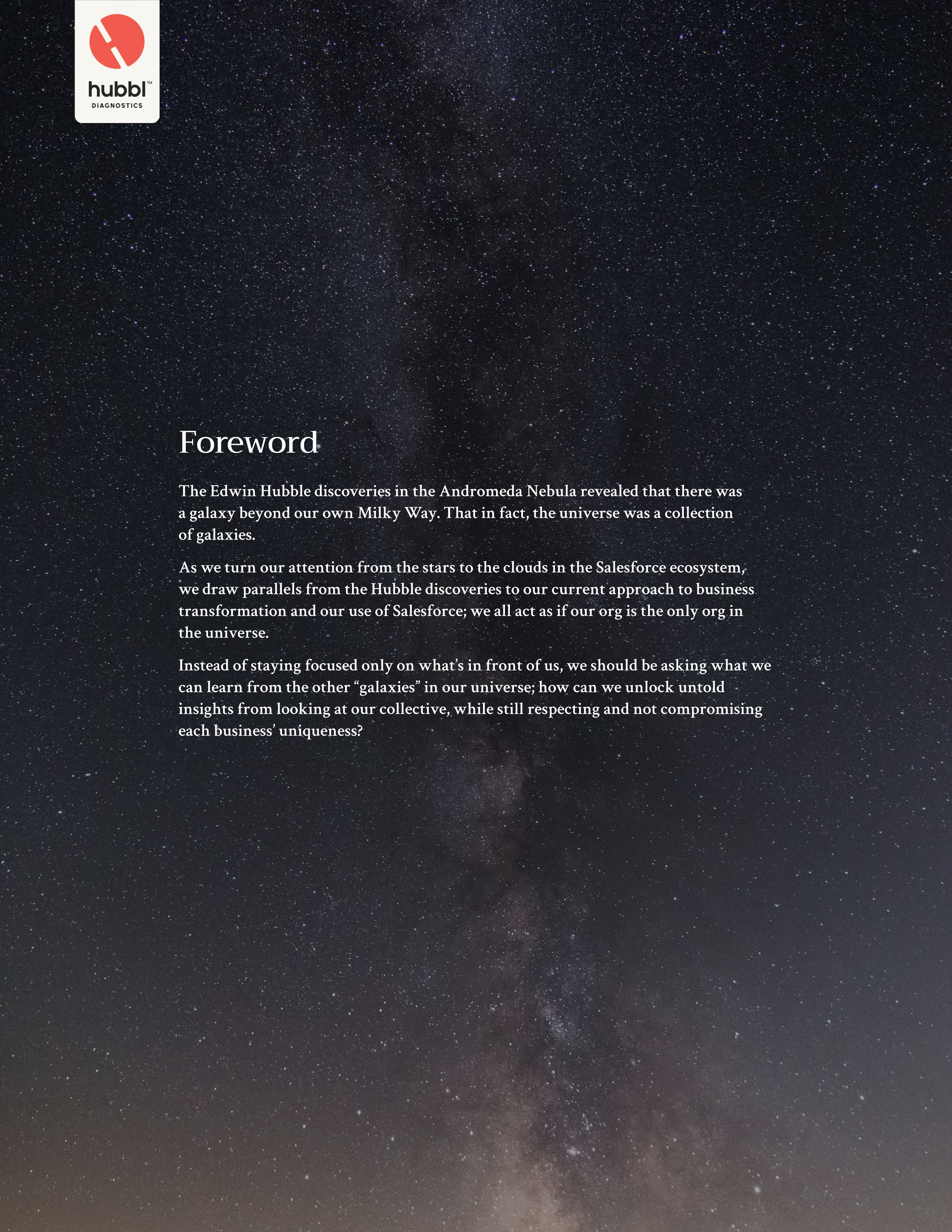
HUBBL DIAGNOSTICS

Benchmark Report 2022/23

Salesforce Org Automation Analysis | First Edition

A detailed look into hundreds of Salesforce orgs to uncover the health, complexity, and utilization trends across the Salesforce ecosystem.

***Mike Bogan** | Director Product Strategy, Uncommon Purpose*



Foreword

The Edwin Hubble discoveries in the Andromeda Nebula revealed that there was a galaxy beyond our own Milky Way. That in fact, the universe was a collection of galaxies.

As we turn our attention from the stars to the clouds in the Salesforce ecosystem, we draw parallels from the Hubble discoveries to our current approach to business transformation and our use of Salesforce; we all act as if our org is the only org in the universe.

Instead of staying focused only on what's in front of us, we should be asking what we can learn from the other "galaxies" in our universe; how can we unlock untold insights from looking at our collective, while still respecting and not compromising each business' uniqueness?

Executive Summary

The state of automation in a Salesforce org is of critical importance to businesses utilizing this powerful multi-tenant data architecture. Yet it's surprisingly difficult to rapidly evaluate all of the automation in an org — a task often completed by manually clicking through setup.

Currently, there's no way to view all automation for a single object in a single view. There is also no means to measure the state of automation in one org compared to another. As workflow and process builder feature retirements loom, it's critical that businesses evaluate their position in the ecosystem relative to their peers with regards to automation. Businesses that manage their migration to flow and triggers smoothly will reduce their technical debt, increase their agility, and ultimately outcompete in the marketplace.

Here we report our research exploring the metadata of hundreds of Salesforce orgs. Our results show:

> The average org has about 12% of declarative automation migrated to flow, with 90-95% of the remaining processes and workflow unmanaged.
>
> On average, every object with at least one active workflow rule has six workflow rules on it. This means admins looking to follow best practice will likely need to consolidate workflow since the Salesforce workflow migration tool executes a 1:1 migration.
>
> The average org often deviates from the best practice of a single trigger per object for the same event.

We also introduce several new org intelligence concepts including:

> **Risky objects:** To better interpret the state of record-triggered automation, we introduce "risky objects" and show how to quickly identify them.
>
> **The Hubbl Diagnostics Index:** We built a state of automation benchmark that provides a quantitative and objective weighting of an individual org's complexity relative to the Salesforce ecosystem.

For Salesforce consultants, Hubbl Diagnostics Index scores have proven valuable for resourcing and pricing decisions. For Salesforce admins, the scores provide a new perspective and understanding of where their organization fits in comparison to the ecosystem.

Since our results show that the average org has at least 125 hours of technical debt analysis work ahead of itself, it's imperative that organizations catalog the state of automation in their org, prioritize, and plan their technical debt remediation efforts.

What's Inside

Foreword 2

Executive Summary 3

What's Inside 4

Introduction 5

Results and Discussion 6

- THE BASIS OF SALESFORCE ORG BENCHMARKING: UNDERSTANDING MULTI-TENANT ARCHITECTURE 7
- INSIGHTS FROM AN AGGREGATE VIEW OF THE SALESFORCE ECOSYSTEM: RECORD-TRIGGERED AUTOMATION 8
- DECLARATIVE AUTOMATION FEATURE RETIREMENT: HOW MUCH WORKFLOW AND PROCESS IS IN THE TYPICAL SALESFORCE ORG? 10
- TOO MANY TRIGGERS: ARE YOUR OBJECTS FOLLOWING BEST PRACTICE? 15
- FIND YOUR RISKY OBJECTS 16
- COMPARE AND RANK SALESFORCE ORGS RELATIVE TO EACH OTHER 18
- ORG SIZE ≠ SIZE OF EFFORT TO ANALYZE TECHNICAL DEBT 19

Key Actions 21

- 1. SCAN: UNDERSTAND 'WHAT YOUR SALESFORCE ORG IS' 21
- 2. ANALYZE: WHAT DOES THE DATA MEAN? 21
- 3. PRIORITIZE: WHAT TO DO, WHEN 21

Methods 22

About 23

Introduction

The proliferation of cloud software offerings that are more intelligent, targeted, and competitive, are creating an ever-increasingly complex technology ecosystem. Paired with the increasing need for businesses to adapt to changing macro conditions, like new working models and increasing global competition, buyers are finding themselves spending and building more. Bessemer's State of the Cloud 2022 Report illustrates that 92% of CIOs plan on spending more or the same amount this year.

