

Perform ServiceNow upgrades quicker and more effectively

Minimize support issues and maximize the value of your Now Platform

What's in this Success Playbook

Staying current with your Now Platform lets you to take advantage of the new features and functionality designed to help you achieve your business outcomes. Upgrading your software may seem daunting but the good news is that there are steps you can take to simplify the upgrade process today and in the future.

By keeping your ServiceNow environment up to date, you maximize your investment, reduce the risk of encountering issues that have already been resolved, and avoid the issues inherent to running an unsupported release.

This Success Playbook will help you reduce the time it takes to conduct ServiceNow® upgrades by:

- Providing proven processes and tools to conduct your upgrades
- Showing you the best ways to make changes to your baseline system when required

Key takeaways

The most important things to know

- Treat your upgrade like any other project so your time, resources, and costs are visible to everyone.
- Follow the ServiceNow upgrade process to ensure you stay on track.
- Review skipped changes and make decisions about them.
- Create test plans before you upgrade to standardize the process—and shorten testing and upgrade time.
- Future-proof your upgrades—follow the technical best practices and check for vulnerabilities.

The payoff of getting this right

With the right planning in place, you can reduce the time it takes to conduct ServiceNow upgrades and position your organization to take advantage of new ServiceNow capabilities available with each upgrade.

What you need to get started

Leadership support

Enlist one or more certified ServiceNow system administrators.

Prerequisites

You need one or more ServiceNow instances that are not at the current release.

Playbook overview

Follow these stages to ensure a safe and effective upgrade:

Stage 1 – Use project management practices

Stage 2 – Follow the documented upgrade process

Stage 3 – Review skipped changes

Stage 4 – Create and maintain your test plans

Stage 5 – Streamline future upgrades

Stage 1 – Use project management practices

Treating your ServiceNow upgrade like any other formal IT project makes the time, resources, and cost of testing and remediation visible to everyone in the organization.

KEY INSIGHTS

- Ensure you have the right resources at the right time to deliver a higher degree of success.
- Communicate before, during, and after the upgrade to keep everyone in sync.

Approach your ServiceNow upgrades like you would any other important project. This includes applying smart project management concepts like getting sponsorship, identifying stakeholders, assessing risk, making time estimates, allocating resources, communicating, and more. Be sure to use a project management application to track and communicate your progress. The ServiceNow® Project Portfolio Management suite, which is included in ServiceNow IT Business Management, is a great choice, but other tools will also get the job done if you have not purchased IT Business Management.

Set project phases and timelines

Use the upgrade process in [Stage 2](#) to draft your project phases and timelines based on available resources. Modify the project plan as necessary depending on the number of instances, depth of testing, and other factors that may impact your schedule.

TIME ESTIMATE TO COMPLETE

Two to 10 hours depending on how detailed your project plan becomes.

Allocate resources

In the past, we've had customers tell us that they underestimated the resources required to do an effective upgrade. But upon closer inspection, it wasn't that they needed more resources—they needed people in different or additional roles.

When you identify who you need to participate beforehand, you can allocate them to your project plan and avoid surprises (also known as delays) in your project schedule.

EXPERT TIP

Use a change management process. Customers who do deliver higher-quality results. If you don't have one, create one using this [free ITIL guide](#) or use the [Change Management application](#).

CONSIDER INCLUDING

Project manager	To manage the overall project
ServiceNow system administrator	To execute the clone requests, perform the upgrade, and manage the update sets
ServiceNow developer	To remediate any issues found, review skipped logs and to use best practices to avoid future issues. Develop tests using the Automated Test Framework.
Test plan managers or admins	To create and maintain the test plans
Tester(s)	To carry out functional, UAT, and other tests
Governance, risk, and compliance staff	To ensure controls are in place that satisfy regulatory and compliance requirements

Table 1: Suggested upgrade team members

TIME ESTIMATE TO COMPLETE

Two or more hours depending on your resource allocation process.

Communicate

Communication is key to any project, and upgrading is no exception! If you already have processes in place to effectively communicate changes, continue using them. If you don't have a communication process, here are a few suggestions:

- Notify your ServiceNow stakeholders of the upgrade process, schedule, and impact in advance, such as change records and email notifications. Gather their contact information in case you encounter issues.

