How Can IT Empower Citizen Developers?

Governance, Tools, and Training for Low-Code Application Development

Table of Contents

82%

of IT leaders say that business units are willing and ready to create apps using a low-code approach.[1]

Introduction

The growth of low-code development – where business users can build enterprise applications with point-and-click tools without direct use of coding languages – has resulted in both an increase in productivity and a new set of challenges for IT. These business users, termed as 'citizen developers', have discovered they are able to use their aptitude and their understanding of business operations to create powerful applications. The applications they create have been deployed within their organization to teams ranging in size from small to large, **with development times as short as two weeks.** The results have included dramatic advantages in terms of business process automation, digitization, and productivity – all without too much hand-holding from IT.

How does IT feel about the rise of low-code development? According to a recent study, IT is ready to embrace this change:[2] 88% use or plan to use low-code development tools within the next 18 months. In the same study, two-thirds of IT leaders say that improving development speed is a high or critical priority. If we put the two together, it is apparent that the need for speed sparks these new strategies around low-code application development. IT executives realize that they can free up more of their time and innovate faster if they a) involve business-savvy citizen developers in mission-critical app development, especially at the prototyping stage, and b) reduce their backlog of non-mission-critical apps by empowering a group of citizen developers to build those apps independently – with limited involvement from IT. However, the limits as well as the mandates of that involvement have to be defined and documented to ensure successful adoption of low-code development. IT's role in understanding and governing low-code development projects is critical. The question is: How and when should IT step in to review, assist, or assume control over citizen developers' projects?

[1] Top Trends in Low Code Development, Salesforce Research, February 2017.
[2] State of IT, Salesforce Research, April 2017.

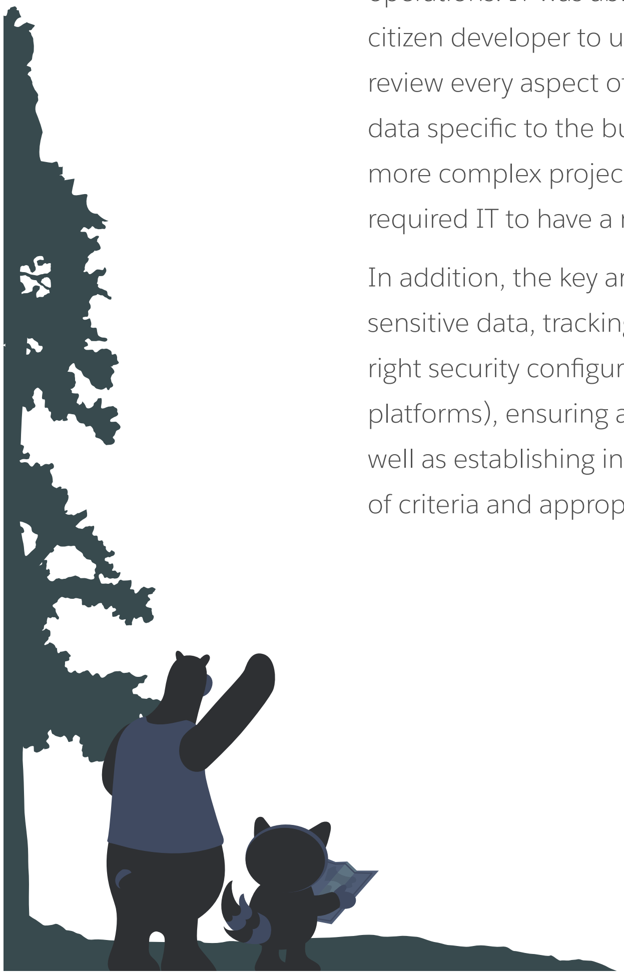
The Need for IT in Low-Code Development

The next generation of cloud platforms, like Salesforce's high-productivity application platform as a service (aPaaS), gives nontechnical users visual tools to build highly functional business applications – without the need to write thousands of lines of code. People who understand their business' needs and have a grasp of high-level technology concepts can create apps even if they may not know how to write code and have no formal schooling in the standards and disciplines common to IT professionals.

