

Manage application development on the Now Platform

Build your capabilities to manage and scale application development on the Now Platform

What's in this Success Playbook

This Success Playbook will teach you how to build your organization's ability to realize value from application development on the Now Platform®. You'll learn:

- How to build effective capabilities for managing application development
- How to scale your application development capabilities on ServiceNow®

Key takeaways

The most important things to know

Align application development on the Now Platform with your digital transformation goals. Start first by identifying development ownership and roles, then establish processes to align with scoped and global application needs. This will give you a solid foundation to build applications efficiently and expand the Now Platform's capabilities to citizen developers.

The payoff of getting this right

With effective management in place, your application development efforts will support digital transformation, so you can innovate and see value faster as you reduce development costs and minimize your technical debt and risk.

What you need to get started

You should have a governance model in place for ServiceNow (or one developed in parallel). For guidance on this, see our [Governance Workbook](#) on how to build a dedicated, dynamic governance process, policies, and team. You'll need the engagement and support of your [executive sponsor](#) to align your application development strategy with your roadmap and objectives for ServiceNow. See our [Customer Success Center resources](#) if you need help with this.

When you should start this activity

It's best to start considering how you'll manage application development as you're evaluating ServiceNow as your solution of choice. If you have already started developing on the Now Platform, determine if you have effective governance in place to support expanding your development activities.

Playbook overview

ServiceNow recommends these four steps to manage an application development strategy on the Now Platform for new business value and innovation.

Step	Outcome	
<p>Start</p> <p>At the beginning, you'll build your initial or foundational capability. This includes setting up initial frameworks, defining roles, and clarifying your objectives.</p>	<p>Step 1 – Define your application development strategy and governance requirements</p>	<p>You have a clear understanding of your objectives for application development and the decision-making authorities needed to guide application development.</p>
	<p>Step 2 – Build effective application development processes</p>	<p>You have effective processes in place to deliver high-quality applications on the Now Platform.</p>
<p>Improve</p> <p>As you improve, you'll take steps that help you reach your objectives and see value fast.</p>	<p>Step 3 – Build portfolio management capabilities</p>	<p>You have established portfolio management capabilities to curate and maintain applications that support your organization's objectives.</p>
<p>Optimize</p> <p>Last, you'll refine and expand your capabilities so you can scale as you grow and continuously get more from using ServiceNow.</p>	<p>Step 4 – Scale application development</p>	<p>You have the capability to scale application development with citizen developers.</p>

If you haven't explicitly defined your application development strategy, demand intake, and portfolio management processes, start with Step 1.

If you've already started developing apps on the Now Platform and are looking for guidance to expand development activities, start with Step 4 but look at Steps 1–3 to make sure you have a strong foundation in place.

Terms and definitions

Application – Applications are collections of files and data that deliver a service and manage business processes. For the purposes of this document, an application consists of new functionality developed with customizations and configurations.

Citizen developers – These developers are end users from outside your IT organization who create new business applications on platforms sanctioned by IT.

Delegated development – This is a ServiceNow capability that allows designated users without a system administrator role to develop applications on the Now Platform. This includes line of business development teams and/or citizen developers.

Demand management – This planning methodology is used to forecast, plan for, and manage demand.

Global application – A global application provides no boundaries compared to a scoped application, which has strong boundaries.

Governance – Governance refers to the set of policies, processes, and organizational capabilities that define how decisions are made and who is responsible for the outcomes.

No- and low-code development – These types of development include capabilities that allow developers of varied experience levels to create applications for web and mobile environments using drag-and-drop components and model-driven logic through a graphic user interface.

Migration – Migration is the process of evaluating, prioritizing, and moving applications from one operating environment to another.

Process – A process is a repeatable, structured series of steps—often described as a workflow—that's designed to accomplish a particular outcome (like application development intake).

Process owner – This person is responsible and accountable for process design and resource management to deliver quality outcomes.

Scoped application – These applications have strong boundaries and run-time isolation compared to global applications.

Shadow IT – In this environment, applications are built and used without the approval of IT and may not be consistent with requirements for control, documentation, security, or reliability.

Step 1 – Define your application development strategy and governance requirements for ServiceNow

KEY INSIGHTS

- Align your application development strategy and objectives with your ServiceNow roadmap and digital transformation goals.
- Ensure you have appropriate decision-making authorities and roles to guide application development.

Application development on the Now Platform can help IT teams be faster and more efficient in delivering value. When IT uses low- and no-code development, organizations can expand their application development beyond IT to citizen developers who can further support innovation and digital transformation. But keep this in mind: If you implement application development capabilities without strategic guidance or governance, you can create unmanageable maintenance challenges.

Forrester Research notes an example in which a large insurance company introduced a low-code platform without governance, resulting in 17,000 unsupported apps¹. To avoid that kind of disaster, make sure that your application development efforts are harnessed to your strategy for digital transformation and that you have appropriate guardrails in place to limit technical and portfolio risk.

Begin building your strategic foundation with these two steps:

1. Explicitly identify how application development on the Now Platform will support your digital transformation objectives.

When you develop applications at scale on the Now Platform, you can expand your organization's opportunities for innovation, productivity, and improved customer and employee experiences. Frame your strategy around one or more of the following objectives, based on your experience and maturity:

- Developer efficiency
- Project velocity

¹ "Use a Light Touch to Govern Low-Code Development Platforms," John Rymer, Forrester Research, February 13, 2017

- Innovation capture

2. Establish governance roles and responsibilities to ensure application development activity aligns with your objectives and guards against risk.

Once your strategy is defined, the next step is to establish decision-making roles and responsibilities for ensuring application development remains aligned with your strategy and objectives. The extent of governance you need should be calibrated to your organization's risk tolerance (for example, by establishing guidelines around where citizen development will and will not be permitted). Most importantly, good governance should *increase* your organization's velocity, establishing clarity around business value criteria, risk assessment, and decision-making authority while reducing technical debt or misaligned development.

Customer insights

Our customers' experience points to three insights you should keep in mind as you set your strategic foundation.

INSIGHT 1:

Answer three big questions to define and align your application development strategy

The biggest risk to application development is that it can become siloed across development teams and result in outcomes that aren't always in step with your digital transformation objectives. There are three big questions to consider as you set and align your application development strategy:

1. Who will build applications on the Now Platform?

Determine if application development will be limited to the ServiceNow platform team, include distributed development teams, and/or include citizen developers. Ideally, your strategy should include all three populations to maximize your efficiency and innovation gains, and you should determine how you'll introduce capabilities in phases across these populations. Your roadmap should set "entry criteria" for each population by defining training requirements and allocating appropriate resources for training and communications for each population.