Whether building on pre-existing infrastructure or investing in new platforms, it's becoming increasingly difficult to manage the growing complexity of our systems. Complexity reduces our ability to ensure a strong return on investment and positive impact on our customers and places greater strain on our most critical resources: our people.

For more than a decade, our team has worked with clients who run their business on the Salesforce platform. During this time, not only has our collective approach to cloud technology changed drastically, but so have the leaders and business objectives set to those solutions. Naturally, this evolution results in customizations, adaptations, new best practices, more data, new users, and so much more. The connective tissue of our systems continues to evolve with changing business requirements.

Hubbl Diagnostics is on a mission to enable our ecosystem to easily and safely get ahead of the ever-expanding org complexity through actionable insights. We've developed a robust Salesforce org intelligence platform that allows us to quickly analyze the metadata in a Salesforce org and compare it against an aggregated view of others.

We're thrilled to introduce the first edition of the Hubbl Diagnostics Benchmark Report to the Salesforce ecosystem, a collective group of innovators, builders, dreamers, and leaders who have evolved cloud technology in leaps and bounds over the last 20+ years. This report draws data and analysis from two key areas:

> **The first and more anecdotal area** is from our own experience of Salesforce consulting, having worked with over 2000 customer orgs over more than fourteen years.
>
> **The second, which is more quantitative,** is from the aggregated metadata scan results collected from hundreds of Salesforce orgs. Analysis of the metadata scans has helped us uncover how organizations are actually using the Salesforce platform and allowed us to baseline the ecosystem.

With the data presented here, we've begun to unravel a view into the Salesforce ecosystem that's never been seen before; one that tangibly relates the objects, triggers, and workflows in your org, to better business results for you and the entire Salesforce ecosystem.

Results & Discussion

The average org only has about

12%

of **declarative automation** migrated to flow

19 **million hours**

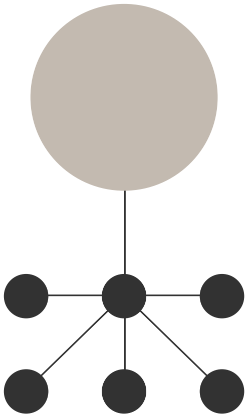
of workflow rule migration

&

5 **million hours**

of process builder migration

estimated effort required across the entire Salesforce ecosystem



On average, every object with at least one active workflow rule has

6 **workflow rules on it**

The average org has more than

125 **hours**

of technical debt analysis effort lurking inside it

90–95%

of processes and workflow are

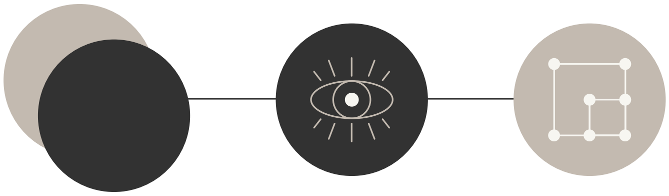
unmanaged

Most orgs have at least one object with

more than one trigger for the same event

$3B

estimated **in technical debt remediation professional services** required within the Salesforce ecosystem



Risky objects should be

uncovered, monitered, & managed

appropriately, using **Hubbl Diagnostics**

The Basis of Salesforce Org Benchmarking: Understanding Multi-Tenant Architecture

Salesforce's success stems from its ability to standardize functionality (common metadata) hosted in a polymorphic multi-tenant platform and provide it to customers via licenses. In multi-tenant database systems, customers may share database resources. The databases themselves are typically shared, and each tenant (customer) is associated with a unique identifier that's used to identify the rows and columns that belong to each tenant. Each tenant can then provide their own custom data to populate the shared objects in the database (referred to as "standard" objects).

Tenants can also define custom objects and custom fields (tenant-specific metadata). Users of the multi-tenant system develop applications that interact or integrate with the multi-tenant system and utilize the data from an associated tenant space. Salesforce, the host of the multi-tenant architecture, groups together collections of objects to form classes of function (referred to as "clouds") that solve a particular business challenge or data architecture requirement. Tenants utilize the common metadata in the clouds to varying degrees based upon their business needs.

Many businesses start with "out-of-the-box" functionality (common metadata) and then customize. Poorly managed customizations lead to technical debt that can hinder a business' ability to innovate. Salesforce's meteoric growth has meant that both organic and acquisition-based growth of business functionalities in the form of "clouds" (e.g., Sales Cloud), constantly increases the number of standard objects available to customers in the multi-tenant database (Figure 1).

AUTHOR'S ASSESSMENT

Multi-tenant architecture is incredible as it means that thousands of organizations utilize the so-called standard objects as the basis of their business and all rely on that exact same core functionality. What defines a business is how it deviates from the core functionality. It is basically a business printer.

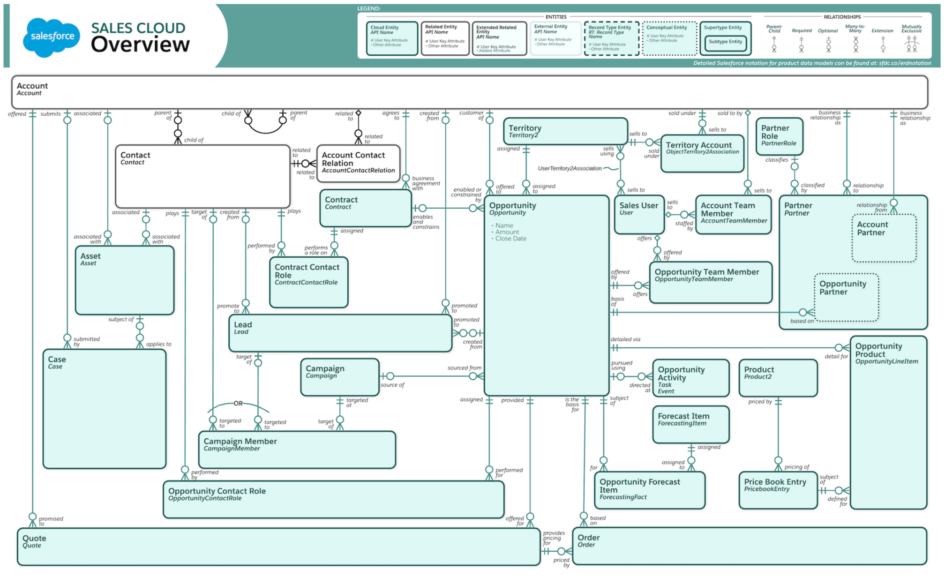


Figure 1 Sales Cloud data model from architect.salesforce.com.

Keeping pace with the growth in standard functionality is challenging for Salesforce consultants and administrators. For example, the data models for the Sales Cloud features of Salesforce show the already complex nature of common metadata without customizations built on top. The complexity required to manage increases as businesses "land and expand", meaning they start with one cloud and then license additional functionality later. As feature adoption increases, so does the number of data models and expertise required to manage them. Sometimes custom builds of similar functionality are attempted, leading to even more technical debt once the standard object capability is licensed.

These concepts form the basis of the benchmarking we have achieved with Hubbl Diagnostics.

In this report we describe our research analyzing hundreds of orgs through the lens of automation layered on standard and custom objects. As more orgs are scanned by Hubbl Diagnostics, our aggregate analysis grows, building a more comprehensive image of the Salesforce ecosystem and providing deeper insights for all participants.

