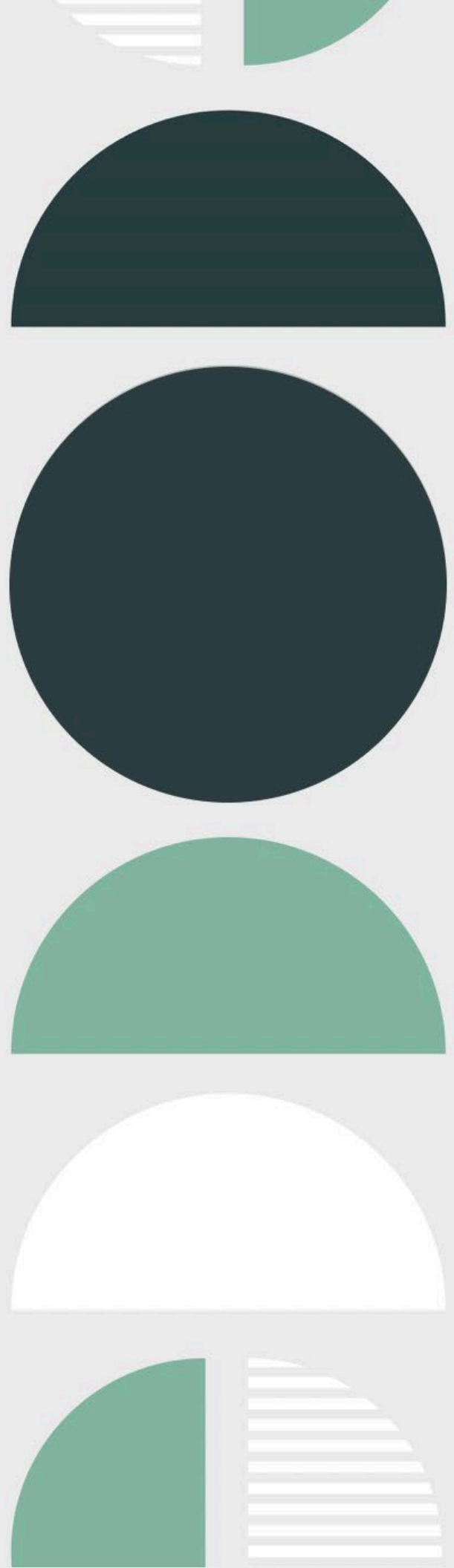


Business-smart customization

Developed by the ServiceNow Best Practice Center of Excellence

September 2020



What is business-smart customization?

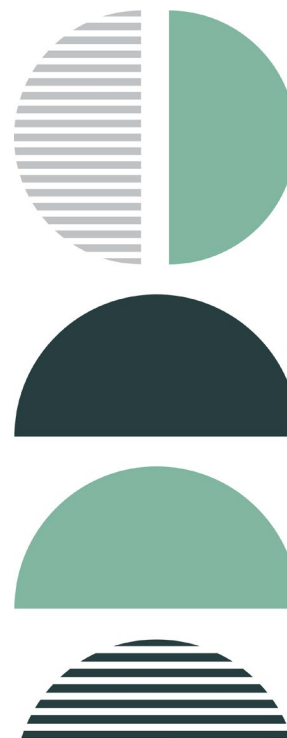
With business-smart customization, your decisions about customization always start with a *value* perspective: Does the *business value* promised by customization outweigh the *risk* it presents, in terms of your technical debt and upgradeability? Customers who adopt this approach are able to take better advantage of the Now Platform®'s powerful capabilities while maintaining a healthy and well-managed instance.

Terms and definitions

Customization – This Success Workbook uses *customization* to refer to business demands for custom functionality. When referring to how these demands are implemented, this workbook refers to both *configuration* and *custom development* (both on ServiceNow applications and in the development of new applications).

Configuration – Configuration includes capabilities and/or features on the Now Platform that allow you to deliver against custom business demands in a safe and supported way, such as through modifications to forms or table extensions. While you should avoid overconfiguration—for example, by adding excessive forms or UI policies—you should still use configuration as your first or preferred option to meet business demands.

Custom development – This includes custom development on both ServiceNow applications and new applications you develop by changing baseline business rules or developing new applications from table extensions. Custom development is recommended when configuration can't meet new business demands; it should employ consistent, recommended tools and methods (as outlined in this document); and it should be overseen by clear, effective governance.



Why should I invest in business-smart customization?

First, why customize at all? First, the Now Platform gives you the ability to create digital workflows quickly based on emergent business needs. Second, customization can improve the user experience, so that business services and workflows align with what you need to drive adoption. Finally, with customization, you can scale against your specific automation needs—and your organization can develop and scale workflows, often workflows that are specific to your company or industry and that aren't currently offered out of the box.

But in each of these cases, customization should follow business-smart approaches. When you invest in these approaches, you can:

- **Meet business demand while minimizing technical risk and complexity –** Customization can be a powerful tool, but it needs to be business-smart—only used when it can deliver clear value, so you can limit your technical debt.
- **Upgrade faster –** Excessive and complex customization can slow your upgrades, since each customization is skipped during the upgrade process and requires manual review. With a business-smart customization strategy, you prevent unnecessary customization, reducing the number of skipped records requiring review.

What will this workbook help me do?