- Notify your biggest audience—the end users. Let them know what to expect and why they can look forward to the upgrade. Some customers get very creative on how they communicate with their end users. You might want to try:
 - Video “commercials” on a TV monitor in the break room
 - Upgrade parties themed after the upgrade name, like a New York party
 - Signs posted inside the elevator, bathroom stalls, or wherever you have a captive audience!

TIME ESTIMATE TO COMPLETE

An email communication takes about 30 minutes.
More time is required for some of the creative ideas.

Here is an example of an effective email communication:

Great news!

We are upgrading to the Chewbacca ServiceNow release on January 1. Soon you'll have access to these new capabilities:

- Capability #1
- Capability #2
- Capability #3

In addition, these capabilities have been enhanced from previous versions:

- Capability #1
- Capability #2
- Capability #3

We expect all upgrades in all instances to be complete by January 25th.

If you are unable to log in to ServiceNow, call the service desk at 000-555-1212 and our agents will be happy to assist you.

If you can log in, but you are having issues with the platform, navigate to www.issues.com and enter a self-service bug report to describe your issue.

Thank you,
Your ServiceNow team

Figure 1: Sample email communication

Stage 2 – Follow the documented upgrade process

ServiceNow has experienced and documented thousands of customer upgrades. Following this documented upgrade process ensures you don't miss a critical step.

KEY INSIGHTS

- Review the release notes for all releases between your current release and the target release.
- Upgrade the instances farthest from production first.
- Consider using a sandbox instance to do a mock upgrade to reduce risk.

For a smooth upgrade, follow the documented upgrade process on the product documentation page and the release notes for your product upgrade.

The instructions listed here summarize this documented upgrade process.

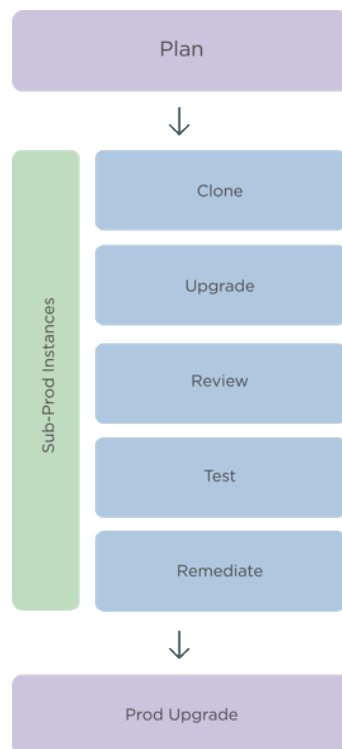


Figure 2: The upgrade process

Step 1: Read the release notes and plan your upgrade

Start by reading the release notes for your target version so you understand the required upgrade and migration tasks.

If multiple software versions are involved, be sure to consult the release notes for each version between your current version and the target version.

Additionally, don't forget to review the release notes for each product you have licensed. There are details on ServiceNow ITSM, ITOM, Security, HR, and more.

Step 2: Prepare for the development instance upgrade

For a better understanding of your production upgrade duration, request a clone of your production instance onto a sub-production instance that has similar hardware and your full database. Confirm your current and target release versions—you'll use this information later when you schedule your upgrade in HI.

Consider designating a sandbox instance to upgrade first. A sandbox instance is a clone of your production instance that's not part of the deployment stack.

If you don't already have a sandbox instance, you can purchase one as part of your company's family of instances. Or you can use a free [personal developer instance](#) from ServiceNow to learn about new release features. Keep in mind, however, that you cannot clone your production instance to a free personal developer instance.

EXPERT TIP

Upgrading a sandbox instance minimizes risk because it lets you test on production data without impacting your development cycle.

Step 3: Verify your upgrade configurations and schedule the

development instance upgrade in HI

Check the configuration of your upgrade's scheduled job to view how often and when it runs. Then, schedule your upgrade in HI.

Step 4: Upgrade and validate the development instance

Track the progress of your upgrades with the Upgrade Monitor. For your first sub-production instance upgrade (your development instance), use the Upgrade Monitor to process the skipped list of records that were not addressed in the upgrade. Then identify your update sets and perform functional testing.

EXPERT TIP

Refer to the product-specific upgrade instructions, like these for [Performance Analytics](#), and this video on [upgrading to a new release](#).