AUTHOR'S ASSESSMENT

Salesforce is now >20yrs old and technical debt is growing. Many early orgs highly leveraged soon to be deprecated automation. In a way, the deviation from core functionality that defines the business needs to be course-corrected to spur the next wave of agile innovation.

Insights from an Aggregate View of the Salesforce Ecosystem: Record-Triggered Automation

In our experience, we've found that one of the most straightforward ways to think of an org is as a collection of data tables (objects) and their related automated code (e.g. flows, triggers). With this in mind, the best way to immediately understand what an org is doing, is to also view it from the object level, with its various layers of automation applied. The process can go much deeper with interfaces, APIs, and more. However, this view has proven particularly useful in org discovery scenarios, such as when consultants are working with a new customer or an admin has taken over an org for the first time.

But, let's start bigger. Edwin Hubble taught us to look beyond our own galaxy, our own Salesforce org. What do we see if we look at the aggregate results of automation built on top of standard objects in hundreds of orgs?

While what we find for relative automation tallies on objects is not entirely unexpected, this aggregate view helps us uncover trends across the ecosystem. Many of the core standard objects we saw in the Sales Cloud and Service Cloud data models of Figure 1 lead the pack with regards to automation built on top of them (Figure 2), (e.g., Case and Opportunity). The sizable automation on the User object should be approached cautiously. We can also see outliers in this view, like the large number of approvals for some objects (Contract, Quote, and Knowledge) compared to other objects that carry more automation of other types. Also notable is the large number of sharing rules on the Account object. When sorted by automation type, the object automation table provides the priority list for tackling unruly technical debt.

This data creates the basis for the benchmarking, and the object automation view we introduce in Figure 2 will prove even more valuable when looking at each individual org in the collective. These results also provide administrators and consultants with a starting point for identifying highly automated objects and potential technical debt in need of remediation. Additionally, given that these standard objects form the foundation of core Salesforce platform usage and business process for thousands of organizations, the associated risk of unmanaged technical debt is higher and more costly.

AUTHOR'S ASSESSMENT

Our results clearly show there is a vast amount of workflow and process builder automation built on standard objects in the ecosystem. But, we must then ask, *what is in an average org?* How much effort can we expect to spend on migration of this legacy declarative automation?

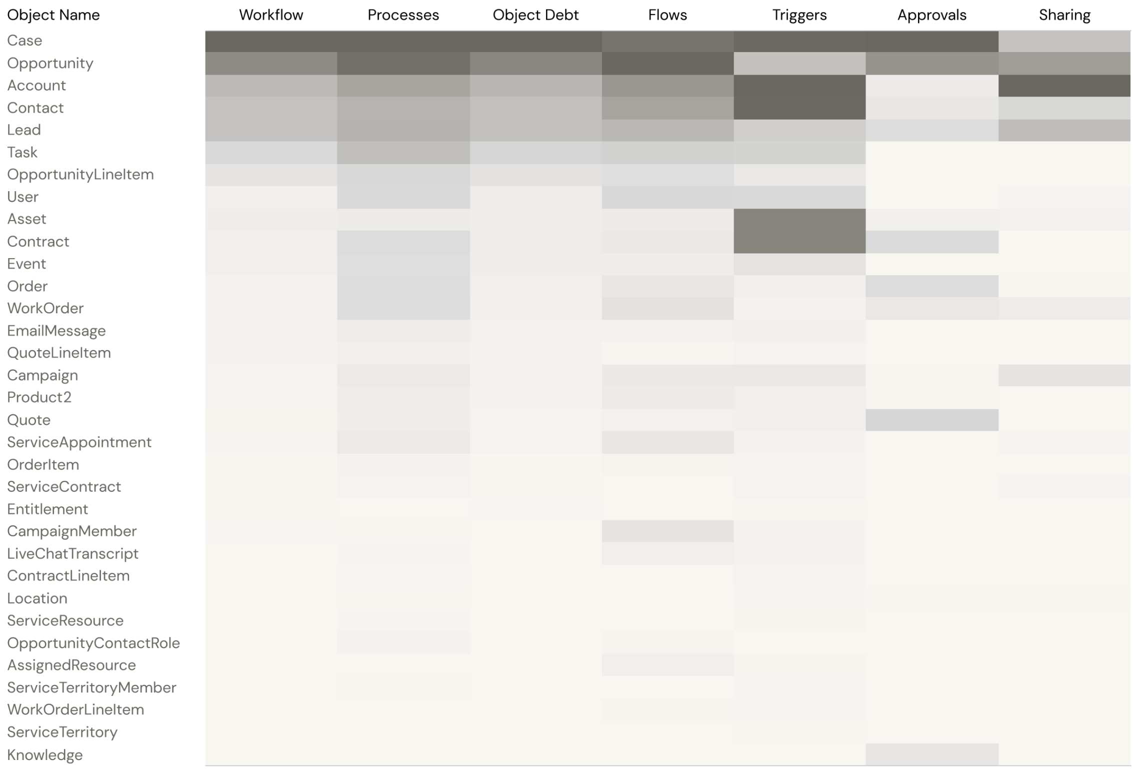


Figure 2 Record-triggered automation for standard objects detected in our metadata sorted by Object debt, the sum of workflow and processes on an object. The darker the color, the more automation present.

Declarative Automation Feature Retirement: How Much Workflow and Process is in the Typical Salesforce Org?

One of the most anticipated announcements in the Spring '22 release was the updates to Salesforce's point-and-click automation tool, Salesforce flow. With these enhanced features, Salesforce also announced the retirement of workflow rules and process builder creation.

What does this mean for the average Salesforce org? We used our Hubbl Diagnostics dataset to find out.

Declarative automation is the workhorse of the typical Salesforce Administrator. Flow is now considered the premier declarative automation tool in the Salesforce ecosystem. However, we found that of the declarative workflow, process, and flow automation across the ecosystem, most orgs have legacy automation built in workflow and process builder. Only 12% of the declarative automation is built in flow.

We found that 100% of orgs have active workflow that will eventually need to be migrated, with the average org having 127 active workflow rules (Figure 4). In our sample, several orgs have >500 workflow rules. Considering that these may skew the average, it is worth noting the median number of workflow rules for an org is 71.

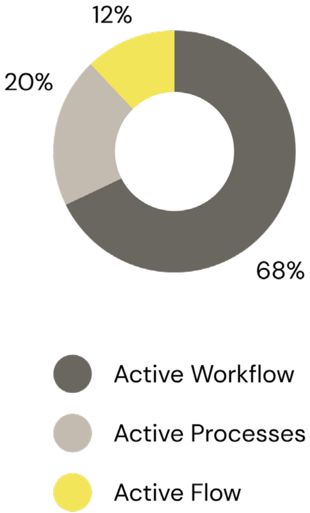


Figure 3
State of workflow, process, and flow automation in the ecosystem: **88% of the declarative automation needs to be migrated to flow or triggers.**

Based on analysis of our aggregate data, we predict the average Salesforce org will:

Have only
12%
of declarative automation migrated to flow

Have approximately
150
active workflow and processes to migrate

Include approximately
50
inactive workflow to be evaluated for removal

This will require at minimum ~150 hours of effort per org, depending on code complexity

Deprecation Timeline: Workflow and Process Builder

→ Spring '22: launch workflow migration tool—complete

→ Winter '23: launch process builder migration tool

→ Winter '23: disable creating new workflow rules

→ Spring '23: disable creating new processes

→ Estimated '24: workflow and process end-of-support

Workflow rules migration to flow can sometimes be a simple one-to-one migration. This scenario is aided by the Salesforce workflow migration tool, which is designed for 1:1 migrations. However, it's often much more complex. As we see in Figure 5, when objects have workflow, they rarely have just one. Across the ecosystem there are typically ~6 workflows on objects with workflow. Best practice recommendation for flows is to keep the amount of flows to a minimum. The migration can be taken as an opportunity to consolidate multiple workflow rules on a single object into a single flow or trigger, so the order of operations can be better controlled. To reconcile this gap, an admin likely needs to work together with the appropriate business unit stakeholders to validate the business process. If multiple flows remain, Salesforce's Flow Trigger Explorer can help control order of operations.