When you complete the steps in this workbook, you'll have:

- Clearly defined governance measures that limit customization to clear business needs
- An application development strategy that allows you to scale development in support of new customization that also keeps your technical debt and complexity to a minimum
- A portfolio management strategy that keeps customization limited to current business need and that eliminates any unnecessary customization

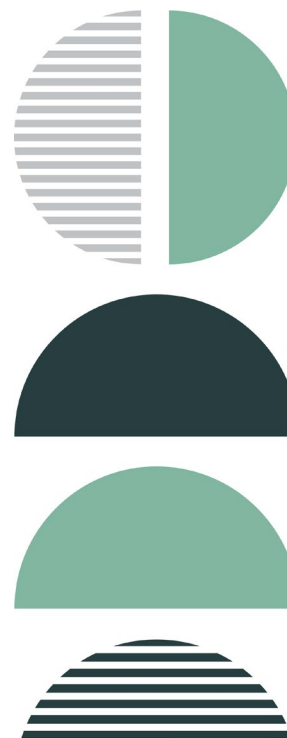
How to use this Success Workbook

This Success Workbook will guide you through the steps to evaluate and govern your approach to customization on the Now Platform.

Start by reading through the [Workbook Checklist](#). Review the steps and tasks that need to be completed before you start to promote business-smart customization.

From the checklist page, you can either proceed through the workbook page by page or navigate only to the sections you need.

Each section includes “check your progress” questions that you can answer to test whether you need to complete the tasks listed on that page or should move forward. Hyperlinks are included throughout the workbook so you can navigate back and forth between the checklist page and different sections.



Workbook Checklist: Business-smart customization

Step 1: Guide teams to use ServiceNow out-of-the-box capabilities

- Educate teams on how ServiceNow works out of the box (OOTB).
- Highlight success stories to encourage use of OOTB functionality.
- Identify where teams are not positioned to take advantage of OOTB functionality.
- Build an understanding of the drivers of demands for customization.
- Educate stakeholders on the capabilities coming in ServiceNow upgrades.

Step 2: Define demand management policies for configuration and customization

- Provide a clear, central intake mechanism for new customization and configuration requests.
- Define approval authorities for new customization and configuration requests.
- Define guidelines for how configuration and customization requests will be evaluated and approved, using simple scoring techniques.

Step 3: Provide configuration and customization guidance to development teams

- Give local development teams guidance on using scoped applications to respond to customization demands.
- Define a standard process for developing and releasing custom applications.

Step 4: Audit customizations and configurations for retention or removal before upgrades

- Develop a process to audit customizations and configurations regularly for retention or removal.
- Review your backlog of customization and configuration demands to determine if any demands can be removed from the backlog.
- Review your existing customizations to determine if any are redundant to new capabilities and determine if you should remove them.



Step 1: Guide teams to use ServiceNow OOTB capabilities balanced with an understanding of their demands for customization

Many demands for configuration and customization can stem from your business users' lack of awareness of the OOTB capabilities already available to them on the Now Platform. Organizations typically drift into customization and technical debt over time due to lack of knowledge and communication about the capabilities available in ServiceNow that can meet their business needs. Any approach to customization depends first on a strong, ongoing education campaign on ServiceNow capabilities and functionality.

Check your progress:

- Is your user base aware of the scope of the ServiceNow OOTB capabilities and functionality available to them?
- Have you identified and resolved obstacles to users' ability to use ServiceNow OOTB capabilities?

If you answered "yes" to both questions above, proceed to the [next step](#). If not, complete these action steps to complete this step:

- 1. Educate teams on how ServiceNow works OOTB.** For each release, conduct workshops for process users (e.g., service desk agents) and other relevant stakeholders to build awareness of ServiceNow capabilities and functionality. You can start with a pre-workshop survey to identify teams' current levels of awareness and target your efforts at common knowledge gaps across teams. You can also use this survey to solicit questions about ServiceNow capabilities.
- 2. Highlight success stories to encourage use of OOTB functionality.** Identify new or under-used functionality that you can pilot with selected teams. As you run these pilots, document specific process improvements and communicate them broadly across the organization through webinars, internal social media, or other channels.
- 3. Identify where teams aren't positioned to take advantage of OOTB functionality.** For OOTB functionality that's persistently underused, work with process user teams to identify the root causes behind their lack of use. The best way to do that is to run interviews or spend a day observing process user teams, focusing on the following questions:
 - Are there any technical obstacles to the teams' ability to take advantage of OOTB functionality?
 - Does the current team structure (or process model) limit the team's ability to take advantage of OOTB functionality?
 - Does lack of skill or training limit our team members' ability to take advantage of OOTB functionality?

Develop proposed remediation measures based on your findings and work with your process user teams and/or governance functions to implement them.

- 4. Build an understanding of the drivers of demands for customization.** To understand what's behind customization demands, focus on understanding your process users' service experience (like your IT service desk staff's experience). Configuration or custom applications could



significantly change how they perform their jobs positively or negatively. Look for:

- How they use any current configurations or custom applications. Process users have often developed or adapted behaviors (or resorted to workarounds), for reasons that may not be clear unless you ask. This can prompt a discussion around when customization is (and isn't) required for them to be effective, and which moments of truth in a workflow have the greatest impact on their performance and experience.
- How to evaluate the value and experience delivered by customization as you set up the demand management processes recommended in Step 2.

To develop this insight, ask a business analyst or business relationship manager to engage process users who are highly influential among their peers and who may drive demands for customization.

Also work to understand the demands for customization on points of interaction with service consumers, particularly in the Service Catalog and Service Portal. Customization needs in these areas will likely focus on improving the navigation (for example, through collapsible menus) and process transparency (for example, in tracking requests). To measure the effectiveness of a customization at a point-of-service interaction, measure its effects on adoption. If you suspect a customization is *negatively* impacting adoption, review its original business case.