2. What kind of applications will be built (and maintained)?

Your answer to this question should align with your roadmap objectives for ServiceNow and digital transformation—and determine which development efforts get prioritized first. You'll need to address what you'll migrate from legacy platforms, what you'll need to develop centrally versus in business lines, and what guidance you'll give for demand intake and management. Your demand board should be prepared to answer the question, "When does a customization become an application?" Your strategy should reinforce that

application development efforts need to meet clear business value criteria, ideally in line with the criteria explicitly laid out in your demand management process. For examples, refer to these Success Playbooks: [Manage platform demand](#) and [Avoid customization pitfalls so you can innovate and meet demand at scale](#).

3. What authorities will we install for effective governance?

Your answer to this question defines the policies you need to have in place for technical and portfolio governance that align with your objectives and organizational risk profile. At a minimum, your policies should include and define oversight authority for:

- Technical standards, approved methods, and best practices for development
- Requirements for testing and deployment
- Requirements for documentation and training
- Requirements for support, maintenance, and upgrades

Alternatives for thinking through an application development strategy

There's no single right answer for how to align your application development strategy with your digital transformation objectives, but customer experience points to several scenarios you can use to frame your strategic governance conversation:

- **Improve the speed and efficiency of IT resources against workflow projects** – Now Platform capabilities open the door for IT teams to identify and tackle major projects for workflow improvement that directly support your business objectives (such as developing an application to support patent application workflow).
Note: Keep up to speed on upcoming ServiceNow releases to make sure you don't spend time creating capabilities that are becoming available on the Now Platform.
- **Free up IT resources from lower-priority projects** – Lower-priority IT projects that can be implemented on the Now Platform with fewer developers (or with lesser-skilled developers) can allow you to reallocate resources to higher-priority projects.
- **Harness enterprise-wide innovation** – The rollout of low-/no-code capabilities can permit teams who are closest to business problems to spot and exploit opportunities for frontline innovation and value. Consider maintaining an "innovation team" of no-, low-, and pro-code developers who can help frontline users/teams build and test these opportunities quickly (rather than have them get lost in a central backlog).

Use these scenarios (and others) to help your strategic governance function and demand board assess how much demand you can anticipate for each and where you may want to concentrate your initial efforts.

Best practice – Identify strategic, technical, and portfolio governance functions

Your answers to the big questions for strategic alignment will guide the actions each of your governance functions will need to take to enable application development. See Table 1.

Strategic governance	Technical governance	Portfolio governance
<ul style="list-style-type: none"> • Ensures application development remains aligned with roadmap objectives • Establishes and monitors measures of success aligned with business value • Establishes criteria to approve application development activities • Oversees budgeting and resource allocation 	<ul style="list-style-type: none"> • Establishes technical standards and best practices • Defines development processes • Establishes authorities/guidance for testing and deployment • Defines training requirements 	<ul style="list-style-type: none"> • Defines and administers the demand intake process • Defines support/maintenance/update requirements and ownership • Defines authorities for rationalization and deprecation

Table 1: Key activities to be included in governance categories

The primary role of strategic governance is to ensure that you develop applications that are aligned with your digital transformation objectives. Your strategic governance function will decide whether applications will be built by the ServiceNow platform team, distributed development teams, and/or include citizen developers.

Technical governance will identify how you'll manage the technical aspects of development, including defining technical standards, development processes, testing, and training requirements. This will include the responsibilities and guardrails needed if/when you include line of business (LOB) development teams and/or citizen developers.

Portfolio governance will establish a demand management process. (For more information on this, see our Success Playbook, [Manage platform demand](#)). This will address how you'll capture and prioritize demand, how you'll determine how each application will be built, and who will build them. Portfolio governance will also conduct portfolio reviews to identify opportunities to extend or retire applications.

At a minimum, your governance should include policies that will:

- Define development standards and review process for any required customization
- Define integration standards
- Establish a platform management policy that defines a process for how platform decisions are made and how to evaluate the impact of changes

- Create a data governance policy that defines data ownership and sets up data standards
- Establish technical documentation formats and ownership
- Establish processes that codify and enact application best practices

Best Practice – Identify responsibilities across your governance teams and key stakeholders

Your governance functions should work closely with your LOB, global ServiceNow platform team, and shared services when establishing governance for application development. This will evolve as you move from developing applications within your ServiceNow platform team to delegating to your LOBs and citizen development. Prepare a RACI chart outlining who is responsible, accountable, consulted, and informed across these functions to reduce the risk of uncoordinated or contradictory decisions. See Figure 1 for an example. Be sure to standardize how these teams will inform and record decisions.

	Strategic Governance	Technical Governance	Portfolio Governance	LOB Application Team	Global IT
Ensure strategy roadmap aligns to business outcomes	Accountable	Informed	Informed	Responsible	Responsible
Establish and implement technical standards	Informed	Responsible	Accountable	Responsible	Responsible
Establish development model (central/delegated/citizen)	Responsible	Responsible	Accountable	Accountable	Accountable
Establish and implement demand management policies	Responsible	Accountable	Responsible	Informed	Accountable
Delegate approvals for low-risk development	Informed	Responsible	Responsible	Accountable	Accountable
Implement data governance policies	Accountable	Responsible	Responsible	Informed	Responsible



Responsible



Accountable



Consulted



Informed

Figure 1: Responsible, accountable, consulted, informed (RACI) chart

Best practice – Track the effectiveness of your governance

Ensure governance is adding value and not presenting obstacles to application development. Do a process audit biannually or annually to identify opportunities to improve governance by taking these actions:

1. Define metrics that measure the effectiveness of your governance process and policies and track them over time to identify where you can make improvements.
2. Plan how you'll identify and remove bottlenecks in governance decision-making.

3. Define how you'll communicate policies to application developers so they know how to adhere to governance policies and make decisions accordingly.
4. Make sure your governance is flexible enough to meet the changing needs of the organization. Plan to collect feedback from people in your organization about where governance is making their work harder and streamline the policies to address concerns where you can.

