Manage application development on the Now Platform

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

Developed by the ServiceNow Best Practice Center of Excellence

March 2021
What is the payoff of getting this right?

Application development on the Now Platform can help developers of all levels be faster and more efficient in delivering value. Low- and no-code development can enable organizations to expand their application development beyond their normal development teams to include citizen developers who can further support speed and innovation. With effective management in place, your application development efforts will support digital transformation, so you can 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.

What will this workbook help me do?

This Success Workbook 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 focus first on building a solid application development foundation with governance and a process in place. This governance and process will establish guardrails to protect against unacceptable risk and give you a solid foundation for building innovative apps efficiently and expanding development to citizen developers.

How to use this Success Workbook

This workbook will guide you through the steps to get started with ServiceNow governance.

Start by reading through the Workbook Checklist and review the steps and tasks to get started with ServiceNow governance. From the checklist page, you can either proceed through the workbook page by page or navigate only to the sections that you need using the hyperlinks.

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 if you’ve already done so and 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: Manage application development on the Now Platform

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

- 1a: Identify how application development on the Now Platform will support your digital transformation objectives
- 1b: Identify strategic, technical and portfolio governance functions
- 1c: Build your application development models to respond to today’s and future business needs
- 1d: Identify a clear ownership framework for global and scoped applications

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

- 2a: Create an intake model to capture demand requests and evaluate and prioritize demands
- 2b: Create a consistent demand management model for global and scoped applications
- 2c: Implement a consistent application build model
- 2d: Implement a centralized and scalable testing and deployment strategy

Step 3: Build effective portfolio management capabilities

- 3a: Provide best fit guidance for applications on the Now Platform
- 3b: Use specific indicators to support portfolio management

Step 4: Expand your community of citizen developers

- 4a: Expand your community of citizen developers
- 4b: Show citizen developers what great looks like on the Now Platform

Terms & Definitions

Additional resources
Step 1a: Identify how application development on the Now Platform will support your digital transformation objectives

Developing applications on the Now Platform can expand your organization’s innovation and productivity, as well as improve customer and employee experiences. To achieve this, develop an application development strategy that aligns with your digital transformation objectives.

Check your progress:

- We have a clear application development strategy that supports our digital transformation objectives.
- We know what kind of applications should be built on the platform to support our strategy.
- We know who will be able to build applications on the Now Platform.

If you answered “yes” to the questions above, proceed to the next step. If not, complete these action steps to complete this step:

1. **Identify how application development on the Now Platform will support your digital transformation objectives.** Here are three examples that you might use to frame your thinking on how application development on the Now Platform will support your digital transformation objectives:
   
   - **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).
   
   - **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- and 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 and teams build and test these opportunities quickly (rather than have them get lost in a central backlog).

Return to workbook checklist
2. **Identify what kind of applications should be built (and maintained) on the Now Platform to support your digital transformation objectives.**

   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. **Identify who will build applications on the Now Platform.**

   Determine if application development will be limited to the ServiceNow platform team or if it will also include distributed development teams and/or 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.
Step 1b: Identify strategic, technical and portfolio governance functions

Your answers to the three points in the above section will inform your efforts to establish decision-making roles and responsibilities to make sure your application development remains aligned with your objectives and organizational risk profile.

Check your progress:

- We have identified strategic, technical, and portfolio governance functions.
- We have determined how governance functions will work with our LOB, global ServiceNow platform team, and shared services when establishing governance for application development.

If you answered “yes” to the questions above, proceed to the next step. If not, complete these action steps to complete this step:

1. Identify what you need from strategic, technical, and portfolio governance.

   There are three different types of governance you need to consider:

   - **Strategic governance will ensure that you develop apps that are aligned with your digital transformation objectives** – Your strategic governance function will set the strategy that will inform what applications will be built by the ServiceNow platform team, distributed to development teams, and/or include citizen developers.

   - **Technical governance will identify how you’ll manage the technical aspects of development.** This includes 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.
Strategic governance

- 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

Technical governance

- Establishes technical standards and best practices
- Defines development processes
- Establishes authorities/guidance for testing and deployment
- Defines training requirements

Portfolio governance

- Defines and administers the demand intake process
- Defines support/maintenance/update requirements and ownership
- Defines authorities for rationalization and deprecation

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

- Define your development standards and review process for any required customization.
- Define your 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.

2. Identify the responsibilities for 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 development to your LOBs and citizen developers.

Prepare a RACI chart outlining the functions that will be responsible, accountable, consulted, and informed to help you avoid making uncoordinated or contradictory decisions. Be sure to standardize how these teams will inform and record decisions.

