DevOps Gaining Speed

Salesforce Lightning provides a unique setting for testing DevOps strengths

The new business apps reality

Flexible applications drive business agility which drives business success and profits. But what drives software flexibility? A number of factors contribute to flexibility and they can include technology and organization culture but at an even more fundamental level, modern software can't exist without a development platform that can generate the lion's share of code that runs a business.

Even taking into account a business' need to develop its own custom algorithms and support for intricate business processes, those organizations that enable administrative users to transform apps through declarative means ("clicks not code") or to generate some or all of the code that weaves together important supporting functionality such as analytics, security, integration, workflows and much more, have a decided advantage over competitors whose practices have not substantially changed in decades.

That's the second part of the digital disruption. We're all familiar with the first part, the idea of leveraging customer data to support real-time decision-making. But that can't be even attempted unless the organization also has the ability to change its business processes and its applications when needed to support agility. After all, what good is knowing something but not being able to act on the knowledge?

DevOps

The other part of a rapid development and deployment strategy for business apps is method or how developers leverage company resources and how companies in turn support them. DevOps is an approach that combines development and operations groups and their practices in an effort to improve software quality and speed to market. There is also a large selection of techniques like Lean and Scrum as well as organizational approaches to either speed development or support new ways of working for developers. DevOps combines some of all of this to offer a streamlined approach to both.

Salesforce presents a well-integrated development and production environment, so applying DevOps standardization on issues like operating systems, databases and even development languages is not the same as in an IT department where there has historically been more wide-ranging selectivity. Certainly, standardization is part of any DevOps strategy, but our point is that the Salesforce Lightning platform reduces that need.

The ongoing digital disruption offers businesses an opportunity to become more competitive and to capture share in their traditional markets by leveraging their customer data to ferret out information that can enhance customer interactions. But this opportunity primarily exists for organizations that can both analyze their data quickly and just as quickly adjust their customer-facing systems to best take advantage of the opportunities presented by analytics. But what does it mean to adjust customer-facing systems? How fast? How often? What are leaders doing? This Beagle Research report is designed to offer some answers.

Numerous approaches to enhancing organizational software productivity by reducing time to delivery and costs are available including Scrum, DevOps, Lean, Kaizen and others. But most approaches focus on team sociology and methods while few focus on the tools developers use. As a result, some of the research data concerning rapid deployment are littered with contradicting information coming from experiences with numerous tools. Is a particular approach the result of tools or methods and if so, which ones excel?

Part of a DevOps approach involves standardizing as much as possible on tools and methods but that still leaves the question of which ones are best. This report focuses on one particular set of tools: those embedded in the Salesforce platform, Lightning, which supports both declarative and procedural application development. DevOps is a good match for the Salesforce Lightning because the platform blends development, integration, testing, release, monitoring, and management, all key factors in delivering working apps quickly.

Key Findings

This report surveyed 201 executives whose organizations employ Salesforce Lightning and a DevOps strategy.

1) Most report significant benefits from their strategy including 17 percent who claim over $5 million in benefits.

2) The group has a good understanding that software flexibility drives their business agility. A majority, or 54 percent, say their lead time for making changes to their Salesforce orgs is between one day and one week. But we can also identify an elite group that operates even faster–21 percent say their lead time is less than a day, and 8 percent say it takes less than an hour. Taken together 83 percent can make changes in a week or less. That's far faster than historical averages but is it enough?

3) To back up the idea of an elite cadre forming, when asked how often these people would like to release new customizations to Salesforce, the numbers jumped. Fully 19 percent want to be able to deploy multiple times per day; another 16 percent would like to release between once per hour and once per day; and 33 percent, want to at least be able to release customizations once per day to once per week. This suggests that many organizations are still trying to reduce their development and deployment times.

4) Quickly delivering stable code repeatedly is one thing but just as important is security and eliminating errors; doing all of this requires management tools. Our survey population uses a variety of tools and strategies. A minority of 36 percent use Salesforce-specific commercial tools but 48 percent also use some open-source products and 26 percent use primarily in-house developed CI/CD toolchains. Many organizations use a combination of several tools though the larger elite organizations have standardized on a single vendor with a pre-integrated tool set to support the entire cascade of processes needed to deliver stable, secure and error-free systems ready for deployment.

5) Issues still arise. 43 percent said they have performance problems after a deployment up to 15 percent of the time. Only 2 percent said they had issues between 76 and 100 percent of the time but clearly, service declines are still an issue that needs to be dealt with. Fortunately, 21 percent said the time to restore service levels is less than an hour and 43 percent said rectifying issues caused by deploying new software could take up to a day. In all nearly two-thirds appear to have the round trip to restoring service in hand but it would be best if degradations didn't happen.

6) Significantly, the wide range of approaches indicated by tool choices suggests that best practices are still being worked out. Moreover, the management tools you use dictate some of your methodology choices. Since many tools are oriented towards code-based approaches that may not take into account the low-code or no-code approaches that Salesforce Lightning offers in addition to its full code approach, it is possible that users are still not receiving the full benefits of the Salesforce platform, despite other very good numbers. Also, there is still a diversity of approaches to tool selection and use. Some organizations use multiple tools from a variety of off the shelf, open-source, and in-house developed sources. They cobble together solutions to fit their needs but the elite, especially large enterprise users have settled on using pre-integrated suites of management tools, often from a single vendor.