INSIGHT 2:

Build your application development models to respond to today's and future business needs

Your ServiceNow platform team will likely be the first team to start building applications on the Now Platform. Your governance teams will need to consider how you'll scale development beyond the platform team by addressing these things:

- You need to identify what is a good fit for the Now Platform. This impacts two areas: what you'll migrate from legacy applications and what new applications should be built on the Now Platform. We show how to do this in more detail in Step 3.
- Your technical governance team should provide explicit best practice guidance on how the capabilities available on the Now Platform will be used to build applications from no-code tools like Flow Designer to pro-code JavaScript capabilities. Your technical governance function should document, using a recipe-like format, how teams should use Now Platform capabilities for different scenarios.
- Your strategic governance function should authorize different groups to build applications on the Now Platform based on your roadmap and the entry criteria you establish. As you consider how you'll scale application development, start by thinking about how the expertise you develop will impact your ability to scale: Ideally, you'll develop expertise among professional developers to use low- and no-code capabilities on the Now Platform. This will enhance your efficiency and will provide coaching and training opportunities as you extend capabilities to citizen developers. To identify who will build applications, start with these questions:
 - What groups will have the most potential to drive strategic business objectives?
 - What level of training will you need to provide?
 - What level of process or workflow expertise exists within these teams?
 - What is your investment strategy to ensure development is appropriately resourced?

What are the benefits of a citizen developer program?

A citizen developer program is key to expanding your low-/no-code developer pool, but it needs structure: Simply opening the floodgates can compromise your ability to provide oversight to development and manage technical debt. Fears of shadow IT should not be an excuse for limiting application development. Use Table 2 to communicate the business case for resourcing a citizen development program.

As your strategy expands application development, identify a few ideal citizen developers for a pilot group. These candidates will have process expertise, be comfortable with spreadsheets and macros, and have a desire to learn no-code app development. Align your pilot citizen developers with the appropriate application team that will apply their process and business expertise. See Table 2 for a suggested profile that you can use with your business stakeholders to nominate potential citizen developer candidates. Our customers recommend starting your search in the more tech-savvy parts of your organization, such as engineering or manufacturing.

Potential benefits of citizen application development

Innovation and efficiency	Citizen developers have front-line knowledge of various business units and are aware of potential process improvements. Citizen development can drive innovation and foster creative thinking.
Speed	Wait times to launch new apps can be reduced.
Improvement	Citizen developers receive feedback directly from their colleagues who are also using their apps. When they realize a fix is needed or there's room for improvement, they can handle the adjustments independently.
Increase organization IP	Apps are developed by citizen developers using company resources. These apps are proprietary to your organization.
May reduce the risk of shadow IT	Shadow IT occurs when employees go outside of the IT-sanctioned applications and servers to do their work. Providing employees with the tools to develop their own solutions can encourage them to create viable solutions within IT-approved space.
Cost effective	Providing non-IT employees with the tools, support, and training to develop on the Now Platform can offer a cost-effective alternative than outsourcing.

Table 2: Potential benefits of citizen development

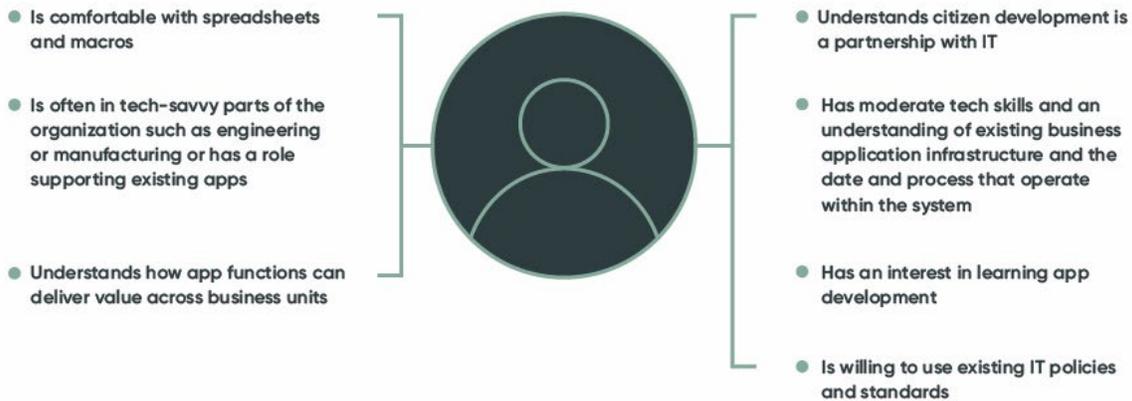


Figure 2: Characteristics of ideal ServiceNow citizen developers

INSIGHT #3:

Identify a clear ownership framework for global and scoped applications

Identify a clear ownership framework that aligns with the division between global and scoped applications on the Now Platform. Your ServiceNow platform team should own all global applications. Custom applications built by development teams should be built in scope, because it ensures that an application has boundaries that:

- Allow the system to uniquely identify application artifacts, like tables, properties, user roles, and public and private API definitions
- Encapsulates the runtime code
- Provides table-level data access control

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 don't interrupt core business services and that other applications don't interfere with a custom application's normal functioning.

Scoped applications should be owned by the LOB development team. LOB scoped applications should have designated owners, architects, and developers. These roles can be shared across scoped applications within the LOB—for example, an architect can be responsible for multiple scoped applications, as long as LOB teams are responsible for ownership and maintenance of the applications they develop. See Figure 3 to see this outlined.

