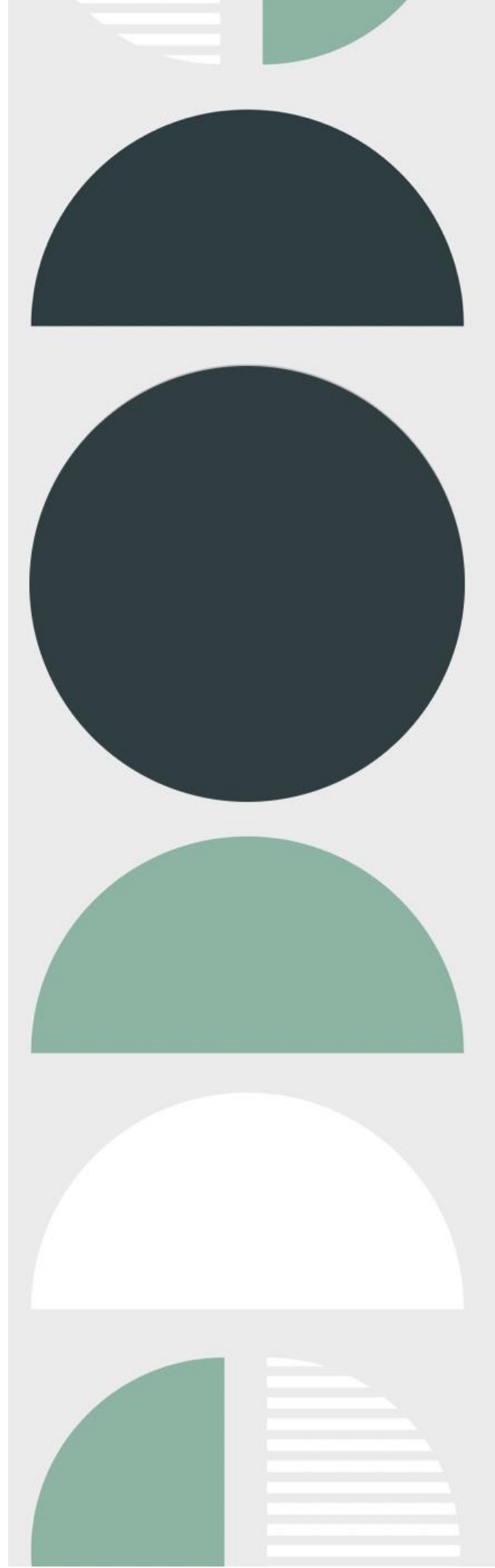


Perform ServiceNow upgrades efficiently and more effectively

Developed by the ServiceNow Best Practice
Center of Excellence

February 2022



Why is it important to upgrade your instance/s?

Staying current with the Now Platform® lets you take advantage of the new features and functionality designed to help you achieve your business outcomes. 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 when running on an unsupported release.

Follow the steps outlined in this workbook to perform upgrades quickly and effectively.

Why should I invest in upgrades?

Regular upgrades bring value to your organization, including:

- Stay current with the latest ServiceNow features and functionality.
- Keep your instance healthy.
- Remain covered by ServiceNow technical support.

What will this workbook help me do?

This workbook will prepare you to:

- Plan for an upgrade.
- Create a repeatable process for future upgrades.
- Create test plans BEFORE you upgrade to implement standardize testing.
- Minimize the time and resources needed to upgrade your instance.
- Avoid potential issues that could occur during an upgrade.

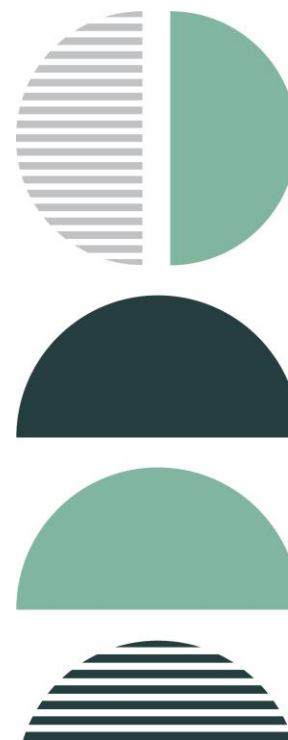
How to use this Success Workbook

This Success Workbook will guide you through the steps to plan and execute a successful ServiceNow upgrade as well as how to create a repeatable process to minimize time and resources needed which will streamline future upgrades.