If you don't have a backlog of customization demands for service consumers, you can use several scalable techniques to identify and understand demands—surveys, focus groups, user acceptance testing, and even design contests.

- 5. Educate stakeholders on the capabilities coming in ServiceNow upgrades.** To Demands for customization and configuration may be redundant to the capabilities and innovations available in upcoming ServiceNow releases. Platform team leaders should allocate time to educate business leaders and their teams on the capabilities they can use in the next upgrade.

You should have a business-facing roadmap that includes

- (a) New products that you're implementing
- (b) Any new capabilities that you're piloting
- (c) New capabilities that will be available in your next upgrade.

Socialize this roadmap in one-on-one meetings with business leaders and key stakeholders to familiarize them with new capabilities and potential use cases. At the same time, review your backlog of any configuration and customization requests with the goal of identifying whether any new capabilities coming in your next upgrade can satisfy the demands in your backlog.



Practitioner insight: To manage demands for customization and configuration effectively, you need a deep understanding of how ServiceNow is used by front-line process users. Process users have often developed or adapted behaviors (or resorted to workarounds), for reasons that may not be clear unless you ask. Conduct interviews or focus groups with process users to understand how they're using OOTB capabilities, and where additional configuration or customization may be needed (or not) to support their performance.



"Most new customers find that they're inundated with demands for custom development right after go-live. That's why it's critical to communicate your roadmap as broadly as possible so that your internal business departments aren't making requests for capabilities you've already got programmed."

– ServiceNow Inspire strategist

Step 2: Define demand management policies for configuration and customization

Business-smart customization requires strong governance. This starts with clear policies that educate development teams and other stakeholders on the “value bar” that customization and configuration requests need to meet for approval.

Check your progress:

- Do we have clear demand management mechanisms in place for evaluating customization and configuration demands?
- Are these mechanisms supported by our demand board?
- Do we have a consistent approach to evaluating customization and configuration demands that’s based on clear business value and technical risk criteria?
- Have we educated our development teams and stakeholders on these criteria?

If you answered “yes” to both questions above, proceed to the [next step](#). If not, complete these action steps to complete this step:

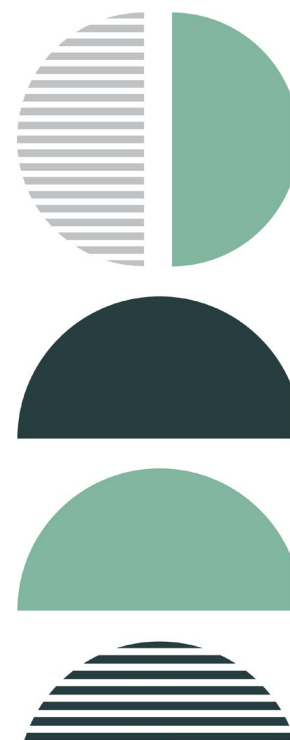
1. **Provide a clear, central intake mechanism for new customization and configuration requests.** Develop a simple, automated intake form (using ServiceNow Demand Management, if possible) that helps development teams and business lines identify the business value, cost, and risk associated with their request.
2. **Define approval authorities for new customization and configuration requests.** Consider creating a [ServiceNow demand board](#) in place if you don’t already have one. Schedule monthly meetings to manage and approve demands for customization and configuration. If this isn’t possible, assign approval responsibilities to your change advisory board (CAB).

When possible, include your platform architect, platform owner, IT process/domain owners, portfolio/service owners, demand managers, program manager, business analysts, vendor managers, key suppliers, and partners. The demand board should be your central point for approvals for customization and configuration.

3. **Define guidelines for how customization and configuration requests will be evaluated and approved.** Build two simple scorecards: one for requesters (as part of your intake form) and one for your demand board.

For your requesters – a value scorecard:

Individual demands for configuration and custom applications can be assigned a simple value score that reflects the reasons behind the demand in terms of business value. Use the table below as a starting point. Keep the scoring consistent with your organization’s existing model for requirements management, using the standard business value criteria that business analysts employ to prioritize requirements and/or user stories.



VALUE SCORE	DESCRIPTION
5 – Mandatory	Customization is required for regulatory and compliance purposes.
4 – Critical	Customization is a must to realize a business value objective and/or adoption requirements.
3 – Medium	Customization supports realizing a business value objective and/or adoption but workarounds are available.
2 – Low	Customization supports service experience for service consumers, process users, and/or developers, but it doesn't necessarily promote a business value objective or adoption.
1 – No Value	Customization does not support improved service experience, value realization, or adoption.

For your demand board, use a complexity scorecard that reflects technical risk.

Most configurations and customizations can be assigned a low, medium, or high complexity score to reflect their volatility and the potential risk they present to upgrades, technical debt, and/or performance. The table below shows a simple rating for common configuration and customization scenarios.

These complexity scores reflect a simple low, medium, or high rating. Your platform experts can assign this rating based on experience with upgrades, data on past support issues, and/or the level of testing required to validate the customization. You can also consult with ServiceNow experts for additional guidance.

Your demand board should rely on this to set thresholds—the minimum level of business value that a new configuration or [custom application](#) should deliver to warrant implementation. Business value, according to the simple scoring framework we defined earlier, should take into account improvements to user experience, adoption/usage, compliance, or the accomplishment of business objectives.