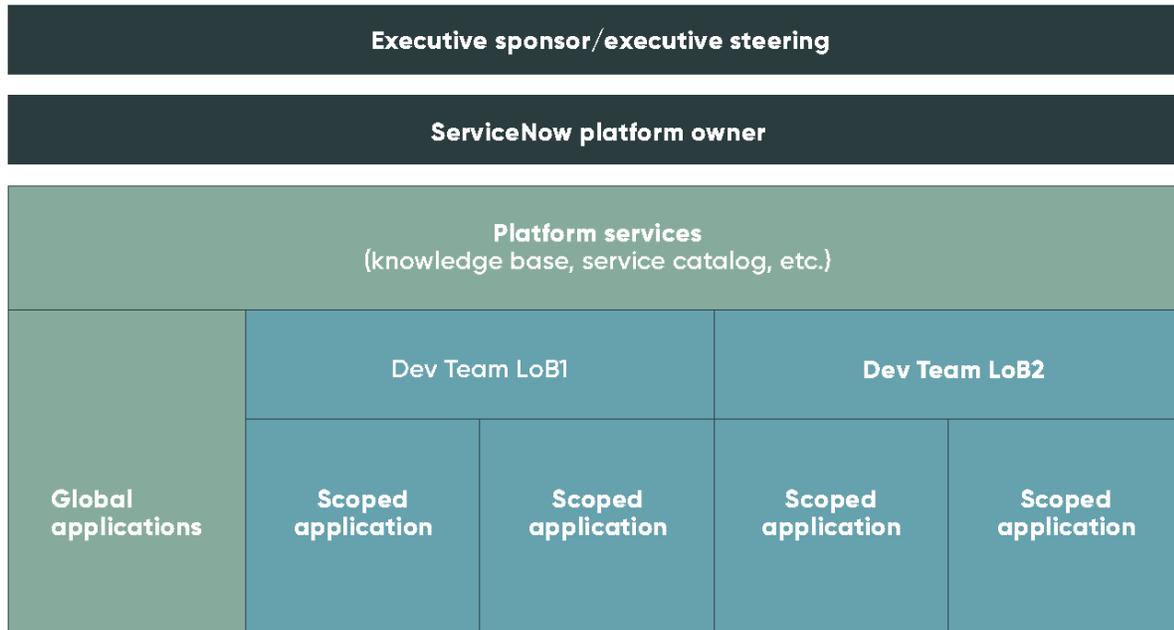


Figure 3: Sample application development ownership framework

Best practice – Every application needs a business and technical owner

Application owners are responsible for the business results that their applications support. This includes defining requirements, funding, and maintenance decisions, including recommendations for developers to act on enhancement requests or defects. These owners should be from the relevant lines of business for scoped applications.

Each scoped application should also have a designated technical owner or application architect who is responsible for the short- and long-term technical strategy of the application, including development guidance and integrations.

Your LOBs will develop portfolios as your application development expands. This may lead to an additional level of governance at a portfolio level and will include defining ownership and responsibilities that align with global authorities.

Scoped applications should also be supported by developers who manage the application's lifecycle. Citizen developers should work in partnership with the development team in their LOB to ensure visibility and avoid collisions in development activity. If you're using citizen developers, your development team should nominate a technical owner who's responsible for interfacing with the platform lead and citizen developer team.

Best practice – Have a development lead on your platform services team

Your ServiceNow platform team should include a development lead who can help to guide or coach development across the company and harmonize activities in development. Your development lead should be a senior technical resource with advanced Now Platform administration skills who can design, build, and customize ServiceNow applications and services. The development lead will work across applications, deliver new functionality, and support the entire development lifecycle and should have a software development background enhanced by training in ServiceNow system administration, scripting, and application creation.

Responsibilities may include:

- Assisting ServiceNow system administrators with incident resolution, as needed (for example, with L3 support)
- Assisting ServiceNow business analysts in estimating the level of effort required for a release
- Designing and developing new ServiceNow applications and services
- Providing mentoring and guidance for ServiceNow system administrators
- Assuming ownership and development of complex business requirements

In addition to having advanced system administration skills, your development lead should have background in enterprise or IT service management, experience with database design schemas and data modeling, and requirements gathering.

Best practice – Establish an engagement model to onboard business lines to application development

To define how you'll engage business lines in application development:

1. Sequence who you'll onboard into application teams based on your strategy and roadmap.
2. Establish the rules for when to develop in global versus scoped applications. (Scoped should be your default, with exceptions defined in the decision tree in Figure 4).
3. Identify what developers will need to understand before they start building. You'll potentially have developers sitting in different parts of your organization: within your ServiceNow platform team, within the business units in a delegated environment, and citizen developers. They'll need an understanding of what they can and cannot do on the Now Platform and have permissions aligned with the actions they're assigned. You'll need to identify and communicate what developers can do independently and what the standards are for development.
4. Identify your training needs. Development teams need to understand no-, low- and pro-code capabilities as well as create guardrails for integrations or other more complex development requirements.