The implications of this are profound. Now businesses can create applications in weeks that would have taken months if they had followed the standard development processes. But there is a point at which factors like security, deployment, and data management of these applications should be reviewed and governed by IT. No matter how advanced the low-code platforms become, there will always be cases where a thorough quality review of the app by IT will be necessary before citizen developers can publish it.

As an example, the IT group at a large bank decided to provide low-code capabilities to its business teams so each unit could develop specific projects important to its own operations. IT was able to limit its oversight responsibilities by identifying and training a citizen developer to use the low-code development tools. It wasn't necessary for IT to review every aspect of every project because the applications were limited to managing data specific to the business group, and would not impact enterprise resources. However, more complex projects in which the apps needed to integrate with and update ERP data required IT to have a more active role in governance.

In addition, the key areas where IT involvement is needed include managing access to sensitive data, tracking and troubleshooting application performance, making sure the right security configurations are set for the app (from the available settings of low-code platforms), ensuring applications can scale both in terms of functionality and usage, as well as establishing integration standards.. Every organization should map out its own set of criteria and appropriate levels for oversight.



Benefits of Low-Code Application Development

There are many benefits to using low-code development. First, if you are already using SaaS apps based on a trusted platform (PaaS), the cost of building additional custom applications is minimal, since low-code development tools are simply an extension of the platform – with all the security, integration, and user management capabilities built in. Besides, the valuable data contained within your SaaS apps, particularly a rich source such as your CRM system, can be easily extended to custom apps, decreasing time to value and providing consistent view of your business data.

In addition, there is little-to-no risk of loss of IT control. Low-code platforms come with configurable security settings, intuitive application lifecycle management (ALM) tools, and usage analytics that IT can easily access by leveraging admin privileges.

Low-code tools allow for a closer collaboration between business and IT, which results in dramatically better apps faster. The power feature of the low-code platforms is that they allow for quick prototyping of a real app. Even in cases where IT leads the development, the visual process, where IT sits side-by-side with the business users and iteratively builds the app with real-time feedback, helps ensure business is happy with the app they get.

The real benefits, however, come from the apps themselves. These apps can enhance business workflows and securely extend the use of enterprise and customer data to everyone in the organization – making more parts of the business more productive.

Finally, adopting low-code development helps grow the talent pool of builders and creators in the organization necessary to speed its digital transformation.

The questions that need to be addressed to make this kind of user-driven development truly appropriate for enterprise use fall into two categories:

Who are the citizen developers?

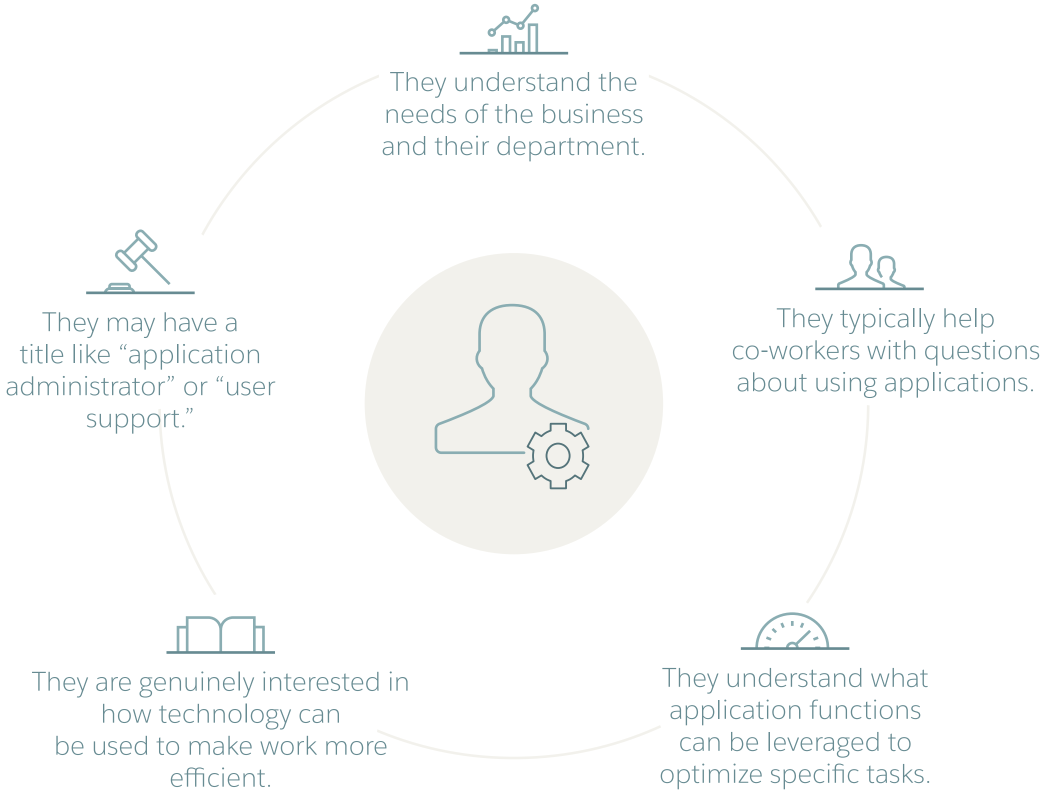
What governance role should IT take?