7) Cross-tabulating the data shows that–measured by number of employees, number of Salesforce users, number of Salesforce orgs, or number of developers-as enterprises get larger, organizational friction slows the deployment cycle. But increasingly a Salesforce-DevOps strategy is emerging that relies on pre-integrated end to end solutions to drive better performance.

Survey

Our data is culled from a survey of 201 executives whose businesses use Salesforce Lightning. It shows how far these organizations have progressed toward having rapid and efficient software development and deployment processes, an effort that remains ongoing though there has been significant progress.

We researched how frequently they deploy new developments and changes to their Salesforce orgs, the number of orgs they operate, their error rates, and more. Our conclusions indicate how leaders in this space use DevOps with Salesforce and suggest some emerging best practices.

ROI

DevOps brings into sharp focus the differences between conventional application development and modern development through the lens of ROI. Understanding this point helps put everything else into perspective. Business software ROI has usually been a function of the benefits derived from automation which can reduce costs from labor and materials, sometimes offloading part of a business process to the customer.

ROI in the era of the digital disruption is different though. While business software still automates business processes, the return is more likely to be derived from external factors than reduction in internal expenses. The digital disruption centers on leveraging customer data through, among other things, analysis and machine learning and often the output of these activities is information that must be fed back into a process for it to be useful. Modified business processes drive additional business or engage customers appropriately so that they return. So it's critical that applications are able to adapt to changing market conditions.

Organizations that successfully navigate this event-chain can be thought of as agile and increasingly that agility relies on software flexibility. A business that can adjust its running applications daily, if needed, is thought to be more agile than one that needs a week to do the same thing. The second organization is more agile than one that might take between a week and a month to achieve the same result, and so on.

Obviously, the faster a business can adjust the behavior of its customer-facing systems, the more competitive it can be. It's this competitiveness that forms the new understanding of value because it directly impacts the organization's ability to spin up a salient that can effectively compete for new business opportunities. This different approach captures value not as conventional ROI but through Time to Value (TTV).

Imagine a business with a customer-facing suite encountering a new business opportunity. Many opportunities can be engaged with no or only minimal changes to their business processes and the supporting software. But in cases where that's not so, and where the business may have to support a new line of business, for example, the business must adjust or it will not be competitive. For instance, an enterprise deciding to get into a new line of business might find its conventional software development and deployment strategy almost useless for dealing with different business practices in sales or billing. A classic example is moving from a product to a subscription service orientation. Various workarounds can be tried but they might not scale and soon a business will need to choose between its software development strategy and its new line of business.

In such a case, the legacy software's value to the business declines significantly if it takes many months and dollars to retrofit a software suite. But if that software was based on a DevOps strategy and platform technology that incorporated multiple subsidiary functions like AI, security, workflow, commerce, integration, and, importantly, code generation, the transition would likely be quicker and less costly. That's where time to value comes in and that's why software flexibility driving business agility is so important to the digital disruption.

A modern business will likely see its IT and related functions not as cost centers but as profit generators and TTV is critical to this shift, therefore so is the software platform and the techniques for using it. Return on investment happens every time a software change or update is quickly, accurately, and successfully propagated to the business. It is derived from keeping customers happily engaged, returning, buying more, or simply acting as unpaid brand ambassadors.

To be clear, platform technology is important, but it represents half of the modern software development and deployment paradigm. The other half involves how organizations organize themselves to engage with the technology and that's where DevOps comes in.

So with this as preamble let's look at how Salesforce customers are leveraging the Lightning platform and DevOps. This is a relatively recent field and our results show a situation that is very much a work in progress.

Survey basics

There were 201 completed responses to our survey from companies of all sizes (Figure 1) that use Salesforce Platform, Lightning, including 22 percent from businesses with 10,000 employees or more. Many industries were represented including Technology (22 percent), Financial Services (16 percent), Retail/consumer/eCommerce (16 percent), and Industrials and Manufacturing (14 percent) led the list. Healthcare and Pharmaceuticals made up 8 percent of the results.

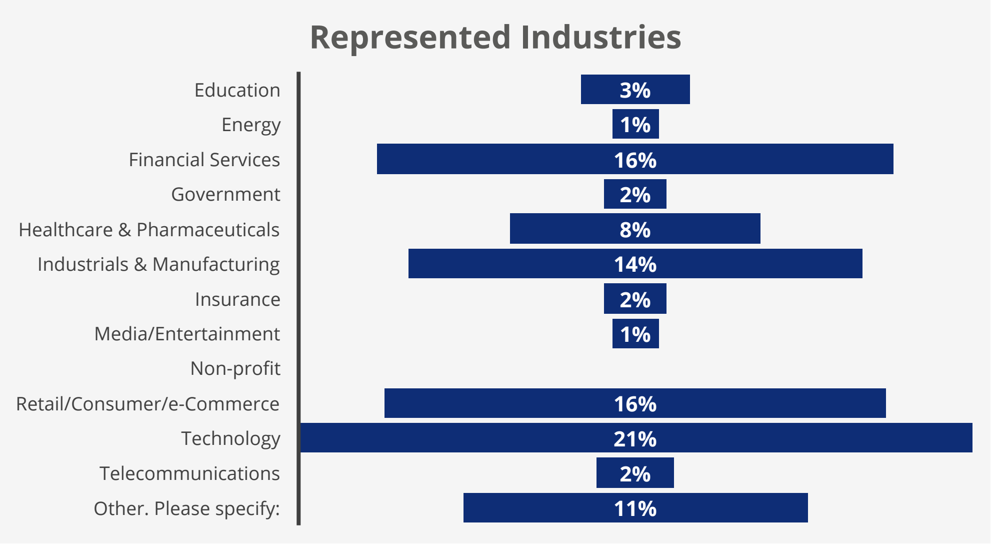

Figure 1 Industries represented in the data.