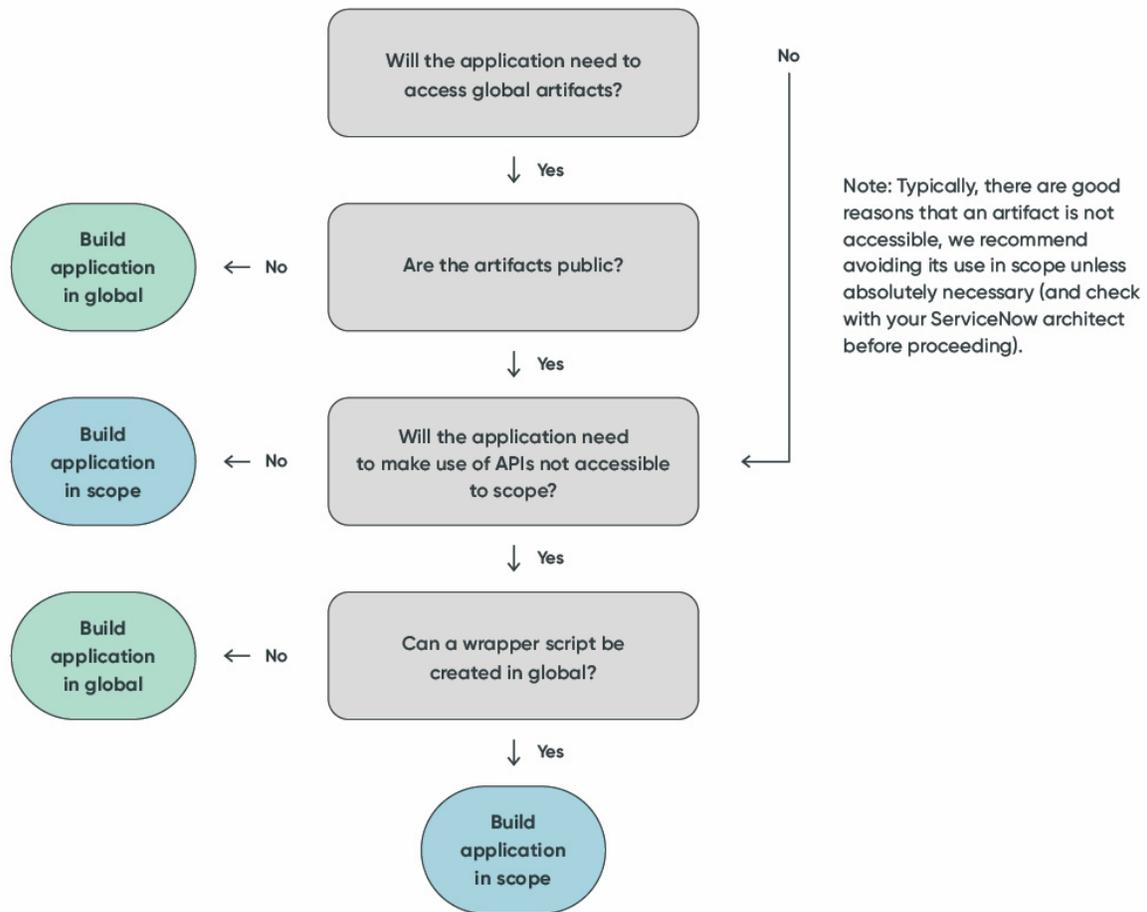


Figure 4: Deciding between global and scoped applications

What to do next

Now that you've aligned your application development strategy and objectives with your digital transformation goals, you're well prepared to begin building effective development processes. Be prepared to revisit your application decision-making authorities and roles, especially when:

- Your vision and business objectives change
- You expand ServiceNow capabilities to new teams

Step 2 – Define how global and scoped applications will move through development

KEY INSIGHTS

- Create intake and development processes that will deliver value and support growth.
- Ensure your processes align with scoped and global application development needs.
- Plan how you'll scale application development.

Now it's time to move into planning how you'll build your application development process. You need to start by defining how global and scoped applications move through application development stages.

Global and scoped applications will go through the same stages of demand and development (see Figure 5) but will take different paths on as they move through demand and development (see Table 3).



Figure 5: Application development stages

Step	Global applications	Scoped applications
Intake	Use demand intake guidelines that allow the demand board to evaluate if an application will deliver value relative to its cost.	Although intake won't go through the demand board, use the same intake criteria to focus teams on value, cost, and effort.
Demand	A demand board provides demand approval, though application owners can make small or tactical demands.	A demand board is required in specific circumstances based on scope, risk, or cost. Otherwise, the line of business and application owner can approve most scoped applications.
Build	Global applications should be built by your ServiceNow platform team. Your demand board should be supported by a design authority that provides guidance on how to develop global applications.	LOB development teams and citizen developers can work on scoped applications. Your demand board should be supported by a design authority that provides guidance on how to develop scoped applications. ServiceNow highly recommends that your citizen developers stay within a private scope to ensure they don't break any existing applications.
	Your ServiceNow platform team should be trained to work with no-, low-, and pro-code capabilities.	Your ServiceNow platform team and delegated developers will have access to no-, low-, and pro-code capabilities. Citizen developers will have permissions to use no- or low-code only.
	In most circumstances, use update sets when you publish global applications.	In most circumstances, do not use update sets to install scoped applications across instances. Instead, publish scoped applications to the application repository.
Test	Testing is centrally supported by the ServiceNow platform team. Use Automated Test Framework (ATF) and perform code reviews at the global level.	Testing is centrally supported by the ServiceNow platform team. Use ATF and perform code reviews at the scoped level.
Deploy	Readiness to deploy is determined by your ServiceNow platform team.	Readiness to deploy is determined by your ServiceNow platform team.

Table 3: Key differences between global and scoped application development

As you define your global and scoped application development processes, take these five action steps:

1. Intake – Create an intake model to capture demand requests and evaluate and prioritize demands.

To create a demand intake model, you'll need to define who can submit and approve demands and establish a demand board. Create a method to capture demand requests that ensure your organization has consistency in how it evaluates and prioritizes demands. Identify the information required to assess the relative business value of a requirement, the cost of its implementation and support, and its potential technical risks. Your design authority should be engaged—especially for global applications—to ensure the build plan meets with architectural and quality standards. See our Success Playbook on [managing platform demand](#) for more details.

Best practice – Develop an intake system to align new requests with your critical business objectives

You'll need to address intake systems for both global and scoped applications. Your intake process should help you to identify how the demand will align with the organizational goals that you identified in Step 1. Will it improve a workflow's or process's efficiency? Will it simplify employee or customer engagement with a workflow or service? Establish an intake process that will identify what the application will accomplish, who will use it, the budget and timeline, and what features will it need.

Sample intake questions

- What problem are you solving?
- Who will use this application?
- What will the application accomplish?
- What features will it need?
- Is any of the data considered sensitive or proprietary?
- What kind of reports do you need to run?
- How will people use the application?
- What is the budget?
- What is the timeline?
- Do you need data from external applications?
- What is the risk of building or not building this application?

EXPERT TIP

Consider having a ServiceNow development lead in your intake process to make sure you understand the implication a new application will have on the Now Platform moving forward—for example, is the need proposed for a new application already going to be addressed in the next ServiceNow release?

Best practice – Intake should evaluate complexity to guide decisions on who should build

Ensure your intake process captures complexity. This will help to guide development, cost, and timeline planning. Complexity isn't always immediately apparent. For example, you may have a request to calculate PTO (paid time off) balances. This might seem like a simple table and form with approval process through Flow Designer, but if you need to know an employee's current PTO balance, you may need to integrate with Workday, which bumps the application from a small to a medium effort.

A simple process like T-shirt sizing can help you identify complexity. Include the user base size and how many systems an application will need to integrate when you consider your scope and scale. Involve your development lead in the intake process so you can accurately capture the complexity, identify opportunities to simplify your solution with low- or no-code capabilities, and determine if phased development stages are an alternative.