Start by reading through the [Workbook Checklist](#) – review the steps and tasks that need to be completed to get started with upgrades.

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: Perform ServiceNow upgrades quicker and more effectively

Step 1: Use project management practices

- Set project phases and timelines.
- Allocate the right resources to ensure success.
- Implement a project communication plan.

Step 2: Follow the documented upgrade process

- Read release notes and plan your upgrade.
- Prepare for the development instance upgrade.
- Verify your upgrade configuration, and schedule the development instance upgrade.
- Upgrade and validate the development instance.
- If applicable, upgrade and validate other sub-production instances, such as test or QA.
- Upgrade the production instance.

Step 3: Review skipped changes

- Read insights on why records could be skipped.
- Review skipped records.

Step 4: Create and maintain your test plans

- Run manual testing with ServiceNow Test Management.
- Run automated tests with ServiceNow ATF.

Step 5: Streamline future upgrades

- Assess and understand the true cost of customizations.
- Observe technical best practices while developing.
- Conduct a configuration review.



Step 1: Use Project Management Practices

Approach your ServiceNow upgrades as you would any other important project by applying smart project management concepts including securing sponsorship, identifying stakeholders, assessing risks, making time estimates, and allocating resources. 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.

Check your progress:

- Do you have the right resources with the appropriate level of commitment to complete their tasks on time?
- Do you have plans for communication, marketing, and training for the duration of the project?

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

1. Set Project phases and timelines

Use the upgrade process in [Step 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.

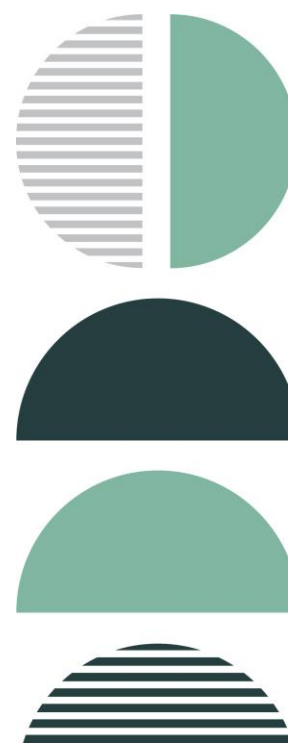
The estimated time to complete this is two to 10 hours, depending on how detailed you choose to make your project plan.

2. Allocate the right resources to ensure success

We’ve had customers tell us that they underestimated the resources required to do an effective upgrade but, upon investigation, we found that they didn’t need more resources—they needed people in different or additional roles.



Practitioner insight: If you are on Paris release or later, you can take advantage of the Upgrade Preview to see how many skipped changes you will have in production. You can use this information to help plan and scope your upgrade. Plus, you can resolve some of those skipped changes in the preview and have fewer to resolve later. You can learn more on the [Upgrade Center page](#).



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.

The estimated time to complete allocation is two or more hours depending on your resource allocation process.



Practitioner insight: Customers who use a change management process deliver higher quality results. If you don't have one, create one using 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 administrators	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

*These roles could be the same person depending on your staffing model.

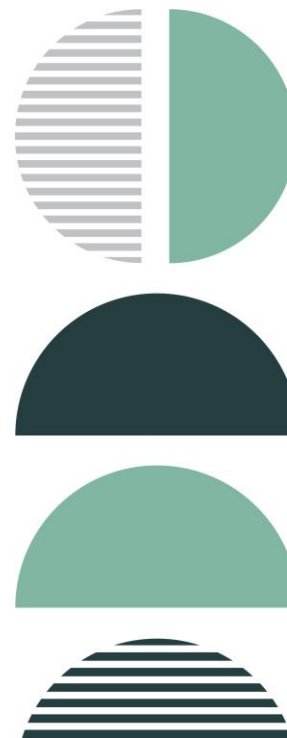
3. Implement a project communication plan

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 Quebec party
 - Signs posted inside the elevator, bathroom stalls, or wherever you have a captive audience

The estimated time to complete these tasks will vary depending on the medium. An email will take an average of 30 minutes, but most other options will take longer.



Here is an example of an upgrade email:

Hi!