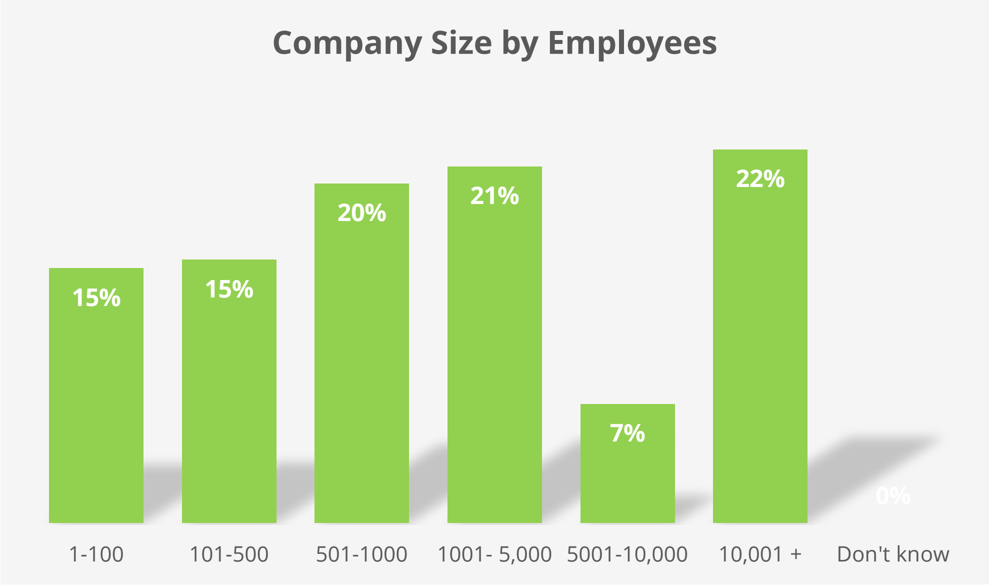

Figure 2 Survey population by number of employees.

Nearly half of the respondents, or 48 percent, were C-level executives including 23 percent CEOs while 40 percent represented as upper management.

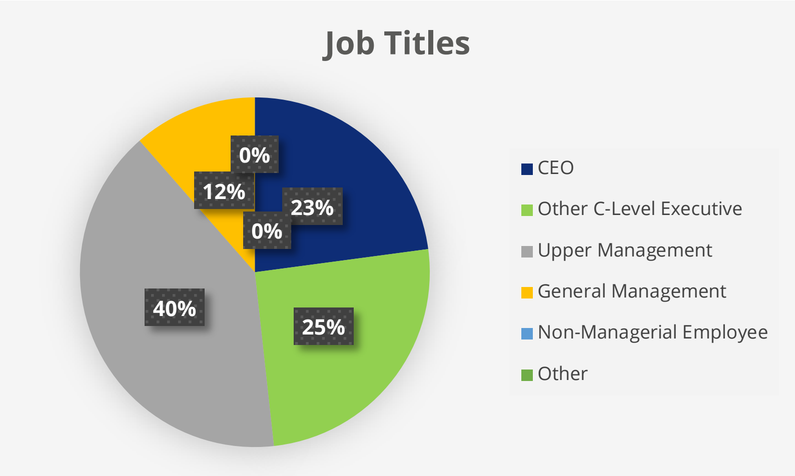

Figure 3 Respondents by job title

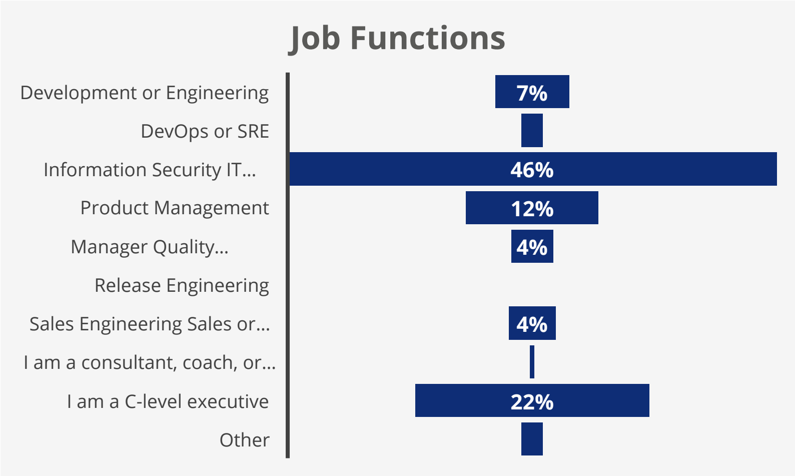

Figure 4 Job functions of respondents.

Most worked around Information Security and operations, (46 percent), Product Management (12 percent), and Development or Engineering (7 percent).

The average organization has been using Salesforce since 2011 plus or minus 5 years indicating a solid base of experience. There is no correlation between length of time in use and DevOps performance.