Return to workbook checklist
Step 1c: Build your application development models to respond to your business needs today and the future

Your ServiceNow platform team will likely be the first team to start building applications on the Now Platform, so your governance teams need to consider how you’ll scale development beyond the platform team.

Check your progress:

- We’ve identified what a good fit for the Now Platform is.
- Our technical governance team has provided best practices for development.
- We’ve determined who is authorized to build on the Now Platform.

If you answered “yes” to the questions above, proceed to the next step. If not, complete these action steps to complete this step:

1. 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.

2. Make sure your technical governance team provides explicit best practice guidance.

   This guidance should focus on how developers will use the capabilities available on the Now Platform to build applications using no-code tools, like Flow Designer, to pro-code JavaScript capabilities. Your technical governance team should also document how teams should use Now Platform capabilities for different scenarios.

3. Plan to scale application development based on your roadmap and entry criteria.

   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. For more information on this, see our What are the benefits of citizen development Success Insight. To identify who will build applications, start with these questions:

   - What groups will have the most potential to move strategic business objectives forward?
   - What level of training will you need to provide?
   - What level of process or workflow expertise exists within these teams?
   - What investment strategy will you follow to appropriately resource development?

Return to workbook checklist
Step 1d: Identify a clear ownership framework for global and scoped applications

Identify a clear ownership framework that supports the differences between global and scoped applications on the Now Platform.

**Check your progress:**

- Our ServiceNow platform team owns all global applications.
- We have clear ownership frameworks for our global and scoped applications.

If you answered “yes” to the questions above, proceed to the next step. If not, complete these action steps to complete this step:

1. **Define clear global and scoped application ownership.**

   Your ServiceNow platform team should own all global applications. Scoped applications should be owned by The LOB development teams should own the scoped applications so they can make sure they maintain appropriate boundaries.

   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 the LOB teams are responsible for owning and maintaining the applications they develop.

---

<table>
<thead>
<tr>
<th>Executive Sponsor/ Executive Steering Committee</th>
<th>ServiceNow Platform Owner</th>
</tr>
</thead>
<tbody>
<tr>
<td>Platform Services (knowledge base, service catalog, etc.)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Global applications</th>
<th>Dev Team LoB1</th>
<th>Dev Team LoB2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scoped App</td>
<td>Scoped App</td>
<td>Scoped App</td>
</tr>
<tr>
<td>Scoped App</td>
<td>Scoped App</td>
<td>Scoped App</td>
</tr>
<tr>
<td>Scoped App</td>
<td>Scoped App</td>
<td>Scoped App</td>
</tr>
</tbody>
</table>
2. **Ensure every application has a business and technical owner.**

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. Be sure to follow these best practices:

- **Give scoped applications joint ownership** – 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** – Make sure that your citizen developers work with the development team in their LOB for visibility and to 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.

3. **Include a development lead on your platform services team.**

Your ServiceNow platform team should include a development lead who can help guide or coach development across the company and harmonize the 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, so they 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 a background in enterprise or IT service management, experience with database design schemas and data modeling, and requirements gathering.
4. Establish an engagement model to onboard business lines to application development.

Define how you’ll engage business lines in application development. Here’s how:

- **Sequence who you’ll onboard into application teams based on your strategy and roadmap** – Establish rules to determine when to develop in global versus scoped applications. Here’s an example of what that might look like:

- **Identify what developers need to understand before they start building** – You may 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 to know what they can and cannot do on the Now Platform—and their permissions assigned to them accordingly. Make sure developers know what the standards for development are and what they can do independently.

- **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.

Return to workbook checklist
Step 2a: Create an intake model to capture demand requests and evaluate and prioritize demands

Now it’s time to move into planning how you’ll build your application development process. Global and scoped applications will go through the same process steps—intake, demand, build, test, and deploy—but keep these primary differences in mind:

<table>
<thead>
<tr>
<th>Step</th>
<th>Global applications</th>
<th>Scoped applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intake</td>
<td>Use demand intake guidelines that allow the demand board to evaluate whether an application will deliver value relative to its cost.</td>
<td>Although intake won’t go through the demand board, use the same intake criteria to focus teams on value, cost, and effort.</td>
</tr>
<tr>
<td>Demand</td>
<td>A demand board provides demand approval, but application owners can make small or tactical demands.</td>
<td>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.</td>
</tr>
<tr>
<td>Build</td>
<td>Have your ServiceNow platform team build global applications. Support your demand board should be supported with a design authority that provides guidance on how to develop global applications.</td>
<td>LOB development teams and citizen developers can work on scoped applications. Support your demand board should be supported with a design authority that provides guidance on how to develop scoped applications. ServiceNow highly recommends that your citizen developers stay within a private scope, so they don’t break any existing applications.</td>
</tr>
<tr>
<td></td>
<td>Train your ServiceNow platform team to work with no-, low-, and pro-code capabilities.</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>In most circumstances, use update sets when you publish global applications.</td>
<td>In most circumstances, do not use update sets to install scoped applications across instances. Instead, publish scoped applications to the application repository.</td>
</tr>
<tr>
<td>Test</td>
<td>Testing is centrally supported by the ServiceNow platform team. Use Automated Test Framework (ATF) and perform code reviews at the global level.</td>
<td>Testing is centrally supported by the ServiceNow platform team. Use ATF and perform code reviews at the scoped level.</td>
</tr>
<tr>
<td>Deploy</td>
<td>Readiness to deploy is determined by your ServiceNow platform team.</td>
<td>Readiness to deploy is determined by your ServiceNow platform team.</td>
</tr>
</tbody>
</table>
1. Develop an intake system to align new requests with your critical business objectives.