We are upgrading ServiceNow on January 1 to offer new features and functionality that will add:

- New Feature
- New functionality #1
- New functionality #2

In addition, the following functionality will be enhanced:

- Functionality #1
- Functionality #2
- Functionality #3

[Click here](#) for more information about these features and how they will make the world of work work better for you!

[Click here](#) to sign up for training.

If you experience any issues after the upgrade, please contact the ServiceDesk at 000-111-2222 or visit the portal and chat with our virtual assistant.

We look forward to providing you an even better ServiceNow experience.

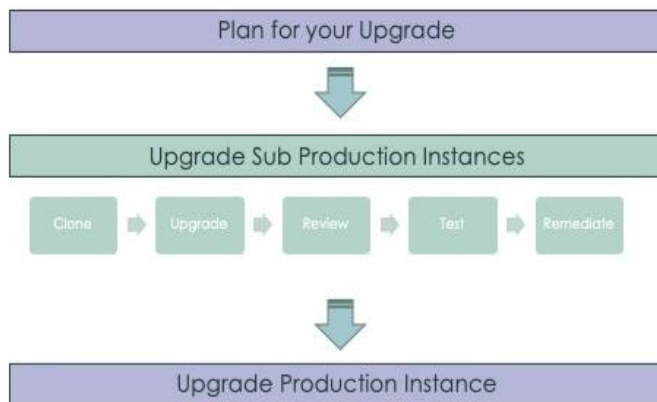
Your ServiceNow Team



Step 2: Follow the documented upgrade process

ServiceNow has conducted and documented thousands of customer upgrades. Follow the documented upgrade process on the [product documentation page](#) and the release notes for your product upgrade.

The instructions listed here summarize the documented upgrade process.



Check your progress:

- Have you thoroughly reviewed the release notes for all releases between your current release and the target release?
- Have you made a decision about using your sandbox instance or a demo instance to do a mock upgrade to reduce risk? If yes, have you created a plan for this?
- Have you upgraded the instances farthest away from production first, such as Sandbox or Test?

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

Consult the release notes for each version between your current version and the target version if more than one release has occurred since you last upgraded.

Additionally, review the [release notes](#) for each product you have licensed. There are details on ServiceNow ITSM, ITOM, Security, HR, and more. You can visit the Customer Success [Upgrade Center](#) and the [Community](#) for more information.

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.

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

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.



Practitioner insight: Upgrading a sandbox instance minimizes risk because it lets you test with production data without impacting your development cycle.

3. Verify your upgrade configurations and schedule the development instance upgrade.

Check the configuration of your upgrade's scheduled job to view when it runs and how often. Then, schedule your upgrade in Now Support (HI) if you are in a pre-Paris release. If you are on Paris or later, you can use the Upgrade Center in your instance to schedule your upgrade.

4. Upgrade and validate the development instance.

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



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



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

If you have other sub-production instances besides your development instance, such as a test instance, request to upgrade them 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.



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

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 based on what you have learned from upgrading the sub-production environments.

7. Upgrade the production instance.

Upgrade your production instance last. Validate that the upgrade was complete, apply update sets and fix scripts, and perform post-upgrade user acceptance testing (UAT).

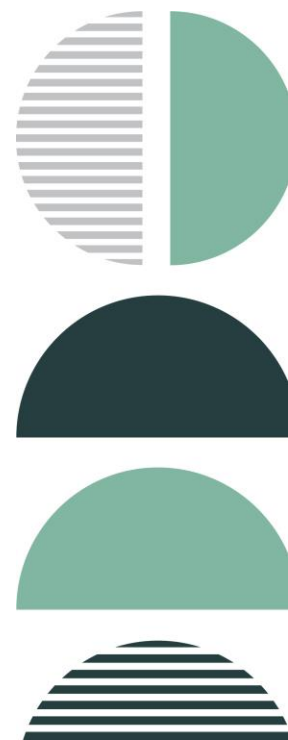
Hold a retrospective after upgrades are completed to identify what worked well and what you can improve.

The estimated time to complete all these steps can vary from two to eight weeks depending on the depth of testing, how recently you last upgraded, the number of customizations, and how well you adhere to development best practices. In addition, your organization may have testing and/or auditing requirements that also add time to the schedule.

Here some related resources you might want to consult:

- [Upgrade planning checklist](#)
- [Cloning a system instance](#)
- [NowSupport: ServiceNow Patching Program FA](#)
- [NowSupport: Features impacted during the database upgrade](#)
- [Free personal developer instance](#)

[Return to workbook checklist](#)



Step 3: Review skipped changes

One of the strengths of ServiceNow is that it is very configurable. ServiceNow does 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, filtering out the noise is a pretty easy process.