Step 5: If applicable, upgrade and validate your other sub-production instances, such as your test instance

If you have any other sub-production instances in addition to your development instance, such as a test instance, request to upgrade them. Request these upgrades after you have configured and refined your development instance. Immediately after the upgrade is complete, apply the post-upgrade changes made to your development instance, including activating optional plugins, installing and upgrading applications, and applying update sets.

EXPERT TIP

Hold mini training sessions or brown-bag lunches to spotlight new features of the target release, especially if there are UI changes.

Step 6: Prepare to upgrade the production instance

After you've configured and refined your test environment to be a good representation of your production environment, prepare to upgrade your production instance.

Step 7: Upgrade the production instance

After you've upgraded your sub-production development and test instances, upgrade your production instance last. Then validate that the upgrade was complete, apply update sets and fix scripts, and perform post-upgrade user acceptance testing (UAT).

Hold a retrospective after upgrades are complete to identify what worked well and what you can improve.

TIME ESTIMATE TO COMPLETE

The time varies from approximately two to eight weeks depending on the depth of testing, how recently the latest upgrade was completed, the number of overall customizations, and adherence to technical best practices.

Related resources

- [Upgrade planning checklist \(New York\)](#)
- [Cloning a system instance](#)
- [Troubleshooting and other upgrade resources](#)
- [HI: ServiceNow Patching Program FAQs](#)
- [HI: Features impacted during the database upgrade](#)
- [Free personal developer instance](#)

Stage 3 – Review skipped changes

One of the strengths of ServiceNow is that it is so configurable. As such, we do not overwrite your changes when we do an upgrade. That means you need to review what the upgrade skipped and make clear decisions on whether to revert, merge, or continue with those changes. Fortunately, we've made filtering the signal from the noise a pretty easy process.

KEY INSIGHTS

- ServiceNow upgrades will not overwrite changes you have made.
- You don't have to remediate all skipped changes.
- You should at least acknowledge all skipped changes.

One of the features of the ServiceNow upgrade process is that we will not overwrite configuration or customization changes you have made to the system. Changes you make fall mainly into two categories:

- New records you have created such as ACLs, business rules, fields, and so on
- Changes to the out-of-the-box records ServiceNow provides

ServiceNow recognizes that new records you create will not conflict with anything we have created in the past or in the future, so there is no risk in upgrading them.

When you modify out-of-the-box records, the process isn't so straightforward. This may happen in these three cases:

Case 1: You deactivate a record

Within the ServiceNow upgrade process, a record with an active field set to false can still receive updates. The active field will remain false and it silently upgrades the record to the new functionality. This can be an advantage or disadvantage depending on your perspective. If you want that new functionality, you'll never receive any sign that it was updated (except, perhaps, the release notes).

Case 2: You modify other fields in the record

In this case, the upgrade process recognizes you intentionally made more extensive changes to a record and takes the position, "You modified it, you own it." If the upgrade were to overwrite those changes you made, it could adversely impact your process, so an entry is made in the upgrade log that the record was skipped. (See the Review skipped records section.)

Case 3: You modify as part of your configuration process

This is very similar to the second case, but the changes were not made overtly. You may have simply made a change to a field label or form layout. The upgrade process is still going to flag these changes as “yours” and will not overwrite them. However, they have a lower priority associated with them, which is helpful when you review your skipped records.

Review skipped records

As part of your process, review the records that were skipped during the upgrade process and decide whether to keep the changes you made or to accept the records the upgrade process attempted. Fortunately, there is a way to filter all the changes to focus on what’s important using the **Skipped Changes** to Review related list on the **Upgrade History** record.

Review skipped changes (details)

These steps show the process of reviewing the skipped records.