You’ll need to create intake systems for both global and scoped applications. Your intake process should help you 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 it will need.

Sample intake questions include:

- 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?

Practitioner insight: 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?
2. **Guide decisions on who should build by capturing an application’s complexity during intake.**

Simple applications will be developed with no- or low-code development, used within a single department, have a relatively small numbers of users, and are likely to use departmental data. By contrast, complex applications will involve components of pro-code development, be used cross-departmentally, have a large potential user base, and/or use enterprise data. Evaluate complexity during your intake by doing these things:

- **Be certain you capture the app’s complexity during your intake process** – This helps guide development, cost, and timeline planning. Complexity isn’t always immediately apparent. For example, you may have a request to calculate PTO balances. This might seem like a simple table and form with an 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.

<table>
<thead>
<tr>
<th>Simple Applications</th>
<th>Complex Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Developed completely with no- or low-code</td>
<td>• Involve components of pro-code</td>
</tr>
<tr>
<td>• Used within a single department</td>
<td>• Used cross-departmentally</td>
</tr>
<tr>
<td>• Has a relatively small number of users</td>
<td>• Large potential user base</td>
</tr>
<tr>
<td>• Is likely to use departmental data</td>
<td>• Uses enterprise data</td>
</tr>
</tbody>
</table>

- **Use a simple process like T-shirt sizing to 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.

3. **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. Also 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.
Step 2b: 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.

Check your progress:

- We have a demand management model in place for global and scoped applications.
- We have defined minimum thresholds for our demand board.

If you answered “yes” to the questions above, proceed to the next step. If not, complete these action steps to complete this step:

1. **Define intake guidelines and the rules that outline what your demand board will review and how.**

   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. If you don’t have a demand management model in place, look to our resources on managing platform demand for best practices.

2. **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.

Return to workbook checklist
Step 2c: Create a consistent demand management model for global and scoped applications

Always follow development best practices and mitigate potential technical debt with a consistent application build model.

Check your progress:

- We’ve implemented a consistent application build model.
- We’ve identified the right potential applications to build.

If you answered “yes” to the questions above, proceed to the next step. If not, complete these action steps to complete this step:

1. **Allow developers to use different tools depending on the code capabilities required and their level of training.**

   Restrict no-code development 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 like the Guided App Creator. Give your pro-code developers access to all tools, depending on their needs, 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.

2. **Call on your LOB development teams to create a development process with appropriate coaching from the ServiceNow platform team where it’s needed.**

   As your implementation matures, development teams should adopt the agile or DevOps practices that the product owners representing the ServiceNow capabilities IT workflow coordinated.

3. **Identify the right potential applications for LOB and citizen developers and look for opportunities to build in phases.**

   When LOB and citizen developers are early in their skill development, identify the 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.
4. 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, you should do the following three steps:

- 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.

**Practitioner insight:** 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.

**Practitioner insight:** 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 checks it to make sure it meets code standards, 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.
Step 2d: Implement a centralized and scalable testing and deployment strategy

Testing should be centralized and scalable across both global and scoped applications.

Check your progress:

- We have a robust testing plan for all applications.
- Our testing plan accommodates the needs of complex applications.
- We have a deployment plan for each application.

If you answered “yes” to the questions above, proceed to the next step. If not, complete these action steps to complete this step:

1. **Include ATF, code review, peer review, user acceptance testing (UAT), and requirements verification in your testing plans.**
   
   While lines of business may conduct testing, your ServiceNow platform team should oversee the testing by centralizing it. In mature implementations, there may be dedicated testing/QA resources on the platform team.

2. **Use unit testing for the code base when there are scripts, script includes, and business rules for complex apps.**
   
   Unit testing lets you to do testing on discrete functions so you can make sure 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 teams can run relatively quickly.