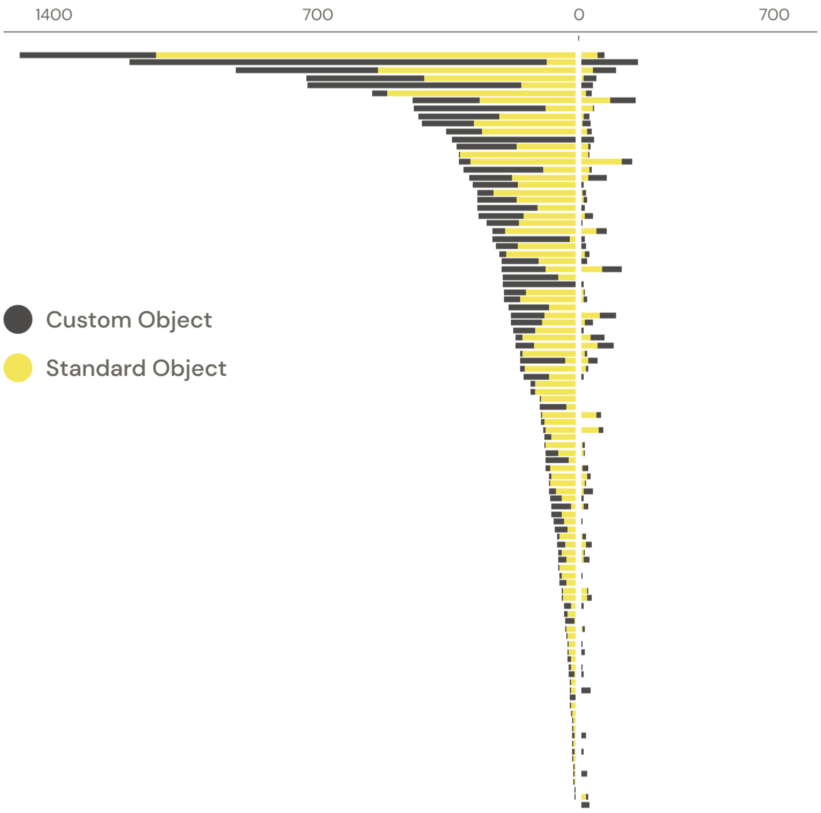


Figure 4
Count of workflows (left) and process (right) on standard and custom objects by org.

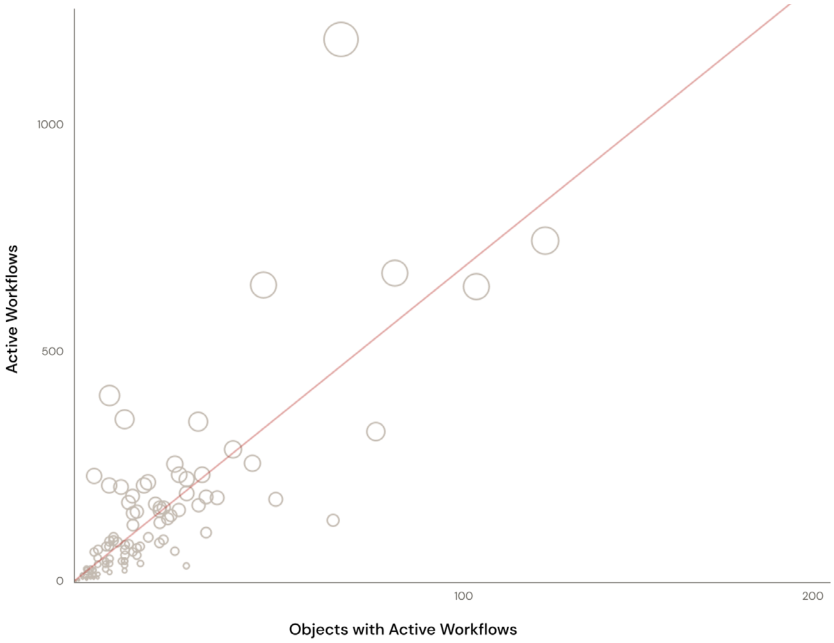


Figure 5 Total active workflow rules vs. objects with active workflows.

AUTHOR'S ASSESSMENT

This presents a challenge for the ecosystem; the high volumes we see across our sample suggest there will be a large impact on maintenance timelines to address this technical debt and increase business agility. When scanning an individual org, the larger the number of workflow and objects with workflow, the higher risk an org is to technical debt issues. We recommend you inventory your workflow and process immediately to plan accordingly.

Understanding the workflow migration effort across the ecosystem.

How much effort is all of this migration going to take? Let's assume it takes 1 hour of effort to migrate each workflow rule, taking into account admin development effort and stakeholder conversations. At 127 workflow rules per org and ~150,000 Salesforce orgs in the ecosystem, this suggests that there is roughly 19 million hours of workflow rule migration effort within the entire ecosystem. Using the median value of 71, that still represents 11 million hours.

> The burden of this effort will land on admins and consulting partners; therefore we recommend identifying strategies to classify the complexity of your workflow rules to prioritize effort.

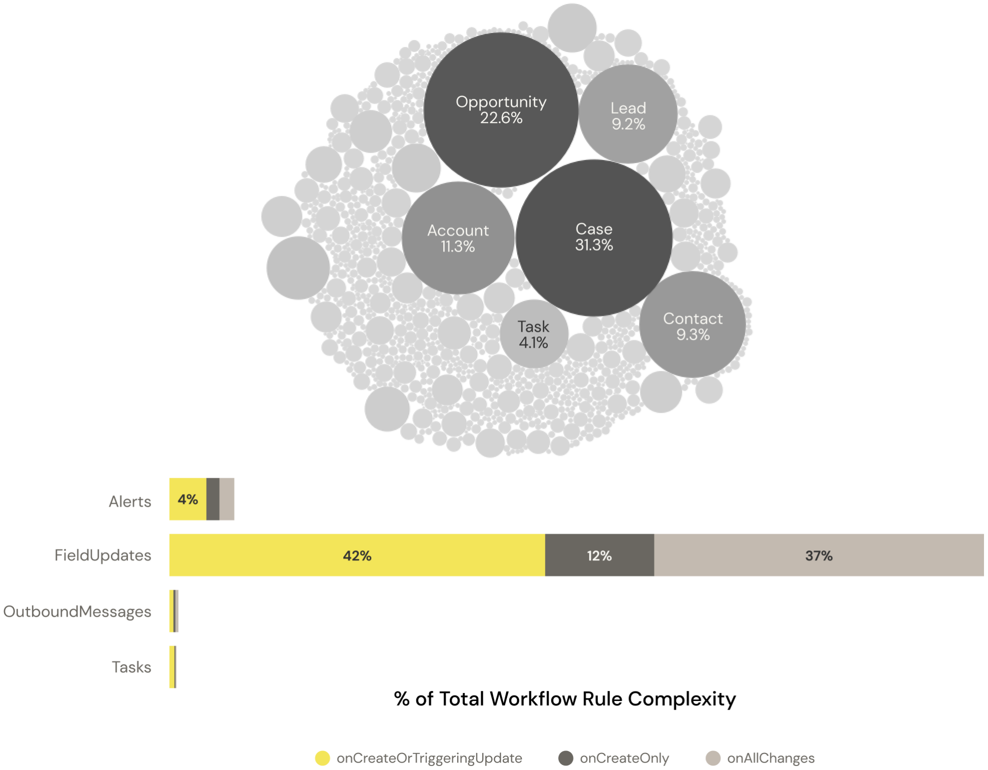


Figure 6
Total active workflow rules on standard objects with a breakdown of workflow action type.