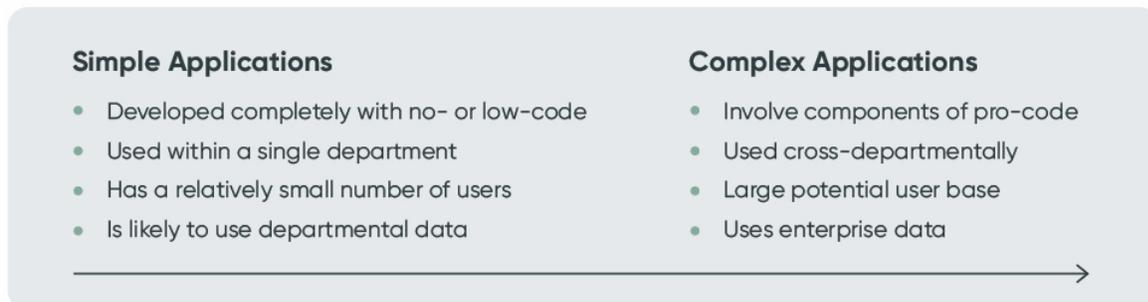


Figure 6: A comparison of simple and complex applications

Best practice – Define guidelines for how requests for new or enhanced applications will be evaluated and approved

Build a simple scorecard that helps requesters define the business value associated with a new custom application. Include a taxonomy of different types of custom application requests. Define the technical complexity associated with each type of request, using a simple low/medium/high scale, with clear explanations. Include a business value score. Additionally, you'll need to define the approvals required for each type of demand request and assign a

business analyst to evaluate incoming requests. See our Success Playbook on [avoiding customization pitfalls](#) for more details.

2. Demand – Create a consistent demand management model for global and scoped applications.

Don't begin application development until you have a demand management process in place—overseen by a demand board—to capture demand requests and evaluate and prioritize demands. Your demand board must define intake guidelines and the rules that outline what it will review and how. If you don't have a demand management model in place, look to our [resources on managing platform demand](#) for best practices. Use consistent demand intake guidelines for all applications—even those that don't go through the demand board—to help teams focus on whether an application will deliver value relative to its costs.

Best practice – Define minimum thresholds for your demand board

The development of any global application should be reviewed by a demand board. But your demand board will get bogged down if it's involved in every application development request. Define minimum thresholds that guide the demand board's involvement in decision-making to be more responsive to critical business requests. Establish thresholds to identify when the line of business can approve scoped applications. Allow application owners to approve small or tactical demands on global applications such as those required for compliance reasons or requiring fewer than 40 hours of effort. This will free the demand board to focus on more complex demands, like those that span multiple business units or those with higher risks.

3. Build – Implement a consistent application build model.

Your developers should use different tools depending on the code capabilities required and their level of training. No-code development should be restricted to declarative tools such as Portal Designer and Flow Designer. Your low-code developers should be able to use all no-code tools, along with tools such as the Guided App Creator. Your pro-code developers should be able to access all tools depending on what best aligns with their need, including more sophisticated tools like JavaScript and ATF. See our Success Quick Answer on how [citizen developers can enhance their learning](#) and go to the ServiceNow [developers' site](#) for extensive information and guidance.

When starting, line-of-business development teams should create a development process with appropriate coaching from the ServiceNow platform team where needed. As your implementation matures, development teams should adopt Agile or DevOps practices, coordinated by product owners representing ServiceNow capabilities IT workflow

Best practice – Identify the right potential applications for line of business and citizen developers and look for opportunities to build in phases

When LOB and citizen developers are early in their skill development, identify applications that require high domain knowledge and low technical knowledge as potential candidates for them to work on. Look for use cases that might be appropriate to address with a low- or no-code capability for basic functionality, and phase in pro-code capabilities, such as advanced automation, at a later point in your roadmap as necessary. With this approach, you can launch quickly while giving your developers time to grow and mature their skills. If you're expanding development to distributed development teams and citizen developers, this will allow you to engage them more quickly.

EXPERT TIPS

Look to build applications that represent quick wins for development on the Now Platform. Our customers recommend that you start by looking at spreadsheets that are emailed, modified, and emailed again. These are your most likely quick wins and could be great candidates for low- or no-code development using the ServiceNow Guided App Creator.

If you're a large company with several instances, you may improve your ability to scale by creating instances that a development team could use for a limited time (similar to checking out a book at a library). In a "checked out" instance, the ServiceNow platform team can clone the production environment to a sub-production environment, scrub any data it shouldn't have, and give it to a delegated developer who will do the development work and build update sets. When the developer is finished, the ServiceNow platform team ensures it meets code standard, runs ATF use cases against the work, and then promotes it to production. The ServiceNow platform team then reclaims the instance, wipes it, and makes it available for reuse.

Best practice – Establish processes to support standardized documentation

Many organizations report that their application development best practices are often ad hoc. While this may seem to avoid speedbumps today, insufficient documentation or failure to adhere to best practices can create challenges to your timeline and strategy. Good documentation will make it easier to onboard new developers, track development status, and continue work if a developer leaves the company.

At a minimum, establish review policies, maintenance guidelines and practices, naming conventions and documentation, UX/UI standards, testing protocols, and security requirements. Ensure these are supported by communication and strict adherence policies. Consider how you'll communicate and enforce these guidelines, including disqualifying or limiting developers who repeatedly fail to comply.

4. Test – Make sure testing is centralized and can scale across both global and scope applications.

All testing should include [ATF](#), code review, peer review, user acceptance testing (UAT), and requirements verification. While lines of business may conduct testing, oversight should be centralized by the ServiceNow platform team. In mature implementations, there may be dedicated testing/QA resources on the platform team.

No-code and low-code apps will test more quickly than their more complex counterparts. For complex apps, use unit testing for the code base where there are scripts, script includes, and business rules. Unit testing allows you to do testing on discrete functions to ensure the

inputs and outputs behave as expected. A unit test should be repeatable to check the behavior of an app any time the team adds a new requirement or makes an adjustment. If the app is properly constructed and follows technical best practices—for example, your code is in reusable blocks—you can create some effective unit tests that can be run relatively quickly.

In addition, you'll need to conduct UAT. Ask users, customers, or other authorized stakeholders to perform UAT. Developers should be responsible for remediating any issues that arise. Testing for complex applications should also include load/performance testing.

Best Practice – Make testing as automated as possible

Use ATF for all global and scoped applications. With ATF, you create and run automated tests on your ServiceNow instance when you upgrade or modify an instance to confirm that the instance still works as designed. ATF will make it easy to run the test or test suites so you can remediate any issues. See our Success Quick Answers on [how to use ATF](#) and [how to create tests](#).

EXPERT TIP

Don't assume that no-code apps don't require some review before moving to production: A citizen developer could create an application that sends a large number of unintended notifications or exposes sensitive data. Include citizen developers in your code review process to coach them and facilitate teaching and knowledge sharing. This effort should improve code quality across the board.