1. Navigate to **System Diagnostics > Upgrade History**.

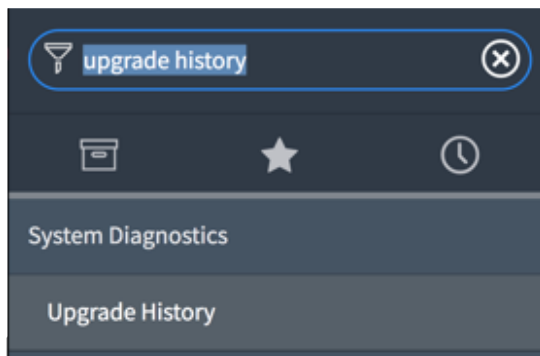


Figure 3: Upgrade History menu

2. Locate and open the record related to your recent upgrade from the list. It should look similar to this:

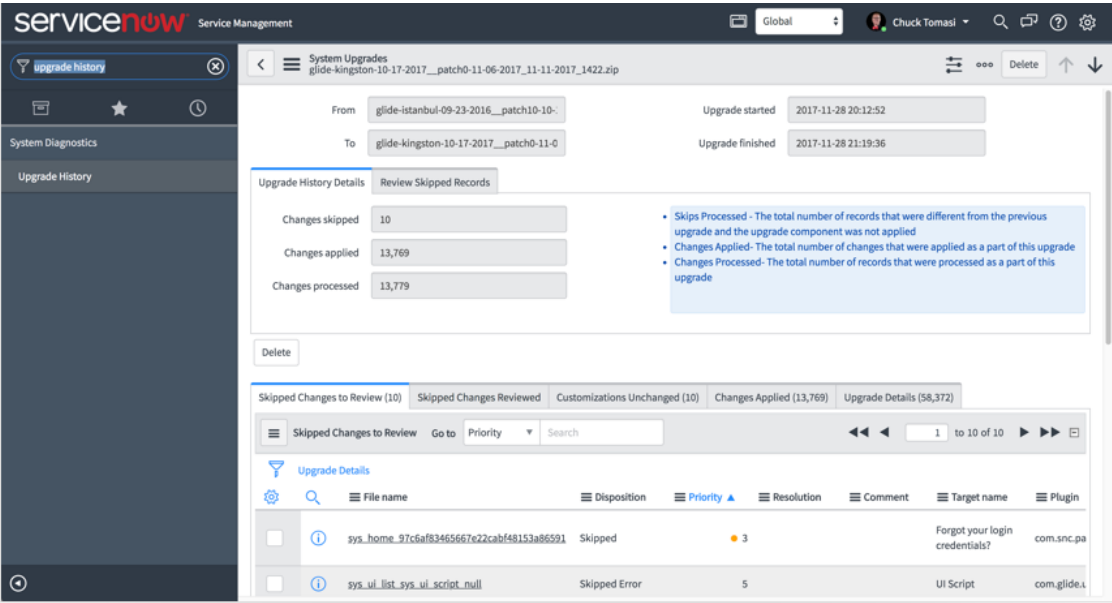


Figure 4: System upgrade page

3. Scroll to the tabs showing the related lists at the bottom of the form.
4. Locate the list titled **Skipped Changes to Review**.
5. Right-click a record with **Priority 5** and select **Filter Out**.