Meet the Citizen Developer

Citizen developers are most likely the employees – analysts, project managers, operations managers – within business units who have created spreadsheets to collect and organize information so they can get their work done more efficiently. They have developed an advanced understanding of the tools they use. Whether they're using spreadsheets like Excel or a database like MS Access, they have learned how to use features like macros and lookup tables to create software tools that help them get their jobs done. But citizen developers could also be administrators of applications or systems who are willing and able to roll up their sleeves to create apps in addition to managing them. Typically, they are also enthusiastic about using technology and often provide mentorship to their co-workers on technical topics.

Citizen developers have a strong understanding of the business and understand the processes involved in making their part of a company successful. They have been in – or aspire to be in – an administrative role where they can handle a technical aspect of an application or process. But most importantly, citizen developers have a desire to learn and create.

Five characteristics of sucessful citizen developers



They understand the needs of the business and their department.

They typically help co-workers with questions about using applications.

They understand what application functions can be leveraged to optimize specific tasks.

They are genuinely interested in how technology can be used to make work more efficient.

They may have a title like "application administrator" or "user support."

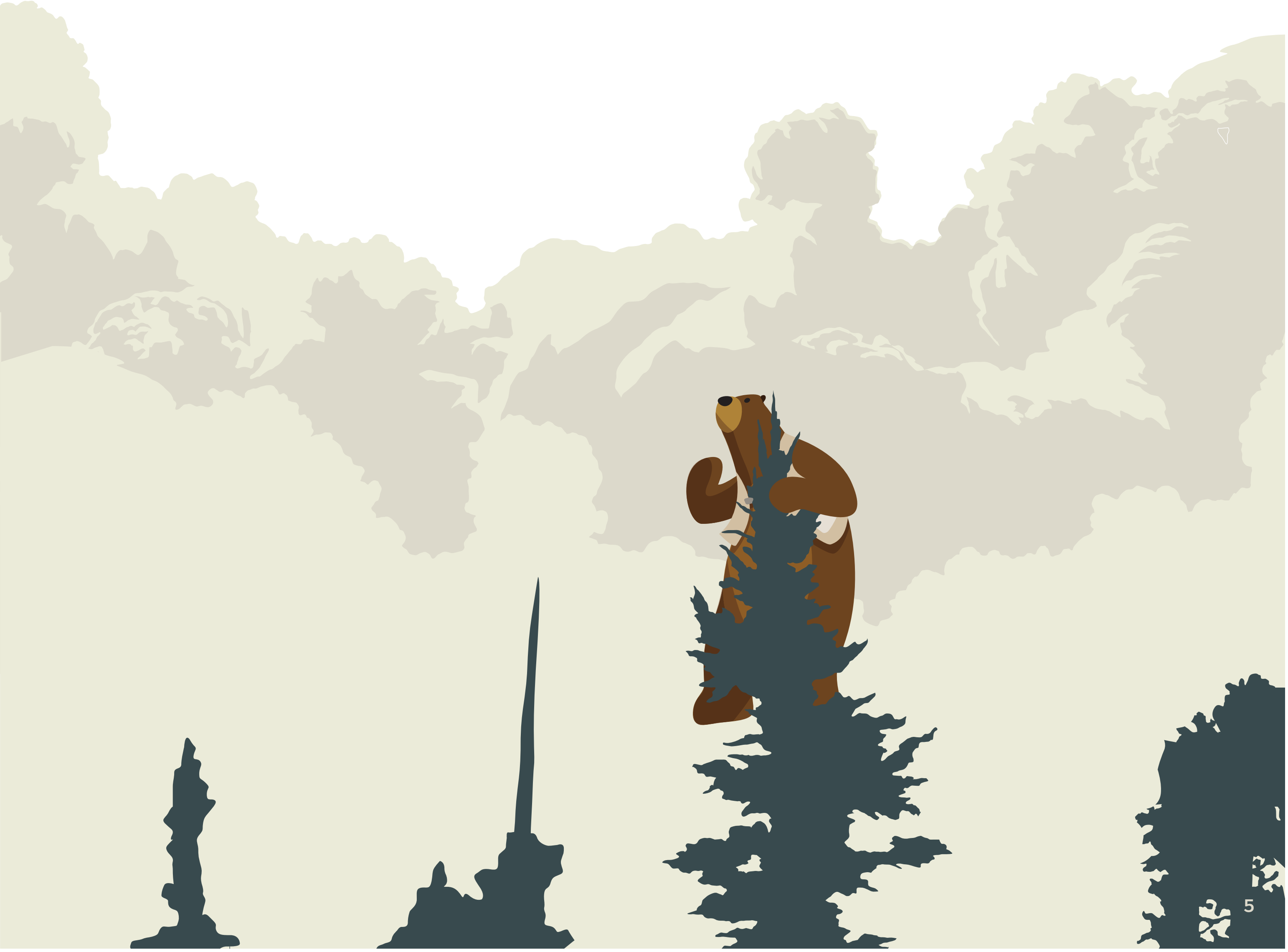
How Involved Does IT Need to Be?