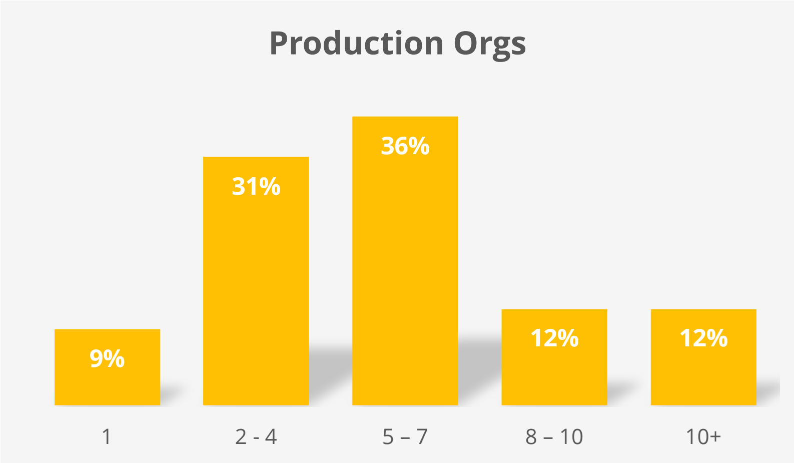

Figure 5 Production Orgs.

Two-thirds of the businesses in this survey support between 2 and 7 Salesforce Orgs. Importantly, half say that every developer has a private development environment or "sandbox" leaving the other half with some other arrangement. Either way though, a significant challenge arises when organizations try to synchronize across orgs. Add to this the complexity of a homogenous team trying to synchronize both declarative and code-based development and you can see that they all need both a disciplined application development lifecycle and management tools.

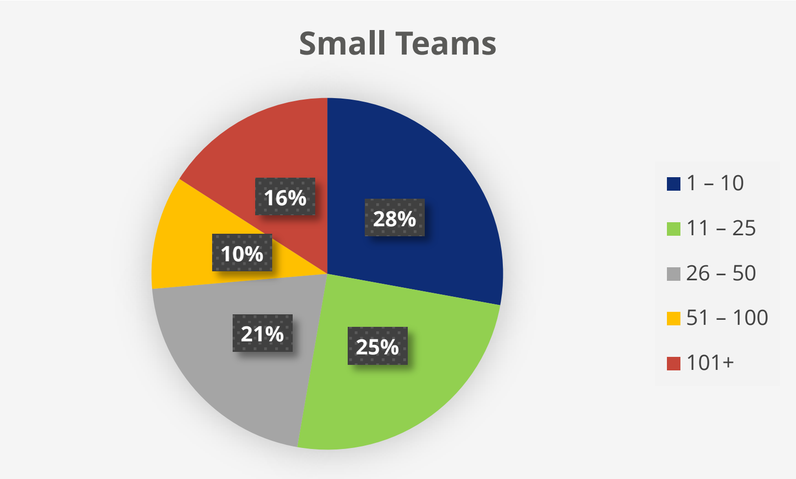

Figure 6 Small teams of developers for most businesses.

And they do it with small teams of developers. Though "small" is a relative term, older coding-based development may have used hundreds or even thousands of developers so the Salesforce approach might look small in comparison even though it is big by Salesforce standards.

With that we can begin to analyze how these organizations work.

If modern business requires speed to change, then the population of survey takers may be among the top performers. Their lead times for implementing changes to Salesforce orgs is impressive–as little as hourly to a mean of between a day and a week, with 21 percent saying they can make changes in less than one day and 8 percent doing it in less than an hour.

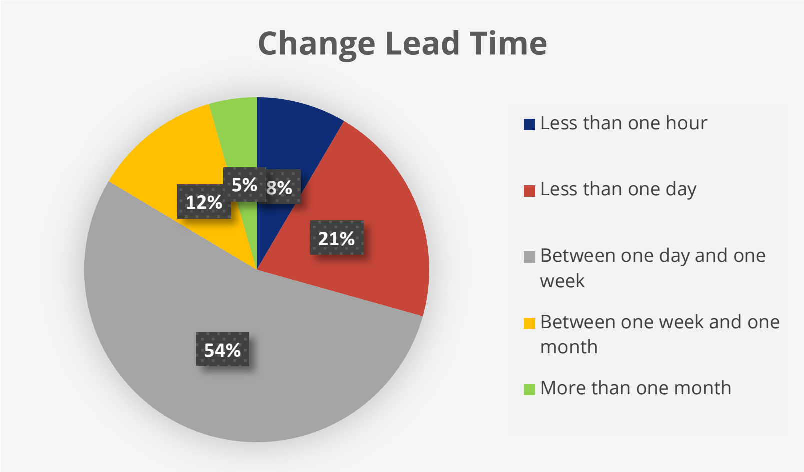

Figure 7 Fast time to make changes.

They can also implement a feature about as fast with 53 percent saying deployment can take between a day and a week and the same 29 percent saying they can implement changes in less than one day.

It shouldn't come as a surprise that 25 percent say they can release to their Salesforce production org multiple times per day. The fact that these businesses have multiple orgs makes this less daunting. But in a world where speed equates with software flexibility and drives business agility there is demand for more speed.

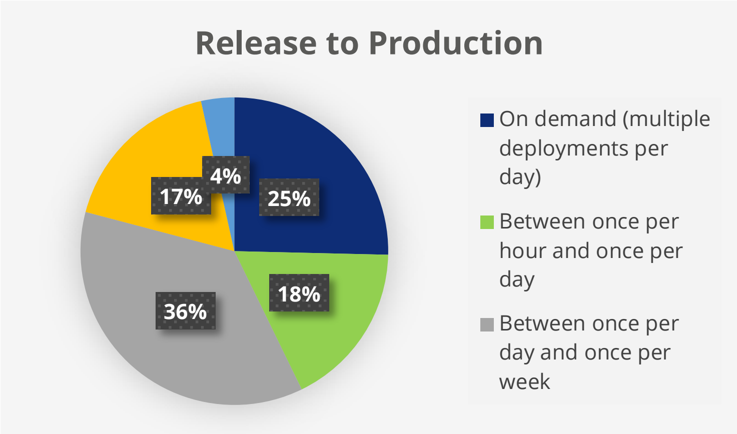

Figure 8 Fast release cycles.