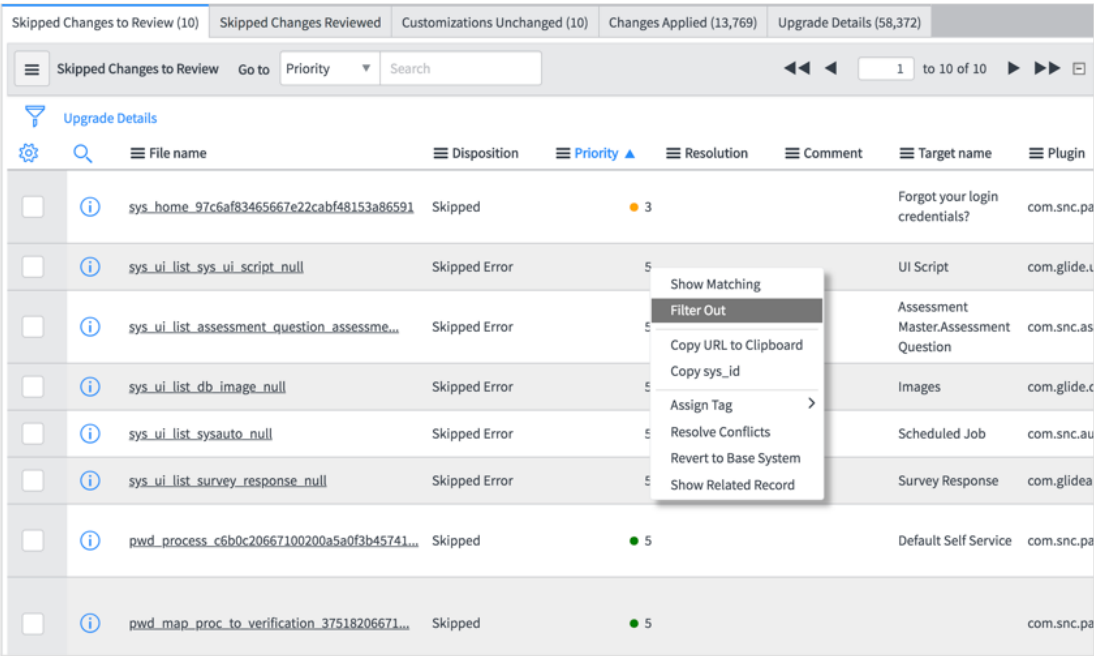


Figure 5: Filtering skipped changes

By filtering out certain records, you eliminate the need to review many low-risk changes like field labels or form layouts. Your list of skipped upgrades is now much smaller and more manageable.

Continue to process the skipped records by opening each one and making the decision to revert, resolve, etc.

TIME ESTIMATE TO COMPLETE

This varies from customer to customer depending on how many changes were made to the system. It could take minutes or possibly days.

Related resources

- [Success Quick Answer: What's the process to review and address skipped changes?](#)
- [Skipped Changes to Review related list](#)
- [Upgrade Skipped Records – ServiceNow eLearning series](#)

Stage 4 – Create and maintain your test plans

When you've created your test plans before your upgrade—preferably during development or implementation—it will help standardize the process, reduce your testing time, and shorten the overall upgrade project. Using tools like Test Management and Automated Test Framework will help.

KEY INSIGHTS

- Create test plans during development so they stay current.
- Use automated test tools where applicable to greatly reduce testing time.
- Avoid moving update between different instances on different releases.

You know that you need to do some level of testing before deploying new software, but you might not have a plan. The best way to start is asking yourself “What do I need to test, and how do I keep track of it?”

You only need to test what you've previously deployed. If your organization is using ITSM only, there's no point in testing ITOM applications like Discovery, Service Mapping, and Event Management. The best practice is to create the test plan(s) as you configure and customize your applications, but if you deployed in a hurry, you may not have test plans. If that's you, then make sure to create and test your test plans before you upgrade.

You can build and maintain test plans in a variety of ways from paper to Excel to purpose-built applications. ServiceNow offers two solutions to help you with your testing—one for manual testing and one for automated testing. Automated Test Framework (ATF) is provided at no additional cost. Test Management is available with a subscription ITSA Unlimited, Service Strategy Worker/Planner, and ITBM Worker/Planner.

Manual testing with ServiceNow Test Management

With manual testing, you give someone a list of instructions to follow with expected results, and they document whether they got the expected results. Pretty simple, right? The ServiceNow [Test Management 1.0](#) application (shown in Figure 6) makes this even easier. If your instance doesn't include the Test Management menu, our product document on [activating a plugin](#) can help. The name of the plugin to activate from the list is Test Management (com.snc.test_mgmt).

The ServiceNow Test Management application provides a tool for manual software testing. Test managers can use this application to manage all phases of the testing process. They can:

- Create and maintain the test repository by creating test suites, test cases, and tests
- Enable test execution by creating test plans, adding test cases, and assigning testers to test cases
- Initiate the testing process and monitor its progress
- Evaluate test results and complete the test plan sign-off form
- View testing reports on the Test Management dashboard

Testers can use this application to:

- Perform tests and record test results
- Update test case status
- Report defects and retest as necessary