What business units will be most impacted by this workflow migration? We can address this question by analyzing which standard objects are most built out with workflow.

Service teams are heavily represented, with workflow built on the Case object leading the way. However, the next six objects (Opportunity, Contact, Account, Contact, Lead, Task and Opportunity Line Item) suggest that overall, the sales organization will be most impacted. This is not unexpected as Sales Cloud led the way for Salesforce and administrators have been building workflow rules there for many years.

How complex are these workflows? By analyzing the workflow actions called by the workflow rules, we can see a breakdown of the different workflow action types (Figure 6). Field updates result in the most complications due to the fact that the resulting update can cause triggers and flows to be fired, sometimes resulting in unintended consequences.

Our data shows that field updates are by far the most common workflow action, suggesting that a large amount of complexity must be unraveled when migrating to flows and triggers.

What about workflow on custom objects? 48% of active workflow rules are built on custom objects. This represents a significant impact to the customizations of an org, which means admins and consultants can expect to spend half of their effort migrating workflow on custom objects. Some organizations are even dominated by custom object workflow (Figure 4).

So far we have focused on active workflow in the orgs. Our analysis found that 25% of the total workflow in an org is actually inactive, likely representing draft workflows or technical debt that should be evaluated for removal. Surprisingly, 96% of orgs had inactive workflow present. Effort to manage this inactive code increases the total anticipated workflow migration effort to >20 million hours.

AUTHOR'S ASSESSMENT

The fact that 96% of orgs had inactive workflow present is **very interesting.** It either means admins are developing by working in a draft state, or there is a huge amount of technical debt just lying around. Either way, this presents an important issue to address.

Understanding the migration effort for process builder across the ecosystem.

Processes must also be accounted for in the migration effort to flows/ triggers. Processes mimicked workflow in that 52% of processes built on

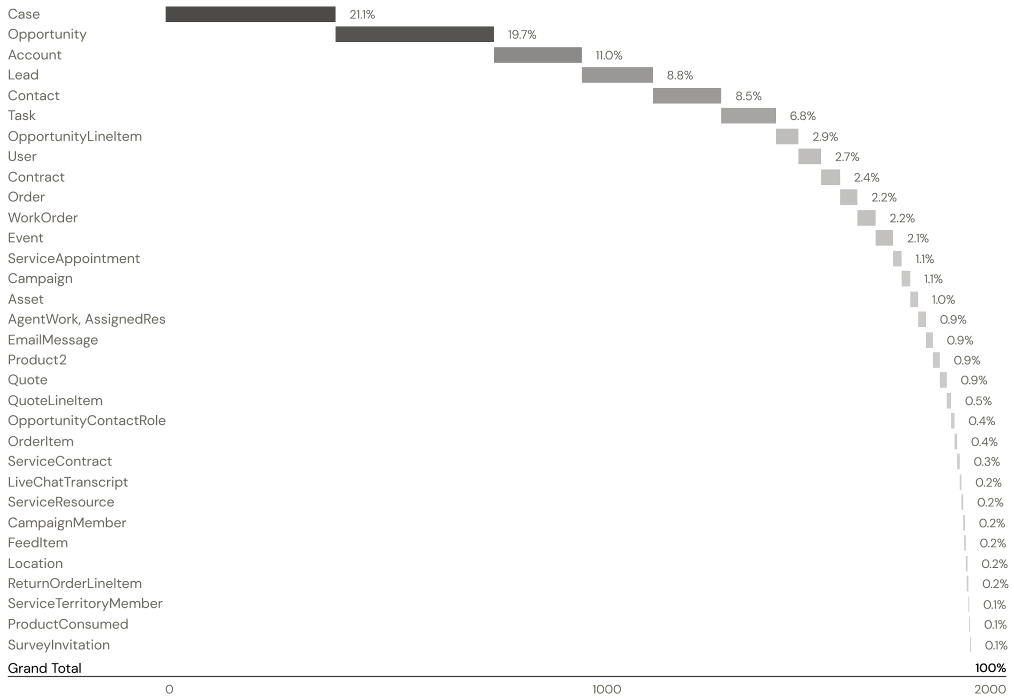


Figure 7 Percent of total active processes on standard objects.

objects are on standard objects and mainly on those same standard objects represented by workflow. However, Figure 4 shows that some orgs have nearly 100% of their processes built on custom objects. For standard objects, Case and Opportunity represent the highest weighting (Figure 7). We found that 85% of all processes were associated with object automation, while the other 15% are triggered by platform events or invoked by another process.

On average, each org has 36 active processes to be migrated. Following the logic presented for workflow, this represents ~5 million hours of migration effort across the ecosystem.

While there is no formal end-of-life announcement timeline for existing workflow and process builder from Salesforce yet, it will become increasingly expensive to maintain both while flow becomes the primary declarative automation tool.

What is the impact of managed code?

A critical divide between Salesforce code in any org is managed versus unmanaged code:

> **Managed code:** Code that was installed in your org via a package so the metadata has a namespace prefix.
>
> **Unmanaged code:** Code that your development team has developed internally or with the help of Salesforce consultants.

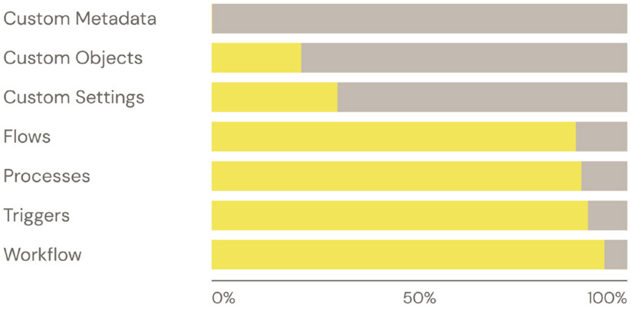


Figure 8
Managed (greige) vs. unmanaged (yellow) code in Salesforce orgs.

The importance of this divide is that details of the managed code cannot be accessed and modified via the metadata API. Thus, when working to resolve technical debt in a Salesforce org, managed packaged code is usually off limits. One scenario where managed code can be removed is for installed packages that are no longer (or never) used. This means that the majority of the effort lies in resolving challenges with unmanaged code.

So, how does our sample data set reflect managed vs. unmanaged code? Essentially all the custom metadata types are managed (Figure 8).

This makes sense as custom metadata is a very common tool for developers to configure their managed packages. Custom Objects and Custom Settings also have a high percentage of managed code at 78% and 70%, respectively. In stark contrast, workflow, processes, flows, and triggers are almost completely without namespace prefix with >88% unmanaged in all cases. This means the majority is built by customers in their own orgs, thus you can expect the majority of code to be fair game when analyzing an org for technical debt remediation.

Too Many Triggers: Are Your Objects Following Best Practice?

Triggers are powerful tools to automate business processes when used properly. A trigger is Apex code which executes before or after an event in Salesforce and is used to build custom automation on record insertions, updates, or deletions. We saw in our aggregate view of the ecosystem, there are many triggers on standard objects (Figure 2), but how many are there on each standard object in an average org? We used our Hubbl Diagnostics data set to find out.