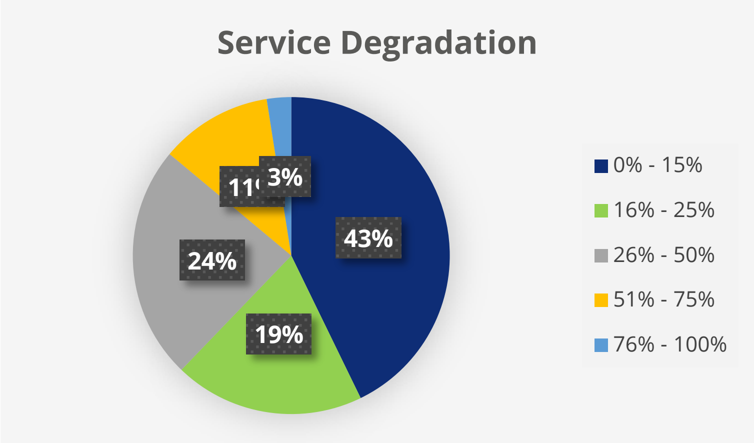

Figure 9 Service degradations can impact productivity.

Perhaps it is surprising that a release to the production org can relate to a service decline, but 2 percent reported a problem at between 76 percent and 100 percent of the time. A plurality of respondents, 43 percent, said a problem occurred up to 15 percent of the time and the vast majority or 86 percent said service degradations happen less than half of the time. This is an important snapshot of the state of the industry. Speed of delivery slightly exceeds stability of releases indicating a need to bring the two metrics more in alignment.

Time to restore service after a service incident or a defect that impacts users occurs (e.g. unplanned outage, service impairment) is typically quick, nearly two-thirds or 64 percent said it takes less than a day.

There doesn't appear to be consensus on DevOps tools and many tools are developed in-house. Using self-developed tools is fine and many organizations have built quality solutions for their own needs. But the existence of so many in-house developed tools makes it difficult to identify standards that can lead to determining best practices. Organizations that develop their own tools do so to support their specific work processes, so as much as anything this data suggests that it's still early days for DevOps use.

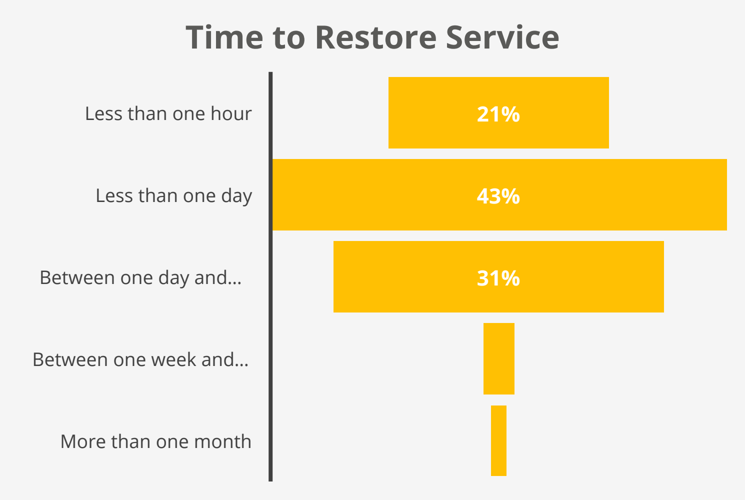

Figure 10 Restoring service is usually quick.

The top three DevOps tool vendors in this study are Flosum, ClickDeploy, each with 33 percent and Copado with 30 percent; however, many organizations may use more than one tool in conjunction including 48 percent using open source tools.