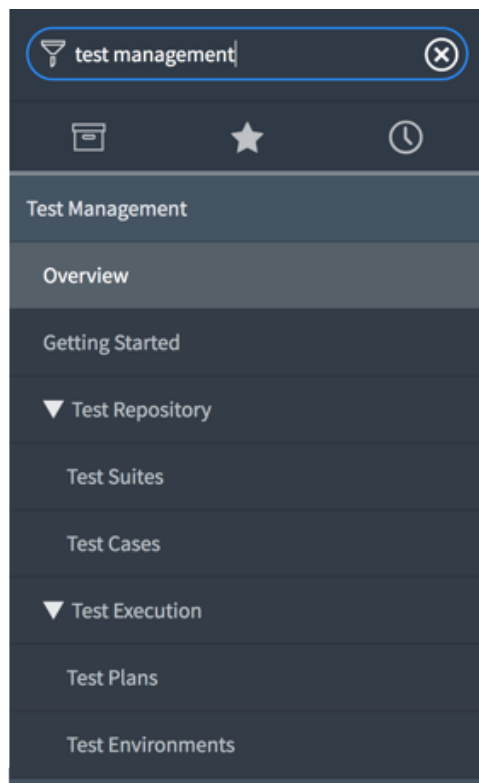


Figure 6: Test Management menu

EXPERT TIP

Customers with great test plans create clear instructions for their tests. Assume your testers know nothing about the software or process they are testing.

Each of your tests should include:

- Clear, step-by-step instructions to perform the test
- Expected results
- Input for actual results
- Input for any error messages
- Input for screen shots

TIME ESTIMATE TO COMPLETE

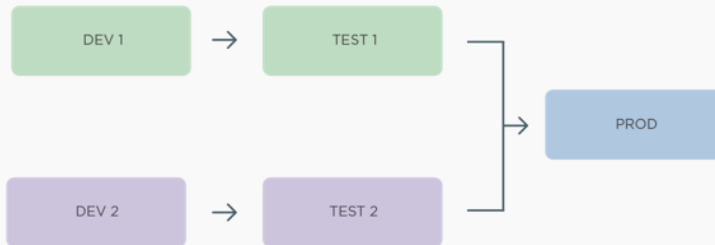
Eight or more hours depending on which ServiceNow applications you have configured and deployed and to what depth you test your applications.

During test remediation, do not move update sets between instances with different versions. For example, if you've only upgraded dev but not test, don't deploy from dev to test until after you upgrade test.

If you have a sandbox instance, consider cloning your production instance back to sandbox after you upgrade your dev instance to the target release and before you upgrade production. You can use this as a dev instance for emergency changes to production on the legacy release while your upgrade to the target release continues.

You may want to spend a little more to purchase additional instances that allow development to continue while upgrades and testing are underway (see Figure 7). This allows development to continue on the legacy release while the upgrade is in process and avoids the conflict of publishing from a newer release back to an older release before the upgrade is complete.

DUAL STACK UPGRADE PROCESS



- ① Upgrade **DEV1** to target release and validate.
- ② Upgrade **TEST1** to target release and validate. During **DEV1** and **TEST1** upgrade, development on legacy release continues on **DEV2/TEST2/PROD**.
- ③ Publish any remaining legacy changes from **DEV2** to **TEST2** to **PROD**.
- ④ Upgrade **PROD**.
- ⑤ Publish any remediation from **DEV1/TEST1** to **PROD**.
- ⑥ Upgrade **DEV2**.
- ⑦ Upgrade **TEST2**.

All instances are now at the target release.

Figure 7: Dual Stack to enable continuous development during upgrades

TIME ESTIMATE TO COMPLETE

It takes 10 minutes to activate the Test Management plugin.

Automated tests with ATF

With the [Automated Test Framework \(ATF\)](#), you create and run automated functional tests on your ServiceNow instance. When you upgrade or modify an instance, simply run your tests to confirm that the instance still works as designed.

The beauty of ATF is that it is automated. With a single click, you run an entire suite of tests and, in moments, analyze the results to determine if remediation is required. You can use the same tests again for your next release or upgrade if you keep the tests in sync with the configurations and customizations they test.

EXPERT TIP

Always remember to update your test plans when you make configuration and customizations to your applications!

ATF is functional testing that allows actions such as creating records, setting field values, checking results of field values, and so on. If a test fails, it simply indicates that the result doesn't meet the test you provided, not where the underlying problem may be—this is very similar to what a human would do with instructions, only much faster.

Search for Automated Test Framework in the application menu to find ATF.

"As you test, prioritize your issue list and review it frequently to ensure your developers are working on the right things."