CONFIGURATION OR CUSTOM APPLICATION SCENARIO	COMPLEXITY SCORE (DEFINED BY POTENTIAL RISK TO PERFORMANCE AND/OR TIME TO UPGRADE)	WHAT MINIMUM LEVEL OF BUSINESS VALUE SHOULD CUSTOMIZATION DELIVER?
Modification to form layout or design	LOW – This will typically not delay upgrades or new functionality, although poorly planned layouts may disrupt the user experience.	Low
Add fields and/or UI policies to forms	LOW – This will typically not delay upgrades or new functionality, but overconfiguration of forms (e.g., adding 50+ fields) risks both the service experience and usage.	Low-Medium
Build a simple custom integration	LOW – Integrations that can use IntegrationHub (see below) will not typically delay upgrades or new functionality.	Low
Extend an existing table (e.g., incident) in scope with new fields only	LOW – This will typically not delay upgrades or new functionality, but additional fields should be strictly tied to validated business requirements.	Low-Medium
Extend an existing table (e.g., incident) in scope with some scripting	LOW-MEDIUM – This will typically not delay upgrades or new functionality, but additional fields should be strictly tied to validated business requirements. Complexity is dependent on the extent of the scripting.	Medium
Extend an existing table (e.g., incident) as the basis for a different application (e.g., HR)	LOW-MEDIUM – This will typically not delay upgrades or new functionality, but additional fields should be strictly tied to validated business requirements. For new applications, this should be preceded by clear process mapping and acceptance testing among knowledgeable users (e.g., HR process owners).	Medium
Use the incident table as is for a different use (e.g., HR)	LOW – This will typically not delay upgrades or new functionality but use of the incident table outside its intended use case (e.g., ITSM) should be preceded by clear process mapping and acceptance testing among knowledgeable users (e.g., HR process owners).	Medium
Change the state choice list (e.g., modify the incident process)	LOW-MEDIUM – This may affect time to upgrade or new functionality if the modification changes underlying business logic, rules, and/or policies.	Medium
Build a new scoped application	MEDIUM – This will typically not delay upgrades or new functionality, but scoped applications should be strictly tied to validated business requirements. The complexity is dependent on the extent of the scripting.	Medium
Build a new global application	MEDIUM-HIGH – This may affect time to upgrade due to testing or new functionality if the application changes underlie business logic, rules, and/or policies.	Critical
Change baseline business rules (e.g., modify the SLA process)	HIGH – This may affect time to upgrade due to testing or new functionality.	Critical/Mandatory
Build complex custom integration	HIGH (DEPENDENT ON COMPLEXITY OF INTEGRATION) – This may affect time to upgrade due to testing or new functionality.	Critical/Mandatory

- 4. Push your demand guidelines to the front lines.** Demands for customization can present a dilemma: you need to establish clear guidelines for business-smart customization—but highly centralized, command-and-control approaches to demand management can create frustration as business lines await their turn in the backlog. In some cases, it can even lead teams to pursue custom solutions outside proper governance channels.

In complex organizations with multiple lines of business, you should federate your model for managing customization demands. In this approach, your central demand board:

- Defines your business-smart evaluation criteria to approve customization demands
- Maintains oversight, review, and approval authority over any global or shared objects, like a global custom application

Your demand board can subsequently delegate approval authorities for customization demands that are low risk or have limited potential for negative impact, have precedents that have been approved by the demand board nearly 100% of the time, and/or will be limited to a scoped environment (see Step 3 for more on this).

It's important to delegate some authorities to the front lines of demand—for example, your business line development teams or product owners—so you can be responsive to business need. To ensure that you avoid risk, this delegation should be accompanied by clear communication of decision-making guardrails. Your demand board should make sure that delegated authorities use the same business-smart evaluation criteria to guide decision-making as the demand board does for global demands. To build this consistency, you need clear, regular communications—and record-keeping—between your demand board and front-line approval authorities.



What about existing customizations? The steps listed above help you make a clear cost-benefit analysis for new configurations and custom applications. But what about those that have already been implemented?

As part of the upgrade process, ServiceNow provides customers the ability to run an automated inventory of changes made to OOTB scripting and configuration. To prevent customizations from being overwritten during system upgrades, the upgrade process skips (does not apply the update to) objects that have been customized. Upgrade Monitor can be used to list all updates skipped during the upgrade process to assist in resolving customizations that need review.

Skipped records (including changes to hard-coded records `sys_ui_form_section`, `sys_ui_related_list`, and `sys_choice_set`; and changes to tables that have at least one field of type HTML, XML, script, or script_plain) are flagged with a priority score reflecting their potential need for remediation during an upgrade. Any of the customization scenarios outlined in the table above will likely be associated with multiple skipped records. As such, customers with existing customizations can:

- Generate a list of skipped records and their priority scores
- Conduct an analysis with a group and associate relevant skipped records with a skipped customization (for example, extending an existing table in scope). You may be able to look to update sets and your scoped applications for documentation on why a specific record was customized.
- Evaluate the customization in terms of its total cost and benefit and determine if any remediation is necessary at the level of individual skipped records.

Refer to our guidance on the process to [review and address skipped changes](#) for additional details.



Changing OOTB business rules – a note of caution. Business rules can be changed to perform several actions, including preventing users from accessing or modifying certain fields on a form, displaying information messages to the user, or changing field values on a form that the user is updating. Changes to business rules in ServiceNow can be complex and should adhere to these best practices:

- **Prevent recursive business rules** – Avoid using `current.update()` in a business rule script. The `update()` method triggers business rules to run on the same table for insert and update operations, leading to a business rule calling itself over and over, causing system performance issues. You can prevent recursive business rules by using the `setWorkflow()` method with the `false` parameter.
- **Where possible, stay small and specific** – Ideally, changes to business rules should remain limited. Big business rule changes should come with strong business justification, given their potential complexity.