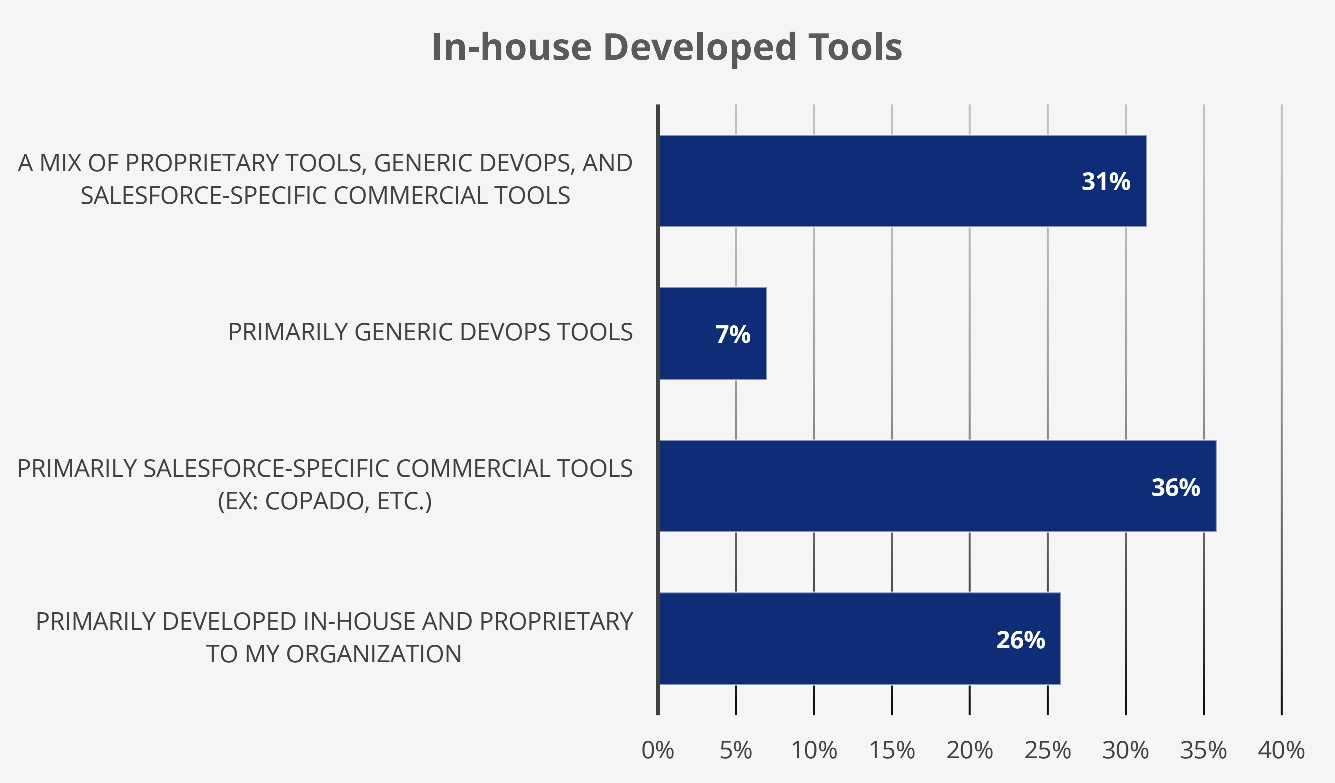

Figure 11 DevOps tools represent a mixed bag.

Best Practices

There is high correlation for automating development and operations practices between leading users of Lightning, for instance,

1) A strong majority (60 percent) say each developer has a private development environment.

2) Also, 77 percent say they use version control to store code and click-based Salesforce customizations.

3) Most synchronize their development environments with the latest changes from other teams with 41 percent doing this on-demand or at most once per day and 42 percent saying they do this between once per day and once a week.

4) 75 percent say changes made in version control trigger automation tests.

5) 87 percent have confidence that when automated tests pass the software is ready for release. However, meta-analysis of the data strongly suggests that the greater a team’s confidence in their tests, the higher their change failure rate. Skeptics who were neutral on this question experienced a 40% lower change fail rate than those who expressed strong confidence in their tests.

On the other hand, there is little consensus on deploying to salesforce instances.

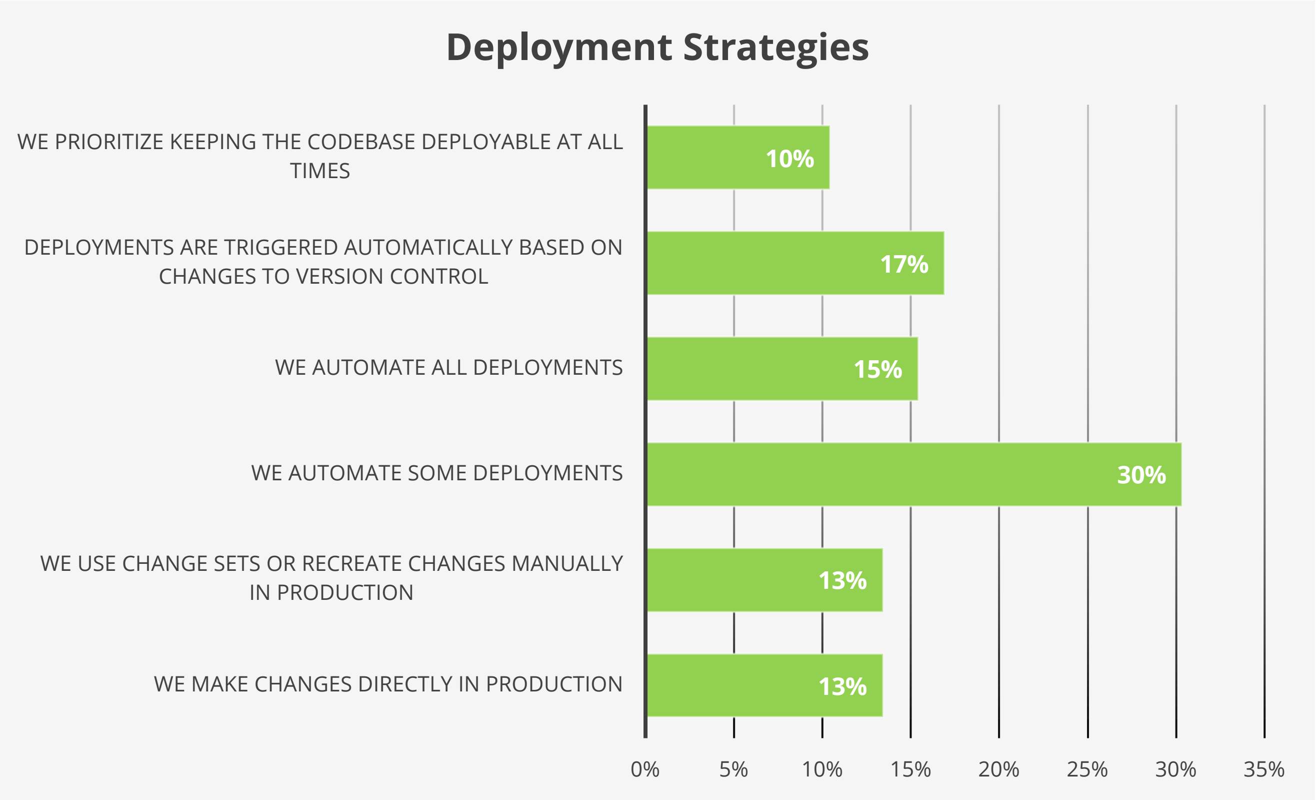

Figure 12 Deployment strategies for deploying to Salesforce instances.