– Gabriela Pospis, Sr. QA IT Manager

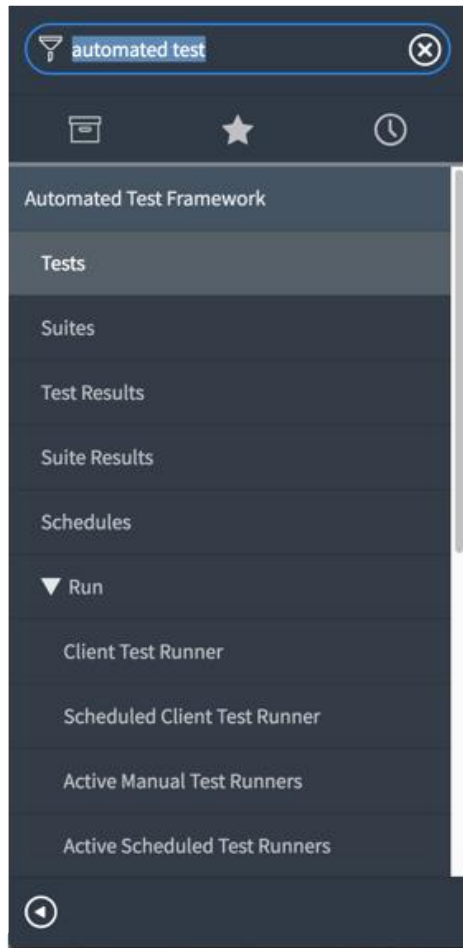


Figure 8: Automated Test Framework menu

TIME ESTIMATE TO COMPLETE

Eight or more hours depending on which ServiceNow applications you have configured and deployed and to what depth you test your applications.

Related resources

- [Automated Test Framework overview](#)
- [Success Quick Answer: When and how should I use Automated Test Framework \(ATF\)?](#)
- [Success Quick Answer: What best practices should I consider when creating ATF tests?](#)
- [Test Management](#)
- [ATF Fundamentals eLearning series](#)

Stage 5 – Streamline future upgrades

You can future-proof your upgrades when you understand the cost of customizations. By following the technical best practices and checking for vulnerabilities before your upgrade, you can reduce the number of issues to remediate.

KEY INSIGHTS

- Avoid customization to reduce time and costs with future upgrades.
- Use the published technical best practices to avoid issues with upgrades, performance, scalability, and user experience.
- Consider purchasing a configuration review to audit against bad practices to reduce upgrade risks.

Assess and understand the true cost of customizations

“If only I knew then what I know now” is a fairly common customer refrain when it comes to customization. In the effort to fulfill requirements or close out stories, developers aren’t thinking of the true cost of a feature—the cost of implementation plus the cost of maintenance, which includes testing and validation during upgrades.

How do you determine the true cost of a customization? Rely on experienced ServiceNow implementers to vet out which requirements involve customizations and which can be done with simple configurations.

“We drive a culture of ‘out-of-the-box’ throughout the organization and place a high cost on customizations.”

– Robyn C. Sah, Blue Cross Blue Shield of Louisiana

In general, customizations require scripting, take longer to implement, and often involve more in-depth testing. The following table provides some examples of customizations versus configurations based on customer feedback.

Note: Your definitions may vary on some items.

CONFIGURATION	CUSTOMIZATION
Add a new field to a form	Modify the service catalog UI page
Create a no-code business rule	Create an advanced business rule
Change the value of a system property	Create a new workflow
Place Service Portal widgets on a page	Create new Service Portal widgets
Add a reference qualifier using the condition builder	Add an advanced reference qualifier using script

Table 1: Configuration and customization

Keep your instance as close to the baseline, or out-of-the-box, configuration as possible. When you use configuration changes, you keep implementation, testing, and upgrade costs down.

TIME ESTIMATE TO COMPLETE

There is no fixed time for this step. It is a cultural methodology you adopt in to your development process.

Observe technical best practices while developing

As some have said, "The great thing about ServiceNow is that you can do anything, and the bad thing about ServiceNow is that you can do anything!" We recognize that customers may have unique requirements and may need to extend an existing application or even create a whole new application to address a business process. That's great—it's part of the reason the Now Platform is architected the way it is.

So, make customizations if you must—but make them safely. As with many software platforms, there is often more than one way to solve any given problem using ServiceNow. Over the years, we have collected and documented these technical best practices to help protect you against common (and some not so common) issues.