Additionally, in the past, the best practice strategy for updating baseline scripts was to make a copy of the record you want to update, make the original record inactive, and change the script in the copy. This complicated the upgrade process since there were two records to maintain for each change. Instead, the new best practice from ServiceNow is to modify the baseline record itself. The record will be available for review and revert as a skipped file during upgrade.

Step 3: Provide configuration and customization guidance to development teams

Governance by itself is not enough: Customization and configuration should be implemented using the right tools and mechanisms that minimize technical risk, particularly to your global applications. Your organization should provide clear, straightforward guidance to development teams on how to implement customization and configuration safely. For custom application development, this should focus on using scoped applications by default. Your design board, supported by your technical governance function, should be the vehicle for defining and communicating this guidance.

Whether you limit development to pro-code only or include citizen developers, your approach to customization and configuration should be embedded in a larger strategy to [manage application development](#) on the Now Platform.

Check your progress:

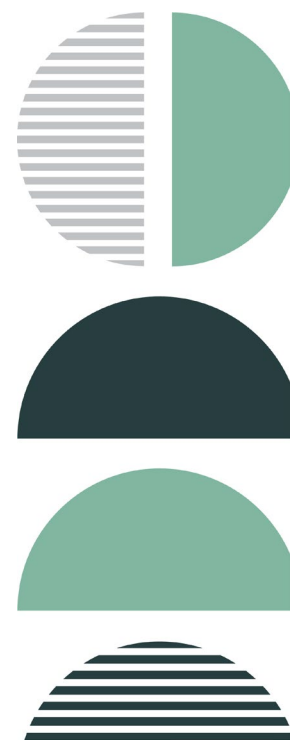
- Have we guided teams to use scoped applications as their default option for custom development?
- Have we established a design board to identify and promote consistent best practices for customization?
- Have we standardized our development process on ServiceNow?
- Have we enabled teams to use ServiceNow development, deployment, and testing capabilities consistently?

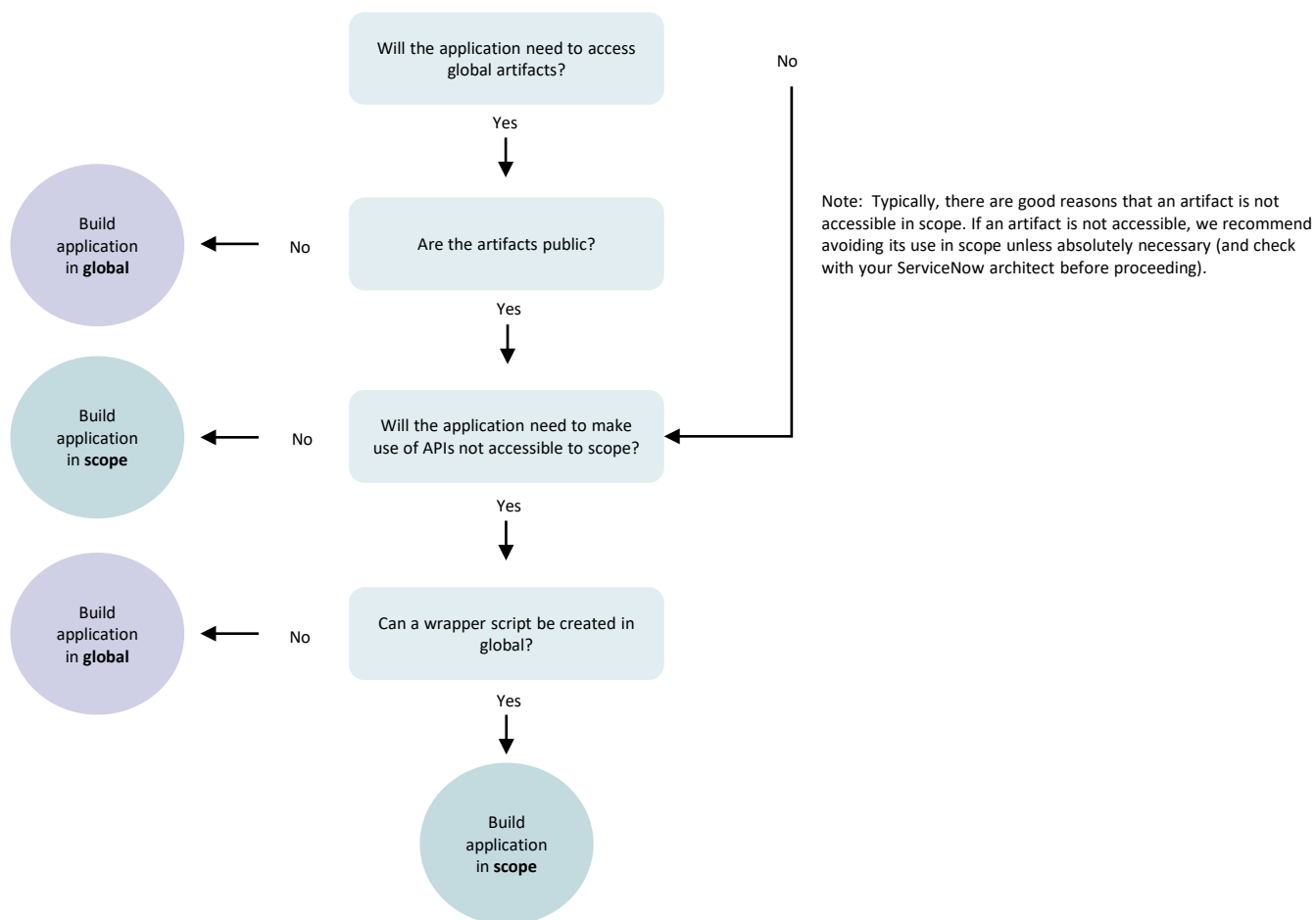
If you answered "yes" to both questions above, proceed to the [next step](#). If not, complete these action steps to complete this step:

