshooting.

Data collected for [The State of Salesforce DevOps 2023](#) report shows that just 21% of teams managing CPQ can deploy in under an hour; for everyone else the figure is 38%. In other words, teams deploying CPQ are almost half as likely to enjoy fast deployments. The same teams also identified "more apps and packages" as the leading cause of rising workload over the last year.

Team impact: silos and poor collaboration

Specialization, of any kind, within a team creates bottlenecks and single points of failure — precisely the problems that teams bought into DevOps are trying to avoid.

It's not that teams can't have subject-matter experts; CPQ specialists will always be the go-to people for their expertise. But collaboration becomes impossible when CPQ is being developed and deployed in an entirely separate workstream that the rest of the team have no visibility over. It's best practice to break down silos in development teams, so everyone can participate in one shared process and on the same platform. CPQ poses a real challenge to that ambition.

While teams working with CPQ aren't the only ones looking to improve in this area, 52% of them report needing better collaboration.

Business impact: unpredictable time and cost of delivery

The wider business, though often oblivious to the technical challenges of CPQ implementation and configuration, is still impacted. The delivery of CPQ changes can easily be delayed, costing the business in terms of developer time and missed opportunities.

CPQ configuration is already a difficult task, even before deployments are factored in. So it's hard enough for development teams (or consultants) to set expectations for the wider business about the feasibility of CPQ changes, the timeline for delivery, and the cost.

End users will make assumptions about how quickly changes can be released and integrated with existing systems. And often, changes need to be released at a particular time, for instance, at the precise moment the business changes its prices. This is hard to achieve when deployment times are unpredictable.

Delays in development all come with a cost to the business. Sales reps and other end users have to wait longer for features that could boost revenue. For example, the business may not be able to pivot quickly and match a competitor's discount. And the development team itself will struggle to clear a backlog of feature requests that the business needs.

It's also important to note that difficulties deploying data come with an increased risk of data loss or corruption. Since the CPQ model has such an intricate web of dependencies, releasing a small error can still have a wide-ranging impact. The data gathered for Gearset's [State of Salesforce DevOps 2023](#) reveals that 83% of orgs with CPQ experienced data or metadata loss in 2022, compared to 75% for other businesses.

The benefits of streamlined CPQ deployments

Solving the challenge of CPQ inverts the negative impacts on developers, teams, and the wider business to **produce a whole raft of benefits:**

- Increased developer productivity
- Better job satisfaction for CPQ specialists
- Improved collaboration across development teams
- Integrated workflows for deploying metadata and CPQ changes
- A unified approach to DevOps with shared tooling and audit trails
- Faster value delivery to the business
- More predictable timelines for CPQ projects
- Reduced costs for CPQ implementation and configuration
- Lower risk of data loss incidents

A better experience for developers

Gearset for CPQ has transformed the workflows of development teams building and deploying CPQ. These teams benefit in terms of time-savings, and also banish the stress of painstaking CPQ deployments.

*“Thanks to Gearset, I can **deploy CPQ data in 5 minutes** and focus on more important tasks!”*

[Katia Gleber | Business Applications Manager | WEKA](#)

“Gearset’s approach to CPQ deployments is groundbreaking. I would estimate it saves us on average 50% of the time we used to set aside for CPQ deployments. But the advantages it provides truly go beyond time-savings. Consultants have so much more confidence (and less stress) when deploying with Gearset.”

[Deepak Veera | Principal Consultant | Cloud Giants](#)

Effective collaboration for teams

Teams also benefit from bringing CPQ into the fold for DevOps. Using one platform for both metadata and CPQ deployments, teams can collaborate on releases and avoid treading on each other’s toes when workstreams collide.

“We wanted a tool to support us in the future — as our team grows, our processes grow alongside them. WEKA wanted something scalable that could support us in our future Salesforce development work.”

[Katia Gleber | Business Applications Manager | WEKA](#)

Predictability and efficiency for the business

Solving CPQ deployments generates time-savings that are ultimately cost-savings. Development teams are freed up to build more solutions, giving businesses more reliable estimates for delivery timelines. And they deliver changes to end users faster, adding value to the business sooner.

“Gearset simplifies our lives and frees us up to focus our attention on other mission-critical work. As a result, we’re able to deliver more value to our clients. It also helps build confidence in the time we’ve scoped for deployments. Gearset lends more structure and predictability. I can confidently go in and say if we follow these processes, this deployment should take two to three hours. Before that, it might be a five-hour thing, it might be a seven-hour thing. We just didn’t know.”

[Deepak Veera | Principal Consultant | Cloud Giants](#)

Next steps

Make the case for a new approach to CPQ

It's always necessary to get buy-in for a new approach. Particularly if you're looking to overhaul an existing deployment process for CPQ, proactively address any concerns the team may have. If you're still at the implementation stage, it's an ideal time to plan for release management and weigh up alternative routes forward. In either case, you should outline a cost-benefit analysis for your options — bearing in mind the costs of choosing or sticking with a manual release process. If CPQ is something of a black box to stakeholders in the decision, why not share this whitepaper to give them a rundown of the challenges?

Explore Gearset for CPQ

To see Gearset for CPQ in action and understand what it can do for your team, get in touch to book a tailored demo.

[Book a demo →](#)

About Gearset

[Gearset](#) is the leading Salesforce DevOps platform, with powerful solutions for metadata and CPQ deployments, CI/CD, automated testing, sandbox seeding and backups. It helps Salesforce teams apply DevOps best practices to their development and release process, so they can rapidly and securely deliver higher-quality projects.

Thousands of Salesforce professionals use Gearset, and have shipped millions of deployments, run billions of automated tests, and backed up billions of records. With inbuilt intelligence that solves the fundamental challenges of Salesforce DevOps, Gearset is a uniquely reliable solution trusted by more than 2000 companies, including McKesson, Accenture and IBM.