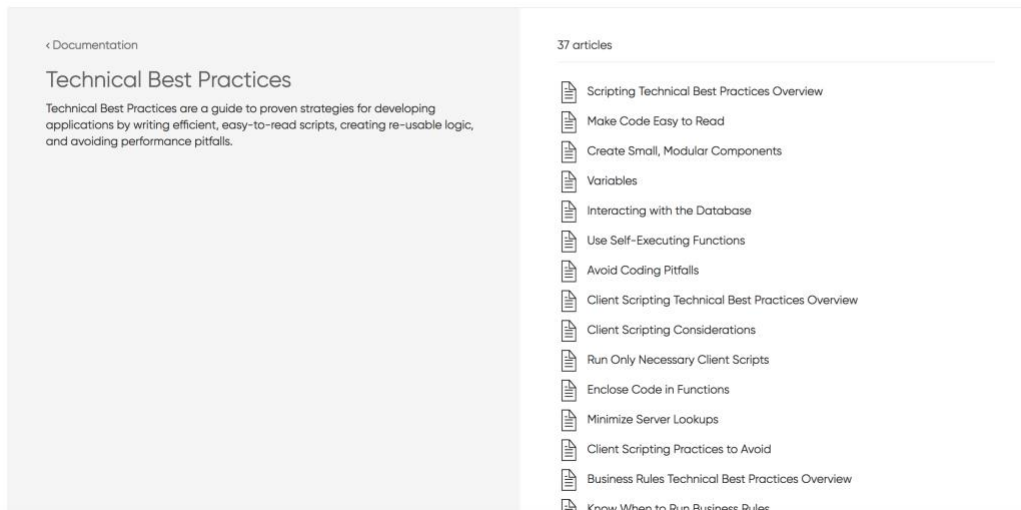


Figure 9: Technical Best Practices page

Learning and practicing the methods documented in the technical best practices can reduce the risk of performance, scalability, manageability, and even usability issues over time. For example, Automated Test Framework may indicate that a particular business rule is functioning properly by responding to the proper trigger and producing the proper result, but when you get it in to UAT and a tester discovers it takes six seconds to run (an eternity if the person can't do anything else in their browser), you may have some bad practices at work. This is where the technical best practices can help future-proof your implementations.

TIME ESTIMATE TO COMPLETE

It takes 10-12 hours to review the technical best practices.
Using them requires no additional overhead.

Conduct a configuration review

Something else that may be extremely helpful in spotting potential issues is a [ServiceNow Configuration Review](#) (aka, Config Review). This is a fixed-fee service available from ServiceNow.

The Config Review is performed by a ServiceNow certified professional services consultant and addresses challenges around upgradability, manageability, scalability, and performance during or after the implementation of ServiceNow applications and the Now Platform. The Config Review provides recommendations to align your configurations and customizations with ServiceNow best practices.

The result of the Config Review is a PDF report with an overall score for each category. The report also details findings, corrective actions, priorities, and links to the specific records on your instance to help you quickly remediate any potential issues.

Work with your ServiceNow account team to schedule a Config Review. For best results, conduct a Config Review before your next upgrade to get in front of any potential issues. The money spent could save you countless hours of testing and remediation!

TIME ESTIMATE TO COMPLETE

Up to 40 hours to conduct the review.

Remediation depends on how many findings were reported.

The takeaway

When you treat your ServiceNow upgrade like a project, you have greater visibility, buy-in, and success. Make sure to use our proven upgrade process as a framework to avoid issues.

Always review skipped changes so you can gain visibility to customizations that can be reverted to the ServiceNow baseline if necessary. And don't forget that using the latest techniques to review skipped changes dramatically shortens your upgrade project time.

Build and maintain test plans so you can guarantee a high degree of success when you promote changes. Follow the ServiceNow technical best practices to future-proof your work from upgrade issues down the road.

And remember that, while the Now Platform is highly customizable, any customization comes at the cost of higher maintenance. Keep your instance as close to the out-of-the-box configuration as possible, and use configuration change to reduce implementation, testing, and upgrade costs.

Related resources

- [Success Playbook: Avoid customization pitfalls so you can innovate and meet demand at scale](#)
- [Success Quick Answer: How does Healthscan help me manage my instance health?](#)
- [Success Checklist: Plan for upgrades at least once a year](#)