1. **Give development teams guidance on using scoped applications to respond to customization demands** Your default for custom application development should be to *build in scope* to avoid global risk and complexity. *Scoped applications* have boundaries that:
 - Allow the system to uniquely identify application artifacts, like tables, properties, user roles, and public and private API definitions
 - Encapsulate the runtime code
 - Provide table-level data control

Application scoping protects applications by identifying and restricting access to application files and data. By default, all custom applications have a private scope that uniquely identifies them and their associated artifacts with a namespace identifier. A scoped application can access and change its own tables and business logic, while other applications cannot without explicit permission. This ensures custom apps do not interrupt core business services and other applications that do not interfere with a custom application's normal functioning.

Scoped applications cannot access the privately marked artifacts of a different application in global scope, and they should be used for building new standalone apps. When you evaluate retained customizations, which we discuss in Step 4, also identify whether you can convert any of your global apps to scoped apps to reduce system complexity. See the next page for a decision tree to help guide when an application should be global versus created in scope.





You can also use this decision tree to help identify global apps that you can convert to scoped apps. To do this, you create a copy of the application on another instance, create a new scope, copy the artifacts in scope, and deactivate the artifacts in global scope.



Global apps and ServiceNow Studio: If you create a globally scoped app in the ServiceNow Studio, you can add existing globally scoped files to it, remove files from it, or move application files between global applications.

Ensure development teams have access to training and documentation materials for implementing scoped applications. Where possible, schedule and run training for development teams on using scoped applications. See that delegated developers also have access to training and documentation.



Ensure citizen developers have the support they need to succeed: Citizen developer programs can give your Now Platform development efforts new levels of scale, velocity, and innovation. But make sure you invest in the training, support, and governance you need for citizen developers to build the right way. See our Success Quick Answer, [How should citizen developers enhance their learning?](#), and our [application development resources](#) to put this support in place.

If your organization has data and process separation requirements (typically in organizations that span multiple, independent legal entities), educate development teams on how scoped applications operate and how to meet these separation requirements using scoped extensions or ACL based segregation. Domain separation is another, although complex, option to consider in such scenarios, so that decision should be made at the executive sponsor level.

You should have a design authority—either a design board or architectural review board (ARB)—establish and communicate this guidance to development teams, with the support of your technical governance function for ServiceNow. Your design board should consist of your Now Platform architects, your development lead(s) for ServiceNow, and UI/UX subject matter experts. The design board should, in collaboration with your technical governance function, define:

- Development standards
- UI/UX standards
- Data governance standards

Additionally, your design board should review any approved demands for global custom applications to make sure that they meet standards for consistency, efficiency, and quality, and don't impact global performance.



Two key design principles for customization:

- **Avoid copying objects** – Instead, update objects in place wherever possible, except for Service Portal widgets and other items designed to be reused.
- **Default to add before edit** – This means that you should, for example, add fields to forms rather than change the type of an existing field. When adding, avoid using the same names as OOTB objects, methods, or classes. Keep the number of fields you add to a minimum—the more fields you have on a form, the longer it may take to load.

Make your development teams familiar with your organization's design principles. Your design board should communicate these clearly as you onboard new development teams, using simple and memorable language (avoid copying objects).

2. Define a standard process for development teams to build, release, and test custom applications. In addition to using scoped applications, your organization should define a standard development and release management process for custom applications using ServiceNow tools, including delegated development, ServiceNow Studio and Flow Designer, application repository, update sets, and source control:

- Use *delegated development* as your standard method for building scoped applications. Delegated development allows non-administrators who've been assigned role permissions by administrators to develop scoped apps. Delegated development also ensures that admins can restrict developers' access to only the specific file types, security records, and script fields they need to support their application. A developer who, for example, has permission to access file types for one application does not necessarily have permissions for any other application. When you use delegated development as your standard—especially in a more decentralized or distributed environment—you can be sure that your administrators can maintain effective governance and security without compromising scalability.

- Custom applications (including both global and scoped apps) are built in *ServiceNow Studio*, ServiceNow's integrated development environment. Encourage using *Flow Designer* as the default tool for building scoped applications, especially for citizen developer teams. Flow Designer provides a means for codeless development, using standard reusable components and customizable user interfaces. Using standard components to construct workflows minimizes technical risk and complexity and allows teams without prior or formal development experience to develop apps.
- Your process should use *source control* to manage development changes and resolve any collisions. Source control lets developers integrate with a GIT source control repository to save and manage multiple versions of a scoped application from a sub-production instance. Your development teams should use source control to share apps across development instances, keeping each instance independent. Developers can use separate instances to work on different features, applications, or product releases at the same time—as well as share code between these instances and resolve collisions throughout the development process. This way, your development is more distributed and changes are effectively managed in development.
- Use the *application repository* in your process to support scoped app deployment and *update sets* to support global app deployment as needed. An update set is a group of customizations you can move from one instance to another when you publish global applications only. You can also use update sets to:
 - Store changes to a baseline or installed application
 - Store and apply a particular version of an application
 - Deploy changes to installed applications across multiple instances

Remember that you can only use update sets to publish global apps, not scoped apps. Instead, publish scoped apps to the application repository. Note that you can also use the repository to manage update sets automatically so any updates made across instances are made to the latest application version only). Source control can also be used to share or install custom applications across instances. The graphic below brings together a view of the development mechanisms, management tools, and deployment methods your development process should include.