How long does this take to complete? That varies from customer to customer depending on how many changes were made to the system. It could take a few minutes or several days.

Check your progress:

- Have you reviewed all skipped changes?
- Have you addressed all the Priority 1 and 2 items?
- Have you decided whether to acknowledge or remediate all changes?

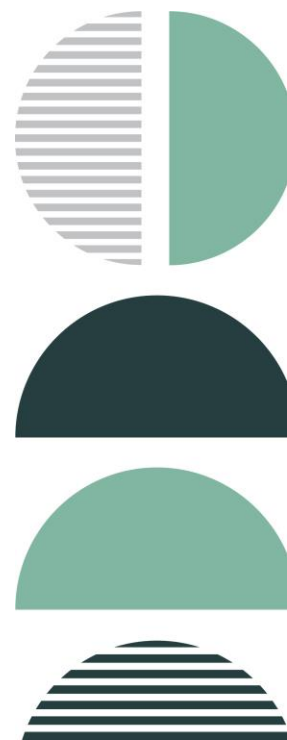
If you answered “yes” to the questions above, proceed to the [next step](#). If not, complete these actions:

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, etc.
- 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.

However, the process isn't as straightforward when you modify out-of-the-box records. ServiceNow skips modified out-of-the-box records so you can review them individually. You need to determine if you can accept the upgrade as is or if some code needs to be adjusted so your customization can work with the upgrade and the change is not accepted.



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.

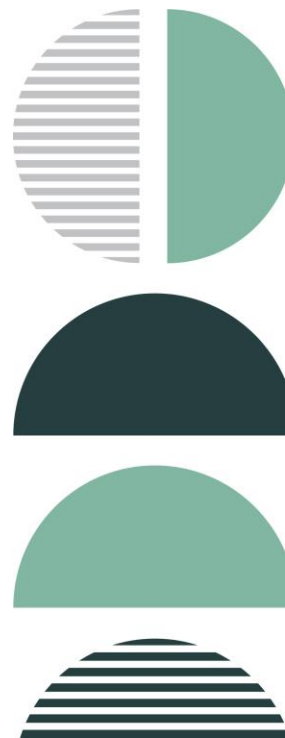
In this case, the upgrade process recognizes you intentionally made more extensive changes to a record and takes the position, "You modified it, you own it." If the upgrade were to overwrite those changes you made, it could adversely impact your process, so an entry is made in the upgrade log that the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

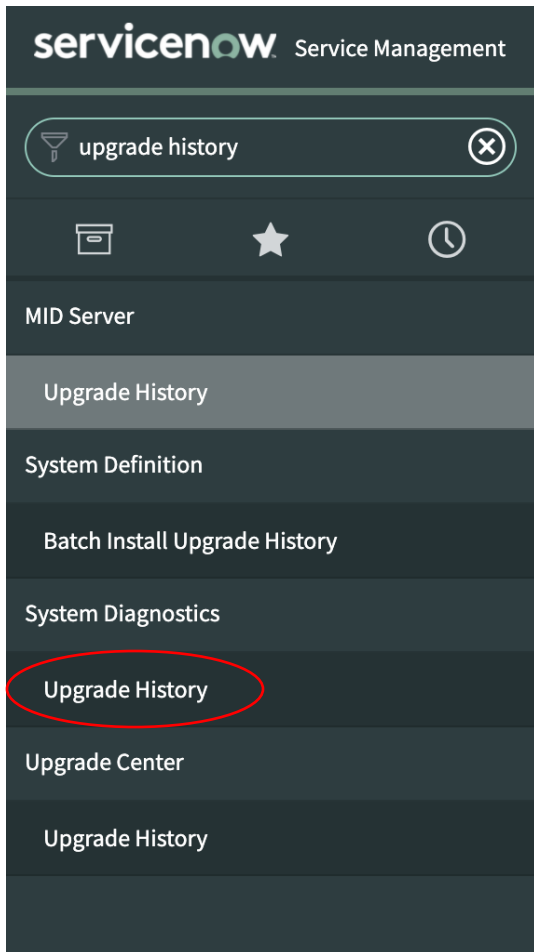
1. Review skipped records.