Best practice automation uses a single trigger per object for the same event and deviation from best practice may denote technical debt. If you develop multiple triggers on a single object, the Apex Developer Guide states that you have no way of guaranteeing the order of execution if those triggers run at the same time (e.g., before insert). This technical debt can cause issues within a Salesforce org that have implications across a business.

Why use a single trigger?

Lower maintenance costs

Easier to troubleshoot and maintain.

Use less resources

Controlling trigger recursion can reduce unnecessary API calls.

Avoid unintended consequences

Consolidate triggers to guarantee the order of business processes.

Improve data quality

Eliminate automated data entry errors.

Accelerate development

A single trigger with a helper class is much more straightforward to modify. This allows for the ability to enable/disable logic from the master trigger and reuse classes outside of triggers, e.g. batch Apex or test classes.

Change business processes more easily

Easier to validate existing business processes and modify them in clean code. Simplifies process mining.

So, does the ecosystem adhere to best practice? On average, no.

To investigate this question we plotted total active triggers for an individual org against the total objects with active triggers in the same org (Figure 9). Each data point represents an individual org in the ecosystem. When plotted this way, if all orgs were following best practice, they would land on the yellow line, indicating a 1:1 ratio of triggers per object.

We found that most orgs deviate from the best practice of a single active trigger per object. For the entire ecosystem we measured a slope of 1.3, indicating that each object has on average of 1.3 triggers. Note that having a large number of triggers in the org is not indicative of deviating from best practice. Orgs with a large number of triggers can be well-managed and follow best practice, just like orgs with a small number of triggers can deviate sharply from the 1:1 ratio. In some scenarios this is acceptable because the triggers are not firing in the same context.

AUTHOR'S ASSESSMENT

We expect that huge efficiencies can be gained by businesses in the ecosystem that are deviating from best practice. A dedicated technical debt remediation effort needs to be applied to unlock that potential. We recommend all admins and consultants review the trigger counts on their objects. If more than one trigger per object is found, investigate the context that they are firing and determine whether they are managed/unmanaged. The results will aid planning and prioritization of remediation efforts.

Find Your Risky Objects

So far, we've reviewed aggregate views of workflow, process, and trigger automation on objects in Salesforce orgs. While we could explore additional automation related to objects such as flows, approvals, and sharing rules, we'll shift our attention summarizing all object automation in a single view allowing us to identify "risky objects".

> **What is a risky object?** Risky objects are objects that contain high record counts combined with high levels of automation. Basically, a highly automated object that gets used a lot. This combination can be a supernova ready to ignite—creating a mess of data within an org.

Essentially all risky objects break Salesforce's best practices for designing processes: for each object, use one automation tool. The addition of new automation or modification of existing automation to a risky object can cause unintended consequences that impact business processes and data quality.

For example, if the data in Figure 2 were for a single org instead of the aggregate view, Case and Opportunity would be the highest risk objects, assuming similar record counts for all objects. We all know that not all orgs are alike. No two Sales Cloud orgs are the same. They will have different risky objects depending on the customization the business has

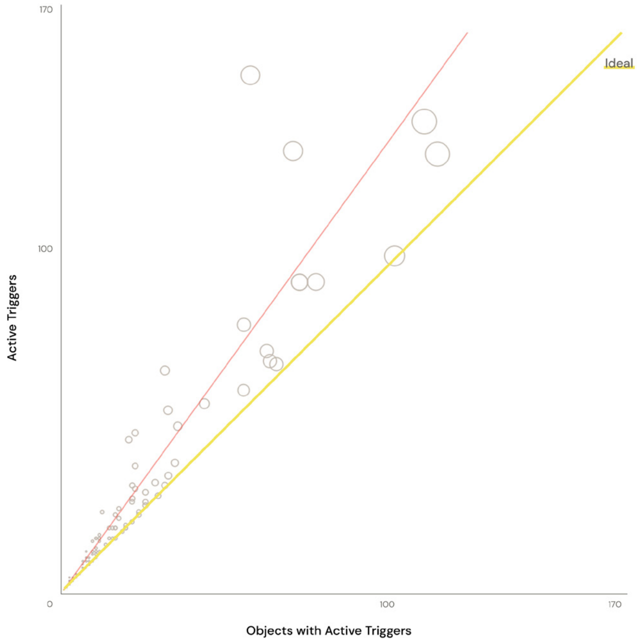
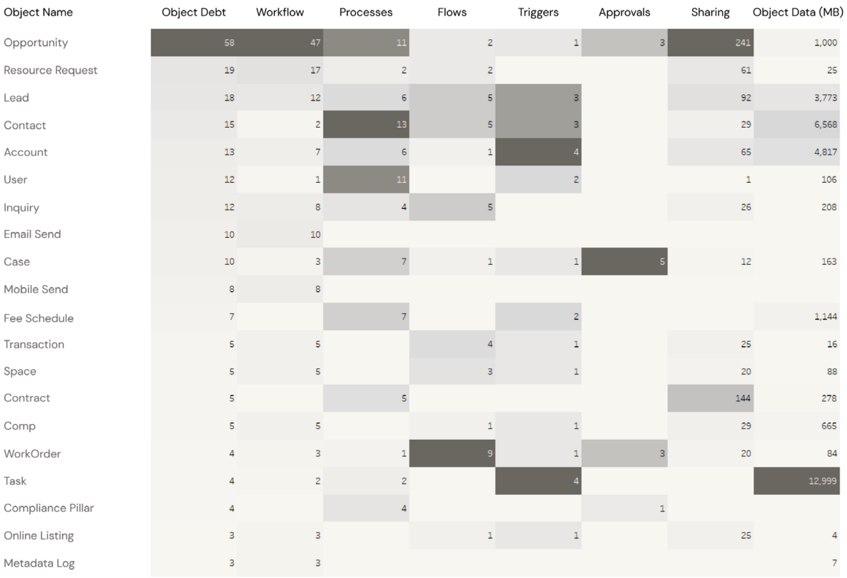


Figure 9

Active triggers per object. A linear trendline fit (red) of the data has a slope of 1.3, indicating 1.3 triggers per object and R^2 of 0.88. The yellow line has a slope of 1, representing one active trigger per object.

built. For example, a subset of the object automation table for a single org in Figure 10 shows that Opportunity is very complex, but several other objects are likely more complex than Case. Note that there are >15 objects that violate Salesforce's one automation tool per object rule in Figure 10. The high number of automation types increases the risk for unintended consequences due to order of operations.



Object Name	Object Debt	Workflow	Processes	Flows	Triggers	Approvals	Sharing	Object Data (MB)
Opportunity	58	47	11	2	1	3	241	1,000
Resource Request	19	17	2	2			61	25
Lead	18	12	6	5	3		92	3,773
Contact	15	2	13	5	3		29	6,568
Account	13	7	6	1	4		65	4,817
User	12	1	11		2		1	106
Inquiry	12	8	4	5			26	208
Email Send	10	10						
Case	10	3	7	1	1	5	12	163
Mobile Send	8	8						
Fee Schedule	7		7		2			1,144
Transaction	5	5		4	1		25	16
Space	5	5		3	1		20	88
Contract	5		5				144	278
Comp	5	5		1	1		29	665
WorkOrder	4	3	1	9	1	3	20	84
Task	4	2	2		4			12,999
Compliance Pillar	4		4			1		
Online Listing	3	3		1	1		25	4
Metadata Log	3	3						7

Figure 10
The object automation table view for a single org from a Hubbl Diagnostics scan.

How are risky objects typically identified in a Salesforce org?

Hint: They often aren't.