- Don't deploy custom applications, whether they're scoped or global, without a comprehensive test plan for functionality and integrations (where applicable). Within your process, ask developers and administrators to use the *ServiceNow Automated Test Framework (ATF)* to build tests for new custom applications. Test before you deploy each app as well as before and after each instance upgrade to evaluate any effects an upgrade may have on the app, particularly for global applications. After upgrading, testers should track any defects or deviations from the pre-upgrade testing results for a custom app. Defect tracking can help identify root causes and create fixes. When a fix is identified, your developers should capture the fix in a single update set. The accumulated fixes and update sets can then be applied to the production instance.

1 Recommended mechanisms for developing custom applications.

Scoped applications ensure that custom apps avoid global risk and complexity

Flow Designer enables no/low-code development based on reusable components and customizable UI

2 Tools to manage distributed development

Delegated development grants non-admin users the ability to develop applications without compromising governance

Source control enables developers to manage development changes and resolve any collisions

3 Tools to manage testing and deployment

Automated Test Framework should be used to create and run tests for custom apps

Application repository should be used to deploy scoped apps (**update sets** can be used for global apps)



Use IntegrationHub to avoid integration complexity: Customers engaged in enterprise services transformation, taking advantage of ServiceNow's 'platform of platforms' capability, will see more demands for cross-platform integration.

This doesn't have to lead to complex, customized integration. ServiceNow IntegrationHub reduces the need for complex, custom-coded integrations using third-party REST APIs as part of a workflow when a specific event occurs in ServiceNow. IntegrationHub supports three use cases:

- **Third-party REST API integrations** – Customers can post messages and ServiceNow incident, problem, and change record details to collaboration channels like HipChat, Slack, or Microsoft Teams.
- **Integrations between ServiceNow instances** – IntegrationHub provides an easy-to-configure spoke allowing for data synchronization across multiple ServiceNow instances.
- **Create REST actions** – IntegrationHub supports developing custom REST web service actions to support API development for web-based applications.

Familiarize your development teams with IntegrationHub and use it as their default option when building integrations.

Step 4: Audit customizations and configurations for retention or removal, in advance of upgrades

Over time, your strategic objectives (and the capabilities available in ServiceNow) will evolve, such that customizations and configurations may no longer have valid business justification, although they may have met a need in the past. Essentially, you have to apply portfolio management practices to customization.

Check your progress:

- Do you have a plan or process in place to review your level of customization before you upgrade?
- Do you have a metrics strategy to help you understand if your level of customization is still delivering business value?

If you answered “no” to either of the questions above, complete these action steps:

1. **Develop a process to audit customizations and configurations regularly for retention or removal.** Start by defining a set of triggers that will initiate an audit of your current customizations and configurations. This should include planned upgrades, a major strategy and/or leadership changes in your organization, and/or a redefinition of the business outcomes you want to achieve in your ServiceNow roadmap.

Begin your process by evaluating the current performance of your instance to assess whether your current level of customization is creating performance issues or presenting an upgradeability obstacle. You can use ServiceNow HealthScan to make this assessment as your starting point (and your administrators should use HealthScan regularly, regardless of triggers) to review your customization portfolio.

Second, use analytics to assess whether customization is delivering the business value you expected from the original need or use case. This should include:

- **Adoption** – How often are custom applications or configurations (such as new or modified forms) used? Is the user base sufficient to deliver value?
- **Efficiency** – Have you added too much complexity into a workflow? For example, you may examine assignment groups built into a specific configuration that have never been assigned tickets.
- **Experience** – Have you improved or worsened the user experience? Here, surveys of user satisfaction may help.

On the next page is a suggested list of metrics support your analysis.



- **Percentage of service fulfiller/administrator/developer teams that are aware of OOTB capabilities (as determined by survey).** Demands for customization may be due to a lack of awareness of ServiceNow capabilities that meet your business and user needs. If this percentage is below 80%, invest in a campaign to create this awareness.
- **Number of skipped records generated during upgrade due to customization and configuration.** This will help you identify the amount of customization implemented between upgrades. If this number grows more than expected, revisit your demand management processes to determine if you need tighter approval criteria or undertake an effort to remove unnecessary customization.
- **Percentage of customizations with clear and documented business justification.** This will help you measure your effectiveness in implementing a business-smart approach that evaluates demands in terms of business value.
- **Percentage of customizations using scoped applications.** This will help you assess the potential risk of non-scoped custom apps to global performance. If this percentage is low, you can identify whether there are global apps that you can change in scope to lower your risk.
- **Percentage of customizations adhering to technical standards.** This will help you measure how well your design board and technical governance function communicate and enforce your customization standards.
- **Percentage of customizations with approved exceptions to technical standards.** This will help you determine if your design board and technical governance functions need to reassess your technical standards for customization to respond more effectively to business demands.
- **Number of incidents affecting ServiceNow performance due to poor development practices.** This will help you determine if you need to enforce more consistent development practices or invest in more training for developers.
- **Number of customizations/configurations removed or returned to OOTB due to redundancy or lack of business value.** Use this metric as an incentive to audit your customization portfolio regularly so you can remove unnecessary customization.

Where needed, conduct interviews with relevant business leaders and/or stakeholders to determine whether customizations and configurations continue to have a demonstrated business need based on the criteria your demand board set.

Review your incident log to understand where poorly developed custom applications have affected instance performance. Remember to confirm that scoped apps are your preferred means of implementing custom apps. You should track the percentage of custom apps that are scoped versus global, to understand where global apps impact global performance and where you can shift their scope.