5. Deploy – The decision to deploy should be made by your ServiceNow platform team.

Business line development teams should participate in testing, but the ServiceNow platform team—likely the application architect—should own the testing to ensure deployment is within the release management process and be empowered to make the final decision that an application is fully tested and ready to deploy.

What to do next

Now that you've created an intake and development process, you're well positioned to deliver value and support growth. Be prepared to revisit the processes you've defined. Look for opportunities to improve and refine processes for both scoped and global applications to make sure you're well positioned to scale.

Step 3 – Build effective portfolio management capabilities

KEY INSIGHTS

- Maximize value on the Now Platform by selecting the right application candidates.
- Make sure your development and portfolio management processes make it easier to curate and maintain applications.

In Steps 1 and 2, you learned the importance of establishing portfolio governance and effective demand processes. Now let's go over how to build portfolio management capabilities as you build and grow. To avoid becoming saddled with technical debt, manage and curate the applications you build. Follow these steps:

1. **Identify, communicate, and enforce the best fit guidance for applications on the Now Platform.**

Create a process that will help you identify what legacy applications you'll migrate to the Now Platform as well as the criteria for identifying new applications that are the best candidates for the Now Platform.

2. **Implement portfolio reviews to ensure apps are driving business value.**

When you follow review processes, you can make sure that your application portfolio remains aligned with your strategic objectives—and you can limit technical debt.

Customer insights

Our customers' experience points to two insights you should follow as you build your portfolio management capabilities.

INSIGHT 1:

Use concrete, specific examples to show development teams which application candidates are right for the Now Platform

Some applications will be better suited for the Now Platform than others. Identify the best candidates for the Now Platform, both for legacy applications you plan to migrate and new applications you plan to build, using specific examples that show development teams what constitutes the best fit.

Best practice – Identify candidates for migration to ServiceNow from your current app inventory based on specific use cases

Provide your development teams with specific use cases to spot legacy applications that can be migrated to ServiceNow. Start with these:

1. Applications running in other low-code environments (like Excel or access-based applications)
2. Process-based applications that have a single purpose and are good candidates to be migrated, retired, or replaced
3. Process-based applications with significant pain points – Look at process performance data to find applications with bottlenecks, long cycle times, or unnecessary data collection requirements
4. Applications on legacy platforms that are no longer supported, such as [Lotus Notes](#)

This process should be iterative, and you can provide incentives to business line development teams or citizen developer communities of practice to hunt for applications that meet these criteria.

Best practice – Use specific examples of requirements that can identify optimal candidates for development on the Now Platform

Your business analysts should use flags in the requirements development process to determine when a potential application is an optimal candidate for the Now Platform (and, conversely, when an application isn't a good fit). Requirement flags for potentially good and poor fits for the Now Platform are outlined in Table 4. Embed these criteria in your business analyst training, requirements evaluation, and demand evaluation process in order to channel the candidates with the right fit that can most effectively deliver value.

Potential good fits for Now Platform	Potential poor fits for Now Platform
<ul style="list-style-type: none"> • Uses simple forms • Uses forms to interact with data extensively • Requires auditing and reporting • Includes a process that is repeatable • Requires data security rules • Requires mobile functionality • Includes processes managed in spreadsheets and email • Includes a request fulfillment pattern 	<ul style="list-style-type: none"> • Uses unstructured data • Requires graphics processing or video/audio streaming • Includes processes that are not structured and repeatable • Requires scientific modelling • Requires sub-millisecond response times

Table 4: Requirements that indicate potential good and bad fits for the Now Platform

EXPERT TIP

If you're new to ServiceNow application development, start by looking for candidates with light processes behind them. This will allow your development teams to begin building skills on ServiceNow before you look to applications with more complex logic or in-depth integrations.

INSIGHT #2:

Use concrete indicators to support portfolio management

Ongoing portfolio management can sometimes be neglected, but your demand board should work with portfolio managers to revisit and audit applications built on the Now Platform to ensure they continue to deliver value. Use specific indicators and measures that prompt your demand board to action to avoid neglecting portfolio management.

Best Practice – Define the triggers for application portfolio review

While you should aim to undertake an application portfolio review biannually or annually, don't allow those reviews to simply be a function of the calendar. Define clear triggers for a portfolio review that include:

- **Upgrades** – Use your upgrade process to determine if any applications in your portfolio have been superseded by new functionality. You can also ask business owners for feedback on the value currently delivered by applications supporting their business lines as you educate them about your upgrade.

- **Strategy change** – Trigger a portfolio review any time there is a significant change in your business or IT strategy that affects your ServiceNow roadmap.
- **Leadership change** – If your executive sponsor changes, trigger a portfolio review to make sure it aligns with your new sponsor's objectives.

You may define other triggers that fit your business context. What's important is that managing the portfolio doesn't become stale or deprioritized amid other activities.

Best practice – Use value measures to support application curation

Decisions that are made to retire or consolidate applications must be guided by the right metrics. For example, adoption doesn't equate to quality—something may have high adoption rates simply because there is no better alternative. Similarly, an application with relatively lower usage could be delivering compelling business results.

Create a value management score for each application based on metrics that align with your objectives. This can be a simple, 1–5 scoring across these categories that you can weigh based on your business context:

- Number of process users, end users, and/or customers served
- Estimated financial savings
- Estimated cycle time savings

You can build in additional categories, such as user experience and/or satisfaction. The point is to ensure that you have a reliable barometer that can help you compare applications and make directionally correct decisions about what to retire, consolidate, and maintain. Explicitly define the ownership of this scoring (and decisions regarding your portfolio) in your portfolio governance policies.

Best practice – Use Application Portfolio Management to align and curate your apps

Deploy Application Portfolio Management (APM) for visibility into your applications. By using APM, you'll have visibility into when your app is down, its problems, the cost in labor hours to make enhancements or fixes, the cost of the infrastructure it runs on, and its license costs. This insight will help you to rationalize applications by comparing application value (using a simple value measurement framework, as outlined above), relative to cost.

What to do next

At the end of this step, you should be selecting the right candidates for the Now Platform and effectively curating and maintaining applications. As you scale, it will become increasingly

important that you can effectively implement effective portfolio management to minimize technical debt and support your organization's objectives. Be sure to go back and audit your processes. Eliminate any bottlenecks. Solicit feedback from development teams and business users to identify any opportunities to improve and potentially consolidate.