3. **Conduct UAT for all apps.**
   
   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 and performance testing.

4. **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.
5. **Restrict the decision to deploy to 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. The platform team and/or application architect should be empowered to make the final decision that an application is fully tested and ready to deploy.

**Practitioner insight:** 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 that 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.
Step 3a: Provide best fit guidance for applications on the Now Platform

In Steps 1 and 2, you learned the importance of establishing portfolio governance and effective demand processes. Now let’s look at portfolio management capabilities as you build and grow. The first step is to identify, communicate, and enforce best fit guidance for applications on the Now Platform.

Check your progress:

- We’ve given our development teams clear examples of best fit candidates for the Now Platform.

If you answered “yes” to the question above, proceed to the next step. If not, complete these action steps to complete this step:

1. **Identify candidates for migration to ServiceNow from your current app inventory based on specific use cases.**

   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.

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

   - Applications running in other low-code environments (like Excel or access-based applications)
   - Process-based applications that have a single purpose and are good candidates to be migrated, retired, or replaced
   - 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
   - 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.

2. **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).
Embed these criteria in your business analyst training, requirements evaluation, and demand evaluation process to channel the candidates with the right fit that can most effectively deliver value. You can find examples of best fit usage guidelines [here](#).

<table>
<thead>
<tr>
<th>Potential good fits for Now Platform</th>
<th>Potential poor fits for Now Platform</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requires auditing and reporting</td>
<td>Uses unstructured data</td>
</tr>
<tr>
<td>Includes a process that is repeatable</td>
<td>Requires graphics processing or video and/or audio streaming</td>
</tr>
<tr>
<td>Requires data security rules</td>
<td>Includes processes that are not structured and repeatable</td>
</tr>
<tr>
<td>Requires mobile functionality</td>
<td>Requires scientific modelling</td>
</tr>
<tr>
<td>Includes processes managed in</td>
<td></td>
</tr>
<tr>
<td>spreadsheets and email</td>
<td></td>
</tr>
<tr>
<td>Includes a request fulfillment</td>
<td></td>
</tr>
<tr>
<td>pattern</td>
<td></td>
</tr>
<tr>
<td>Requires data inputs from multiple</td>
<td></td>
</tr>
<tr>
<td>organizations</td>
<td></td>
</tr>
<tr>
<td>Starts simple but grows in complexity and stays in the platform</td>
<td></td>
</tr>
<tr>
<td>Uses simple forms</td>
<td></td>
</tr>
<tr>
<td>Uses forms to interact with data</td>
<td></td>
</tr>
<tr>
<td>extensively</td>
<td></td>
</tr>
</tbody>
</table>

**Practitioner insight:** If you’re new to ServiceNow application development, start by looking for candidates with light processes behind them. This helps your development teams begin building skills on ServiceNow before you look to applications with more complex logic or in-depth integrations.
Step 3b: Use specific 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 so they continue to deliver value. To avoid neglecting portfolio management, use specific indicators and measures that prompt your demand board to act.

Check your progress:

- We’ve clearly defined triggers for application portfolio review.
- We’re using value measurements to support application curation.
- We’re using Application Portfolio Management to align and curate our applications.

If you answered “yes” to the questions above, proceed to the next step. If not, complete these action steps to complete this step:

1. **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.

2. **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 make sure 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 the decisions regarding your portfolio) in your portfolio governance policies.

3. **Use Application Portfolio Management (APM) to align and curate your apps.**

Deploy 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.
Step 4a: Expand your community of citizen developers

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

Check your progress:

- We’re investing in training our citizen developers.
- We’re promoting standardized development guidelines to our citizen developer community.

If you answered “yes” to the questions above, proceed to the next step. If not, complete these action steps to complete this step:

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

Don’t assume that potential citizen developers understand the Now platform. You need to prepare your citizen developers 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 you can record and make available on demand. These orientations should include what the Now Platform is, what its 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. Communicate these without unnecessary jargon so people outside your IT organization can understand them. This will promote standardization and maintainability so all developers are solving problems in a similar manner.
Step 4b: 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.

Check your progress:

- We’re highlighting our citizen developer program across our organizations.
- We’re supporting a community of practice among our citizen developers.

If you answered “yes” to the questions above, proceed to the next step. If not, complete these action steps to complete this step:

1. Collect and show value stories to inspire potential citizen developers.

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

2. 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.
Terms & 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 development 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.

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
- How should citizen developers enhance their learning?
- What are the benefits of citizen development?
- Managing Application Development on the Now Platform video

Return to workbook checklist
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

- **Strategic**
  - Critical processes
  - Expert insights
  - Common pitfalls and challenges

- **Management**

- **Technical**

- **Tactical**

**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](#).