Should we evaluate every configuration and custom application? For larger organizations, it may not be feasible or economical to run a line-by-line review of every configuration and custom application, especially you have a large number of legacy configurations or applications. In this case, you have two options:

- **Restrict your evaluation to the riskiest points of customization** – You can decide to review only custom applications characterized by medium or high complexity (see the scenarios in Step 2) or applications with known points of overconfiguration. The tradeoff with this approach is that it can leave a number of safe but legacy points of customization in place, regardless of their value. Some of these may not affect your upgrades or performance but could affect user experience.
- **Use a zero-baseline process** – You can decide to remove any configurations and custom applications that don't have a clear need that's validated by a process, service, or workflow owner (with the exception of any customization required for regulatory or compliance purposes).

The challenge with removing legacy points of customization is that it can require organizational change management for users—such as your service desk staff—whose workflows and behaviors may have become accustomed to the workflows in your legacy configurations and custom applications. Make sure you coordinate with process owners and stakeholders before you remove any legacy points of customization so you have time to communicate these changes and socialize them with users. Make your development teams familiar with your organization's design principles. Your design board should communicate these clearly as you onboard new development teams, using simple and memorable language (avoid copying objects).

2. **Based on your findings, identify customizations and configurations that you can remove (or change) to move your implementation closer to OOTB functionality.** Make sure you evaluate the level of effort required to remove customization so you can allocate resources appropriately. Review any plan to remove customization and configuration with your strategic governance committee to determine if you have the resources you need and make sure you have executive sponsor support to prioritize this effort relative to other demands.



Build analytics expertise on your ServiceNow team. You can use Performance Analytics, HealthScan, and other resources to help ServiceNow report on itself to identify whether customizations and configurations help or get in the way of achieving your business outcomes.

Use HealthScan consistently to monitor your upgradeability. HealthScan is a no-charge tool that helps you better understand and improve your ServiceNow instance health. Built on best practice definitions from thousands of instances, HealthScan quickly scans your instance and provides data points for five key categories: security, upgradeability, performance, ease of management, and user experience. Findings are broken down by product suite, so you can see where you have technical debt and get recommendations to improve your instance, ranked by importance. If you need additional support, check out ServiceNow's Configuration Review service, conducted by a trained ServiceNow consultant.



Case in point: the power of simple ratios. One organization we spoke with kept track of two simple ratios to help assess their ongoing need to maintain and implement customization:

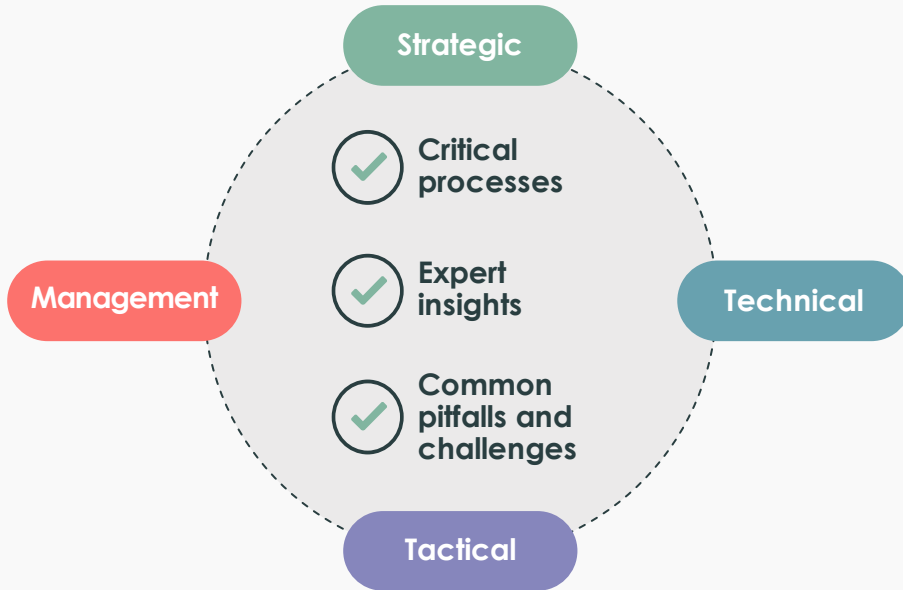
- **Ratio of (total amount of customization)/(ideal amount of customization)** – Your ideal number should represent the number of customizations you have that are backed by a clear and validated business need. This ratio should approach 1 over time, as you deprecate unnecessary customizations.
- **Ratio of (total amount of customization requested)/(total amount of customization approved for implementation)** – This measure acts as an indicator of whether incoming requests reflect clear business needs and whether they're redundant to capabilities that already exist. This ratio should approach 1 over time as stakeholders become more aware of OOTB capabilities and more accustomed to the criteria for evaluating new demands.

Customer Success Best Practices

ServiceNow's Best Practice Center of Excellence provides prescriptive, actionable advice to help you maximize the value of your ServiceNow investment.



Definitive guidance on a breadth of topics



Designed for:

-  Executive sponsors
-  Platform owners and teams
-  Service and process owners

Created and vetted by experts



Best practice insights from customers, partners, and ServiceNow teams



Based on thousands of successful implementations across the globe

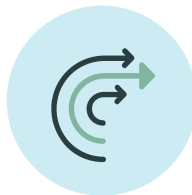


Distilled through a rigorous process to enhance your success

Proven to help you transform with confidence



Practical



Actionable



Value-added



Expert-validated

Get started today.

Visit [Customer Success Center](#).

Contact your ServiceNow team for personalized assistance.