A plurality of 45 percent automates some or all deployments, but other approaches are also frequently used so that there is no consensus or standard.

Business Benefits

Our survey population considered various approaches to improving their businesses through the software development and deployment process. The digital disruption plays an important role in how managers think about how to improve their businesses. Speed to market and developer productivity leading their wish lists which indicates a good understanding of the relationship between business agility and software flexibility.

The survey represents a reasonably satisfied group of executives whose satisfaction may stem primarily from comparisons with earlier development and deployment approaches. It's a reasonable assumption that the major benefits from the Salesforce Lightning-based approach can be traced to having Lightning generate much of the code through low code and no code strategies and not because development organizations are working smarter. At least this is indicated by data for larger organizations.

As noted in Key Finding 7, measured by number of employees, number of Salesforce users, number of Salesforce orgs, or number of developers, larger enterprises generate more organizational friction that holds up deployments.

One crosstab will make the point.

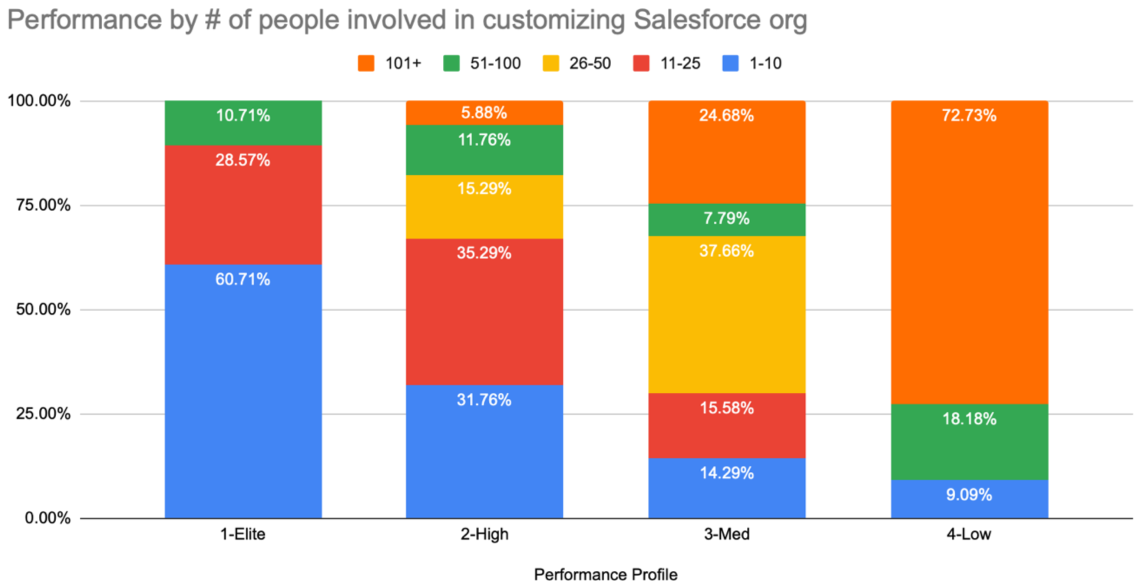

Figure 13 Organizational performance declines with size.

Figure 13 shows that when organizations are rated by their DevOps performance, larger organizations make up nearly three-quarters of the lowest performing group. Conversely, small development organizations make up more than 60 percent of the elite users. This clearly indicates a reciprocal relationship. Whether this discrepancy describes the shorter lines of communication inherent in smaller groups or something else, requires further work.

The cross tabs suggest that larger organizations are not reaping the same benefits as smaller businesses and that more can or should be done. Yet when asked if their Salesforce Dev Ops strategy prepares them for future digital disruption, big majorities said yes, again perhaps indicating satisfaction with a primarily technology (rather than methodology) based improvement from using advanced tools like Lightning.