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.

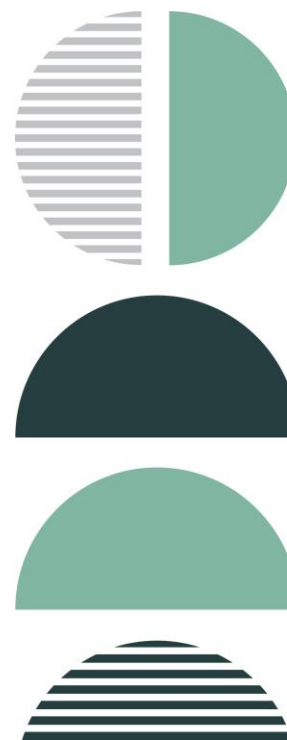
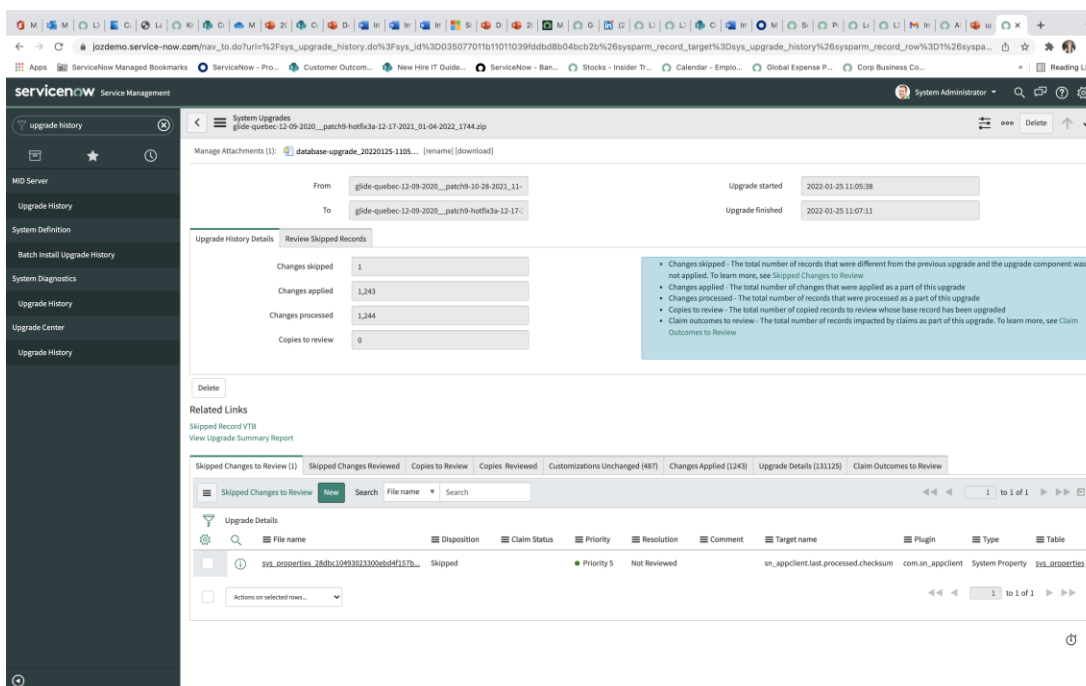


These steps show the process of reviewing the skipped records.

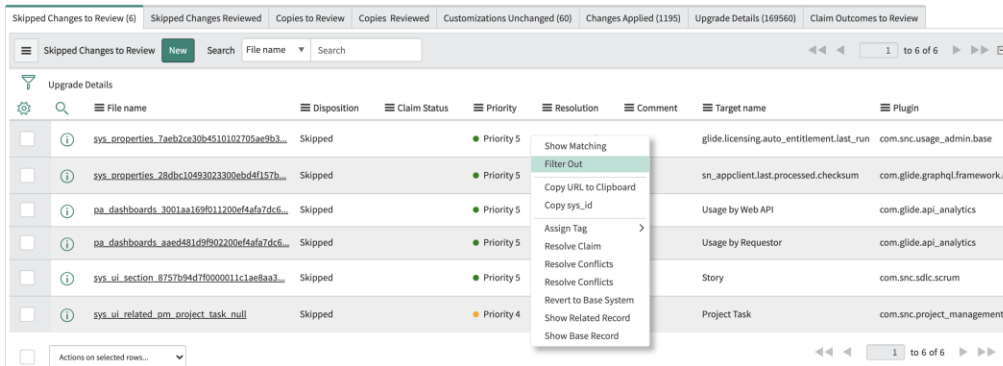
a) Navigate to **Upgrade Center > Upgrade History**.