Identifying risky objects in Salesforce must be done manually, which requires a number of steps in the setup menu and must be done on an object-by-object basis. To identify them, consultants have to work through the manual process of the following five steps:

1. Select object.
2. Review data storage tab, identify number of records (total data/2kb per record).
3. Review each automation tab to identify whether the object has this automation type associated with it.
   a. Workflow
   b. Triggers
   c. Process builder
   d. Flow
   e. Approval processes
   f. Sharing rules
4. Tallying up the combined automation, taking into account active/inactive code.
5. Once they've done this, this process must be repeated for all the objects in the org.

This process can be time consuming and tedious without an org intelligence tool; so much in fact that most admins don't even attempt to do it. Meanwhile consultants are expected to make a level of effort estimate for the amount of work required on specific objects without understanding the complexity of the objects or the org itself, since they are often not granted access during the estimation process.

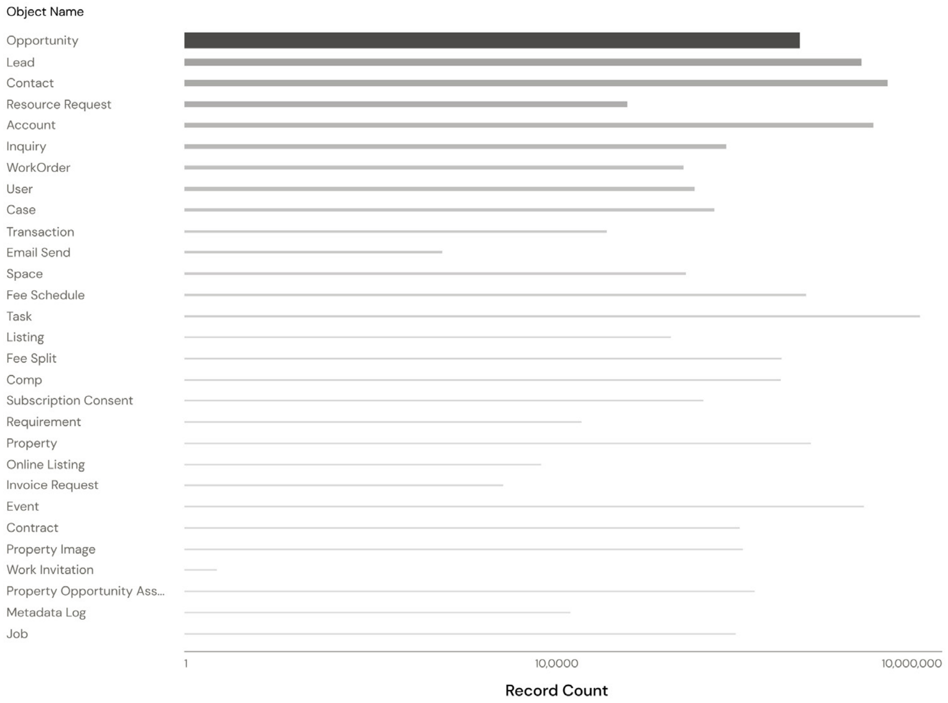


Figure 11
A view of high risk objects in a single Salesforce org. The darker the color and wider the line, the more complex the automation on the object.

As a result, the solution often requires an "org discovery" project that can take 2-8 weeks, extending the customer's engagement and likely increasing costs. The consequences of this endeavor include off-base estimates, resulting in change orders slipping timelines, and worst of all, a bad customer experience.

There's a better way.

To alleviate this effort, we developed a Risky Objects visualization within Hubbl Diagnostics that automates the process for a single org and provides instant insight into how the org is utilized (Figure 11). It can be viewed with sorting by object complexity or record counts by object.

Compare and Rank Salesforce Orgs Relative to Each Other

One of the most common asks among Salesforce customers is, "Where does my org stand in comparison to the ecosystem?". The impetus for this question comes often when seeing the continued growth and evolution of the Salesforce platform. New offerings and verticalized solutions are creating opportunities for platform usage by industry or segment, increasing this desire to know how other comparable organizations are using Salesforce.

We can finally answer this question using the Hubbl Diagnostics benchmarking data set.

To do so, we've created a proprietary method for quantifying the complexity of Salesforce orgs using a single number. We call that number the Hubbl Diagnostics Index. The Index currently takes into account the primary source's object automation and can be used to classify an org instantly. Expressed as a percentage, it represents the percentile of complexity that lies in the org relative to all other orgs that have been scanned by Hubbl Diagnostics. Figure 12 shows an example of the Hubbl Diagnostics Index scores for a subset of ~100 orgs. For example, if an org lands in the first quartile, <11% complexity, there's a good chance the org has little technical debt and project work should be straightforward. If the org lies beyond the 19% median and in the 4th quartile >31% there is likely a need to assess the level of technical debt and discuss remediation efforts prior to undertaking new project work. Clearly orgs with complexity scores beyond 50% are among the most complex and should be approached with caution.

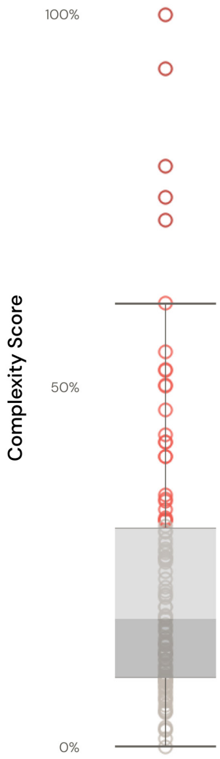


Figure 12
Hubbl Diagnostics Index scores for about 100 Salesforce orgs.

Positive impacts of benchmarking:

> **Solve the org discovery challenge:** Benchmarking is extremely useful for those facing the "new org challenge", such as Salesforce consulting partners, new Salesforce admins, or organizations looking to acquire and merge businesses and their Salesforce orgs.
>
> **Improve consulting experience:** At a quick glance, you can see a ranking of the org within the ecosystem. This can be leveraged by Salesforce consulting partners to de-risk estimates and include in their pricing and resource strategies, improving the overall experience in the ecosystem.
>
> **Understand complexity at a glance:** Benchmarking is also useful for Salesforce administrators attempting to garner an understanding of the complexity of orgs they have worked on or inherited. For example, administrators that have managed more complex orgs, anecdotally, can now prove with quantitative metrics, the complexity of orgs they have worked on. This supports organizations finding the right type of administrator needed to manage their org.

Org Size ≠ Size of Effort to Analyze Technical Debt.

By counting the number of metadata items in a Salesforce org we gain an understanding of org size. However, org size does not directly relate to the amount of technical debt that must be addressed or indicate the level of effort to analyze it prior to a remediation project.

To demonstrate this with quantitative data, we calculated an estimated level of effort to analyze the automation technical debt in each benchmark org that we scanned. We assume that all workflow and processes must be analyzed, due to Salesforce's impending end-of-life, and migrated to flow. We also identify objects that have more than one trigger or flow and sum the number with >1 on each object. We applied a level of effort (1 hour) to each of the workflow, processes, flow, and triggers tallied. This represents a level of effort to just analyze each of these components for each org — not the actual effort to remediate. We ignored all other code in this analysis.

A key observation is that the number of metadata items scanned in an org does not necessarily correlate with the amount of effort expected to analyze the technical debt inside it. Small orgs can accrue more liabilities than larger, better managed orgs. This suggests it's difficult to know how much complexity lies in the org without scanning it using an org intelligence tool or manually tallying it. Further analysis accounting for all technical debt in the org will provide deeper insights.