Step 4 – Expand your community of citizen developers

KEY INSIGHTS

- Show citizen developers what “great” looks like on the Now Platform.
- Use value stories and communities of practice to inspire participation in your citizen developer program.

To expand citizen development, establish resources that will orient your potential developers to ServiceNow capabilities, train them in best practices, and provide ongoing guidance to ensure your applications consistently adhere to best practices. Complete these action steps:

1. Invest in orienting citizen developers to ServiceNow capabilities and their potential value.

Don't assume that potential citizen developers understand the Now platform. You need to ensure your citizen developers are prepared to participate in application development. This orientation should provide citizen developers with:

- **An orientation to the Now platform** – Provide orientations to the Now platform that can be recorded and made available on demand. These orientations should include what the Now Platform is, what the capabilities are, and how developers might participate in application development.
- **Hands-on training and support** – After orientation, training is essential. Provide hands-on training, supported by ServiceNow training content, that lets your pilot group get an instance and build its first app(s) on the developer portal. Be sure to train citizen developers on features they may overlook, such as emailing scheduled reports, and test their knowledge in the classroom before you allow them to develop apps on their own. Support your initial training with coaching and/or office hours and establish a community of practice (both virtually and with regular meetups, if possible) that supports and shares best practices. Look to the “No-Code (Citizen Developer) Guide for the Now Platform” and “Developing as a Team” for guidance on how to automate, extend, and build applications.
- **Clearly communicated guidelines and policies** – Citizen developers will also need to know what guidelines and policies are in place that will act as guardrails for their development. Orientation must include what citizen developers can and cannot do and where to get assistance or approvals.

2. Promote standardized development guidelines to citizen developers.

Your application development process will have established intake and demand processes, review policies, maintenance guidelines and practices, naming conventions and documentation, UX/UI standards, testing protocols, and security requirements. Ensure these are communicated without unnecessary jargon so people outside your IT organization can understand them. This will promote standardization and maintainability to ensure developers are solving problems in a similar manner.

INSIGHT:

Show citizen developers what great looks like on the Now Platform

Active citizen developer programs should be supported by marketing that shows the “art of the possible” to groups who may not be sure whether they should take advantage of Now Platform opportunities. Use frequent communications to share value stories to grow your citizen developer community.

Best practice – Collect and show value stories to inspire potential citizen developers

As you deploy applications on the Now Platform, communicate market examples showing that ideas generated by citizen developers have resulted in innovative applications and experiences. Make sure that your executive sponsor publicly recognizes (or even rewards) applications that produce measurable business returns so other ideas (and potential participants in citizen development) can be solicited.

Best practice – Build a community of practice among your citizen developers

Your development lead should organize a formal community of practice among your citizen developers so they can learn, share, and grow. A successful community of practice will include citizen developers from across your organization who are actively involved in application development. This can help promote consistency among your citizen developers and provides a forum your development lead can use to communicate policy and best practices, identify champions, and promote organizational change management. A typical community of practice provides a range of offerings, including networking opportunities, knowledge sharing, social events, and mentoring.

What to do next

At the end of this step, you'll have effectively scaled application development to include citizen developers. Solicit feedback from your citizen developers, ServiceNow platform team, and business units to find opportunities to improve your training and expand your citizen developer pool.

The takeaway

Focus first on building a solid application development foundation by establishing governance, process, and expertise within the IT organization on no-, low-, and pro-code development. Ensure you've identified ownership and roles as well as established effective processes to curate and maintain your applications. This will give you a solid foundation to build innovative apps efficiently and ensure that you can expand to citizen developers effectively.

What does “good, better, and best” look like for this activity?

Good – You're developing applications on the Now Platform within your IT team and building skills using no-, low-, and pro-code development.

Better – You've established robust governance, processes, and expertise to extend application development to citizen developers in partnership with IT to support your organization's goals.

Best – You have application development happening in partnership between your IT organization and lines of business and have deployed citizen developers with clearly identified roles, responsibilities, and processes.

What should I convey to my team?

Application development can enable speedy innovation. Focus first on building a solid application development foundation with governance and a process in place. Governance and process will establish guardrails to protect against unacceptable risk. This will give you a solid foundation for building innovative apps efficiently and expanding development to citizen developers.

Additional resources

- [Build a dedicated governance process, policies, and team](#)
- [Manage platform demand](#)
- [Avoid customization pitfalls so you can innovate and meet demand at scale](#)

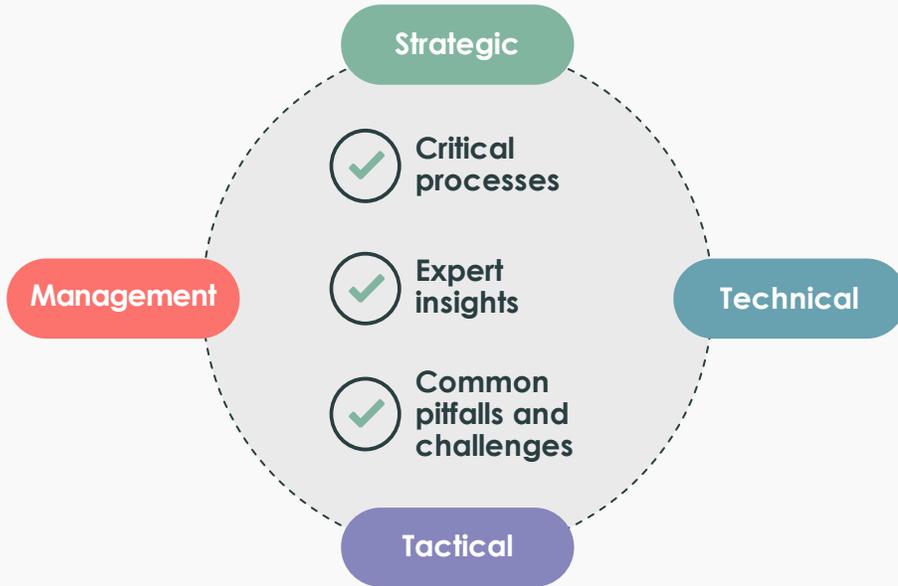
If you have any questions on this topic or you would like to be a contributor to future ServiceNow best practice content, please contact us at best.practices@servicenow.com.

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.