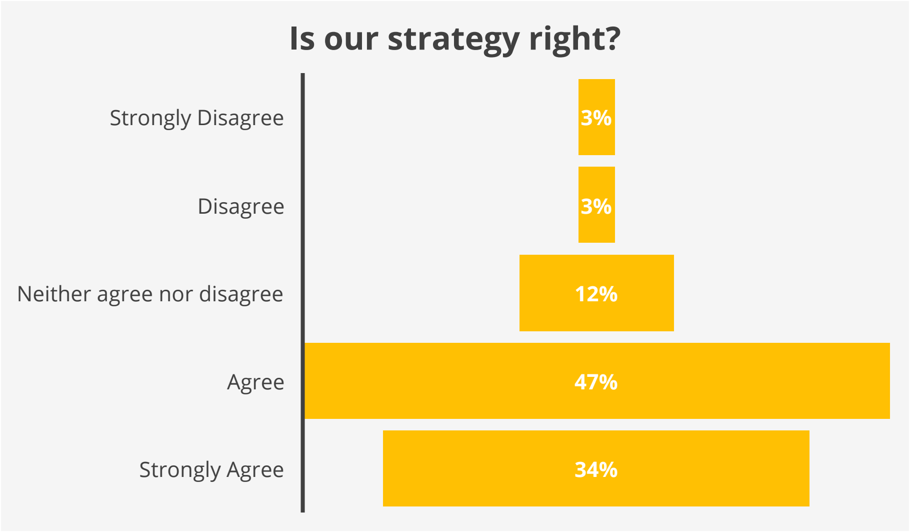

Figure 14 Getting the DevOps strategy right might take more effort

Also, when asked if their Salesforce DevOps strategy is as mature as the DevOps strategies for other technologies in their companies, 77 percent either agreed or strongly agreed. Given the cross tabs, we'd suggest that this might be a blind spot and that more can be done.

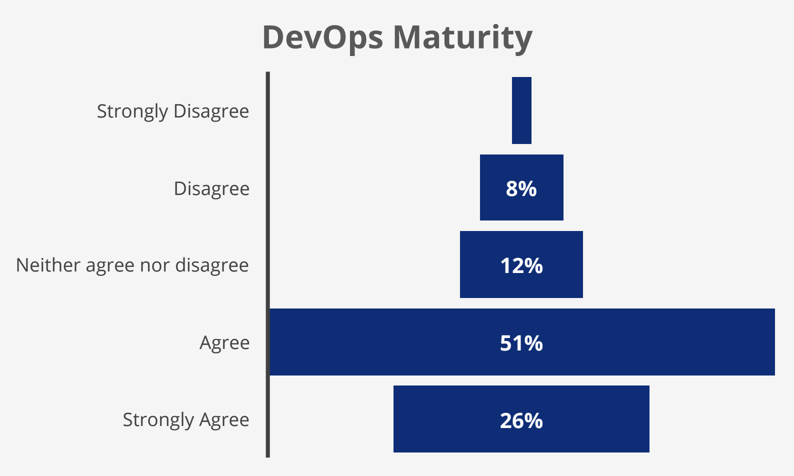

Figure 15 Our analysis suggests that more can be done to improve DevOps results. Getting that word to executives appears to be the next task.

Plans for DevOps

Already 35 percent of the organizations surveyed can see business benefits of one million dollars or more with 17 percent seeing $5 million or more. The top three future plans expressed by our group included

1. First place–speed to market and increased developer productivity tied for first place
2. Second place–making apps more customer relevant and continued innovation also tied
3. Third place–a tie between employee satisfaction and release quality

This is broadly in line with the often-stated objectives of a digital disruption. A business must have the capacity of responding quickly to customer needs before it can consider customer relevance and innovation. With those capabilities in place a business can turn its attention to employee satisfaction and with all of that attention to release quality comes into focus.

So how are they doing?

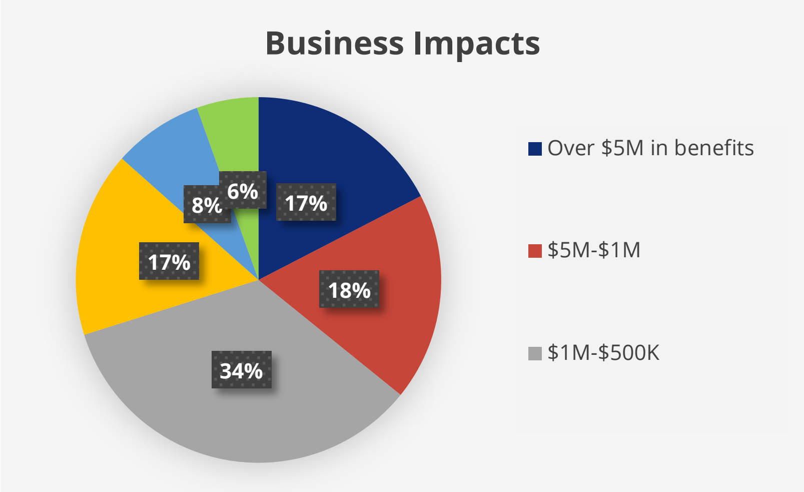

Figure 16 the business benefits of Salesforce Lightning, with a DevOps framework are real and can be substantial

Conclusions

On the Salesforce Lightning platform DevOps is delivering benefits but more needs to be done. Much of the business benefit from a DevOps strategy so far seems to derive from the benefits provided by raw technology.

Cross tabs show that by any measure of company size including number of employees, number of users, number of people customizing the Salesforce Org, or number of production orgs, smaller organizations perform better in a DevOps framework than their larger cousins. But the larger businesses have much to gain from using modern tools and methods.

More benefits can be derived as organizations leverage DevOps development tools to ensure speed, product integrity, and security. In the largest organizations this will require better organization of time and resources.

There can be many reasons that larger businesses lag in realizing the full benefits of the Salesforce platform. Greater size increases the amount of organizational friction encountered by developers with a mission to get apps out the door. Larger businesses have much more need for streamlined business processes that a platform technology can support. They need DevOps to streamline the process of getting well developed and error-free code out of IT and into production. That's not what we see, however. This strongly suggests that we are still in the early part of realizing the benefits of platform-based application development. Nevertheless, by virtue of focusing on one development framework, this research, shows where the bright spots are and where we can learn the most.

We recommend users of Salesforce Lightning or any other software platform identify best practices that implement a DevOps strategy and also to ensure that the organization uses a pre-integrated tool chain that supports the end to end software development process.