The level of IT governance required depends on the projects being developed and the structure of the enterprise. It can range from hands-off to more traditional project development based on IT's internal processes. Even if projects are initiated independently and at a small scale within a business unit, IT can and should revise its oversight as needed, depending on specific conditions and requirements.

There are three approaches to governance that range from virtually hands-off to strong oversight.

There are three approaches to governance that we'll explore in more detail on the following pages:

1. Minimal governance
2. Shared governance
3. Complete governance



Approach #1: Minimal Governance

HOW IT WORKS

- An application is typically used within a single department.
- Data is departmental in nature and does not extend to enterprise data.
- The user audience is small (one to five users).
- Citizen developers are independently competent.
- The business typically finds and deploys the development tool.

CASE STUDY

From Sticky Notes to Optimized Code

As a large bank was expanding and adding staff, the HR manager used a commercial HR application to track and process applicants. But the system didn't allow the manager to add notes that were important to the process, so the manager used a series of colored sticky notes posted to the computer monitor to augment the process. A citizen developer who had been using the low-code app dev tools of the Salesforce platform noticed the jumble of sticky notes and asked the HR manager a few questions. Based on his understanding of the company's processes and the manager's needs, he went to his desk to develop an application designed specifically to address the paper clutter. Two hours later, the HR manager had a fully functional program that allowed him to track the additional information the HR system was unable to handle.

This is an example of tools that let people work together with better speed and make significant contributions to their business. In this case the data being captured was contained within the HR department, through the ancillary app, and didn't affect corporate ERP systems. This allowed the project to be completely handled without IT assistance, processes, or governance.

Approach #2: Shared Governance

HOW IT WORKS

- IT deploys low-code platform for use by citizen developers.
- Business users are trained on using the development tools.
- IT uses a sandbox environment and a testing process for citizen developers to build without breaking things.
- IT creates secure components that connect to enterprise systems when the app needs to access enterprise data.
- IT writes code if custom UI or advanced functions are needed

Planned Shadow IT

CASE STUDY #1

Not all shadow IT, where a business uses systems or apps without IT's knowledge, is viewed as disruptive. A large energy company encouraged a form of shadow IT and promoted its use as a way to supplement IT efforts. This approach is especially helpful when IT's capacity is constrained and central IT organizations face tight budgets and overall cuts. Individual business units still want to fund projects, but if they can't call on IT for development resources, they can decide to build solutions on their own. The energy company's IT team decided to empower its users to create applications that didn't require coding. They trained citizen developers on the low-code approach using the Salesforce platform as a development tool.