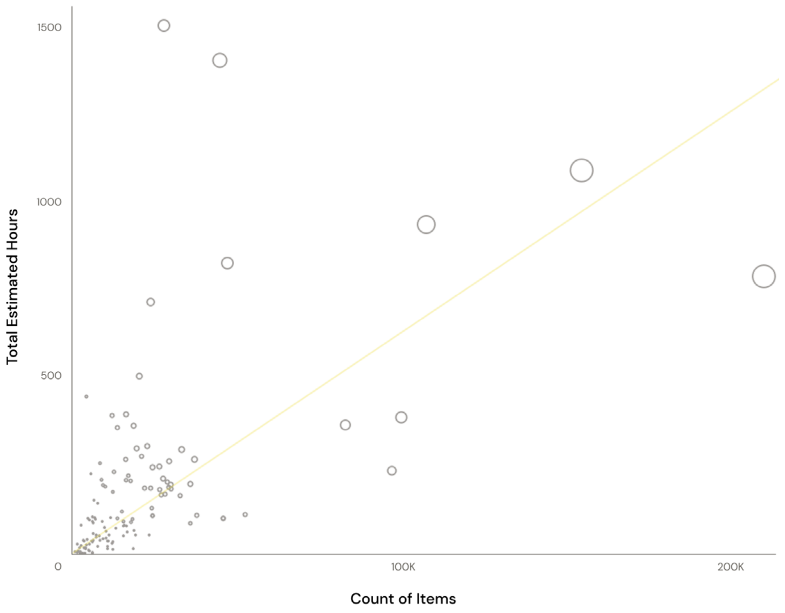


Figure 13
Automation technical debt analysis effort by org.

Building your case for org intelligence.

Bringing it back to the challenges we outlined at the beginning of the report (e.g., the maturing of the Salesforce ecosystem, best practices, innovation, and new user and customer demands) those individuals tasked with Salesforce investment and administration must consider how to quickly and effectively assess org complexity.

Salesforce consulting partners or administrators that rely on making best guesses with manual cursory (or no) review of a Salesforce org will ultimately end up with unhappy customers or stakeholders due to unexpected changes in effort and budgets to deploy enhancements.

Salesforce org admins and architect teams should demand that consultants utilize org intelligence tools to reduce the time to estimate, improve the accuracy of the estimates received, and increase the trust that forms the basis of the partnership. If they don't, they are paying extra for the hours necessary to analyze the org. Those that take over new Salesforce orgs or are attempting to plan their strategy for addressing technical debt must scan their org to understand and monitor where they stand relative to the ecosystem to assure their executive leadership that they can maintain competitiveness with agile innovation.

Key Actions

01. Scan

Understand 'what your Salesforce org is'.

As a Salesforce Consultant or Admin, you may come across a new org and wonder, what's going on in there? What was left behind? Where are the riskiest spots that have developed in the org?

Utilize an org intelligence product to automate the process. It will allow you to focus your precious time on other efforts.

Building this understanding, whilst time consuming, is fundamental to enabling organizational growth.

02. Analyze

What does the data mean?

Perform an analysis of the metadata you have tabulated. Again, an org intelligence product can automatically generate outputs to reduce your manual level of effort and get key data to executives quickly.

03. Prioritize

What to do, when.

Build a tech debt roadmap. Chip away. Watch your legacy declarative automation migrate over to flow quarter by quarter.

AUTHOR'S ASSESSMENT

These observations are also critical for those considering mergers and acquisitions of companies utilizing Salesforce. Although you may be acquiring a small Salesforce org, it may be carrying a large amount of asset impairment and contingent liability due to technical debt.

Based on our data, we're seeing on average

150 hours per org

to perform an object automation audit

This, multiplied by the number of orgs in the entire ecosystem equates to over

$3B

in technical debt remediation professional services

Methods

What's behind the data?

Metadata was collected from hundreds of Salesforce orgs using the org intelligence product Hubbl Diagnostics. Hubbl Diagnostics retrieves metadata like objects, fields, record types, groups, roles, profiles, permission sets, Apex and custom code, as well as record counts for all objects in the org.

The product requires no packages to be installed into the target Salesforce org and authentication is handled via OAuth 2.0. The metadata collected is analyzed, an output is stored as a scan, and the metadata is deleted. The original metadata is not stored by the platform. No records are ever extracted by Hubbl Diagnostics, avoiding data security issues. Thus, no analysis of record data, beyond record counts, is included in the output.

For this report we utilized a subset of ~100 orgs to complete our analysis. The primary criteria for inclusion in analysis were unique production or full sandbox Salesforce orgs and a breadth of complexity scores.

We estimate that the manual effort that would have been required to generate the metadata analysis of all orgs scanned in this report would have been 4000-8000 hours. The manual effort to collect a Hubbl Diagnostics scan takes less than a minute to initiate. So we realized over 2000 times efficiency by using Hubbl Diagnostics to analyze the orgs' metadata. Each scan can take minutes for the most basic org to several hours for very complex orgs. Once the scan is complete the user is notified to review the results.

TO SCAN YOUR ORG VIA HUBBL DIAGNOSTICS VISIT WWW.HUBBL.COM

Contributors

PRIMARY CONTRIBUTOR

Michael J. Bogan, *PhD, Director of Product Strategy, Uncommon Purpose — LinkedIn*

Michael Bogan is an entrepreneurial scientist and inventor who pursues innovation acceleration via the application of cloud technology. In addition to his contributions to the scientific community as a Staff Scientist at Stanford University/SLAC National Laboratory, Lawrence Livermore National Laboratory and as the Director of Development at IonDx, Mike spent the last nine years as a Salesforce.com Director of Research and Development at Traction on Demand. While at Traction on Demand, Mike led the architecture for hundreds of successful Salesforce implementations and contributed critical expertise to Salesforce platform solutions, such as Net Zero Cloud (2019 Partner Innovation Award).

Mike is now the Director of Product Strategy at Uncommon Purpose, a product development organization with a mission to reinvigorate the Salesforce ecosystem with solutions that bring greater clarity and efficiency to business. He is currently focused on Hubbl Diagnostics, and has aggregated the Salesforce metadata to provide critical analysis and insights to help improve usage of the CRM platform for all ecosystem community members.

ABOUT HUBBL DIAGNOSTICS

Hubbl Diagnostics is an org intelligence product developed by Uncommon Purpose for the Salesforce ecosystem. Hubbl Diagnostics is on a mission to help people get ahead of our ever-expanding org complexity by providing the broadest, most actionable insights.

ABOUT UNCOMMON PURPOSE

Uncommon Purpose is more than a tech company, more than an incubator and more than a venture capital firm. Uncommon Purpose generates ideas that can be turned into dynamic business solutions. We do this by focusing on the potential of our people.

ADDITIONAL CONTRIBUTORS

Christopher Bruzzi, *EVP, Product Development and Managing Director of Hubbl Diagnostics, Uncommon Purpose*

Contributions include: As the Managing Director of Hubbl Diagnostics, Chris' contributions run through the entire benchmark report. From analytical insights, content creation, Salesforce ecosystem thought leadership, editing and more, Chris' guidance made everything possible.

Kevin Murray, *Vice-President of Marketing, Uncommon Purpose*

Contributions include: Salesforce ecosystem thought leadership and research pertaining to the state of the current Salesforce ecosystem including critical trends in technical debt. Report strategy and planning as well as editing.

Garrett Shay, *Data Analyst, Uncommon Purpose*

Contributions include: As the data analyst for Hubbl Diagnostics, Garrett's contributions can be seen through the analytical insights collection as well as editing and graphical visualization of critical elements and output.

Carlie Welsh, *Manager of Marketing Communications, Uncommon Purpose*

Contributions include: Salesforce ecosystem thought leadership and research, report strategy, structural development and main editor. Where Mike provided the research, analysis and insights, Carlie provided the structure and editing prowess to create cohesiveness and compelling storytelling.

Alexis Shuffler, *Graphic Designer, Uncommon Purpose*

Contributions include: Working in collaboration with the product and marketing teams, Alexis designed and typeset the report, and helped bring critical graphical elements to life by applying design and brand principles.