Because IT was involved in initiating the departmental deployment, the scope of development efforts was restricted. IT only needed to intervene when enterprise data sources needed to be tapped. Development efforts that maintained data local to the department were encouraged – and they required minimal or no intervention from IT.

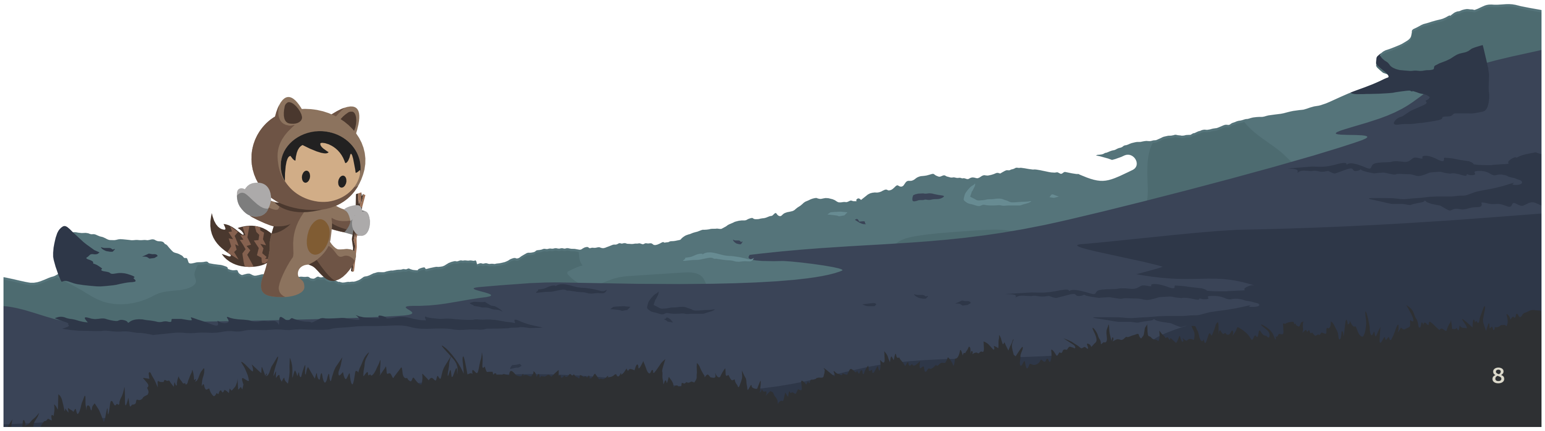
Because of the need for applications to make external API calls, governance tends to become an issue once the applications need to connect to enterprise systems. The issues revolve around security and firewall permissions. Fortunately, the Salesforce platform provides infrastructure, network- and application-level security controls to address this. Especially at the application level, IT gets to control who sees and does what. Once these are established by IT, the apps can be deployed with no need for additional IT intervention.

Moved Fast on Small Initiatives

CASE STUDY #2

Low-code development can bring new solutions quickly to companies that have long development cycles. A large energy company adopted low code as an experiment. They found that business units could take some small initiatives into their own hands and that there was a lot that low-code developers could do. Departments were able to see great results in just a couple of weeks. They started to see that citizen developers using low code could move faster than the company could.

In order to provide the right level of governance for this experimental project, IT identified an agile review and collaboration process that approved the business case for the app and provided the necessary technical support. The first line of review was to conduct an audit to evaluate proposed applications based on the number of users affected, the investment required to get the app into production, and any ROI impact. Once the application request passed initial audit, IT created the necessary data connections with enterprise systems. In this example, IT built connections to external systems in the form of external objects. While these could have been built by savvy citizen developers with just clicks, IT decided to retain control over the integration components of the application because of the standards they maintained internally across all apps. Once IT provided these components, the citizen developers built the entire app without further IT involvement.



Approach #3: Complete Governance

HOW IT WORKS

- IT defines tiers of application types.
- Citizen developers and business units create requests to develop applications.
- IT reviews the requests.
- If approved, IT may offer that the development be done by the citizen developer.
- Where needed, IT builds components for use by citizen developers in composing the app.
- The development of more complex apps are led by IT, with iterative reviews by users.
- Low-code developed applications are tested in a sandbox environment created by IT.

The Power of Tiered Structures

CASE STUDY

Larger organizations may need to put more structure behind their citizen developer efforts. One large financial services company created two distinct instances (orgs) of Salesforce managed by two groups, each with its own level of governance, used by developers with different levels of skills (professional and citizen developers).

The first instance is called “Lightweight” and has a different governance process: It starts with the department submitting a request to create its own application. The application request is then reviewed by IT, which audits the department’s portfolio for any existing applications that can meet the need, then considers any security and legal concerns. That process normally takes two days. If the request is approved, department leaders are asked whether they want to build all or part of the application themselves.

Initially, the challenge for IT was that without governance, the citizen developers had the ability to create modifications to field displays in their own designs that also affected the fields in other applications built on the platform. The company solved this issue by providing citizen developers with a sandbox environment where low-code development could be built and tested prior to being shipped to production. A gatekeeper administrator was then assigned to review the sandbox applications for their potential impact on other groups’ projects before being moved to production.

The second ‘high-control’ instance of the platform is named “Enterprise.” This instance is controlled by IT and follows standard IT processes. Application requests that have been approved, but that need more programming or data access, are handled by IT. If a citizen developer’s application becomes popular and is adopted by large numbers of users, it is moved to the Enterprise instance (org) and falls under IT’s governance.

Choose Your Governance Path

Here is a simple checklist of what you need to do in your path to governing low-code development:

1. Identify existing or potential citizen developers.
2. Establish a credentialing process for citizen developers – training, certification.
3. Determine what projects are being or have been created.
4. Establish where the citizen developer apps are being created (for example, in office tools or in the Salesforce Platform).
5. Establish agreed-upon points of technology that call for IT involvement.
6. Set up standard protocols for low-code development – sandboxing, testing, validation, audit, integration, and deployment.
7. Determine an evaluation process to decide whether a project is a candidate for IT development or low-code development
8. Apply appropriate governance processes based on this evaluation.

Low-code development efforts have become the preferred way to put well-structured and secure applications into production quickly without negatively impacting IT's enterprise projects. But IT needs to establish effective governance to ensure enterprise data security and proper uses of resources.

Learn more about the Salesforce platform for application development.

WATCH DEMO

Calculate the ROI of building on a low-code platform.

SEE YOUR ROI

Authors & Contributors

Kashi Ahmed
Platform Architect

An IT strategist with over 16 years of hands-on enterprise application architecture and development experience of complex computer systems, and more than eight years of technical consulting experience managing software implementations and cross-functional teams of developers.

Jeff Blanchard
Director, Solution Engineering

Performance-driven cloud computing professional with more than 22 years of expertise in technical solution engineering and architecture. Engages with enterprise customers to demonstrate the vision of what's possible with Salesforce's intelligent customer platform.

Cameron Collins
Senior Technical Architect

Seventeen years of information technology management and architecture experience in the natural gas pipeline and transmission industry. Expert in CRM systems, enterprise architecture, application architecture, application development, and customer transactional system implementations.

Bobby Dombos
Lead Solutions Engineer

Passionate about the possibilities with the Salesforce platform since the first time he clicked 'Setup.' Part of the team helping customers innovate with custom apps and a self-taught master citizen developer.

Bryan Jennewein
Senior Product Marketing Manager, Salesforce

A successful grower of people, products, and business, big and small; an inventive and interactive product marketer. Passionate about the Salesforce platform, citizen development, and empowering people to be their best.

Shane McLaughlin
Platform Architect

An entrepreneur who happens to work for Salesforce on new and disruptive things. Specialties: building engaging applications on Salesforce, connected to enterprise data, using standard tools for UI (CSS/JS/HTML, especially AngularJS and Aura/Lightning framework).

Aygun Suleymanova
Director, Product Marketing, Salesforce

Technology product marketer with over eight years of researching and educating IT audiences on trends in cloud platforms and application development. Passionate about mobile, cloud, open source, and new trends in intelligence and distributed systems.