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

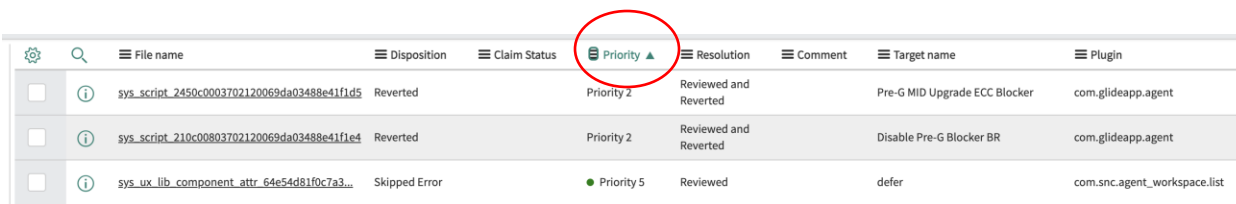


- c) Scroll to the tabs showing the related lists at the bottom of the form.
- d) Locate the list titled **Skipped Changes to Review**.
- e) Right-click a record with Priority 5, and select **Filter Out**.



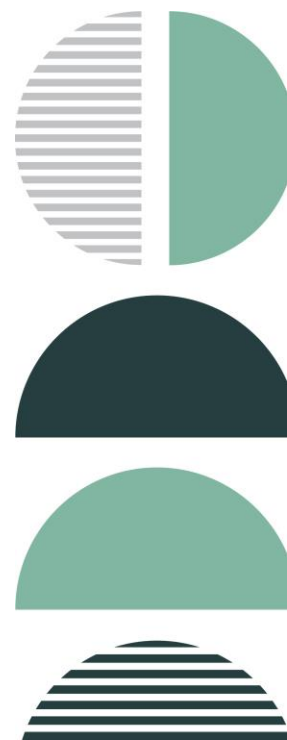
By filtering out low priority records, you eliminate the need to review many low-risk changes like field labels or form layouts. Your list of skipped upgrades is much smaller and more manageable.

- f) Click on the **Priority** header, then click the triangle next to the header to filter in ascending order, i.e. from 1 to 4, so changes appear in order of priority from high to low.
- g) Process the skipped records by opening each one and making the decision to revert, resolve, etc.



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](#)



Step 4: Create and maintain your test plans

You need a plan for how you'll do testing before deploying any new software. The best way to start is by asking yourself, "What do I need to test, and how do I keep track of testing?"

Only test applications and functionality you've deployed. The best practice is to create the test plan(s) as you configure and customize your applications but you may not have test plans if you deployed in a hurry. If that's your situation, 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 testing—one for manual testing and one for automated testing. We recommend that you do manual and automated testing if possible. Manual testing will find issues that automated testing will not, such as poor user experience.

Test Management is used for manual testing, and it is available with a subscription.

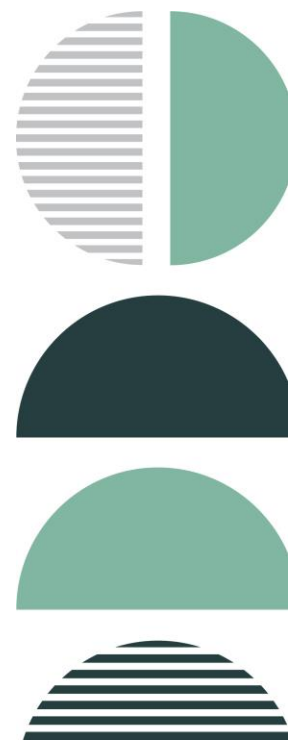
Automated Test Framework (ATF) is an automated testing tool, and it is provided at no additional cost.

Plan for eight or more hours of testing depending on which ServiceNow applications you have deployed and to what depth you have configured them. The more configuration or customization, the more testing time you should allocate.

Check your progress:

- Have you created test plans?
- Have you set up automated testing tools to reduce testing time?
- Do you have strict governance in place to manage or stop development while you are testing?

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



1. Manual testing with ServiceNow TestManagement

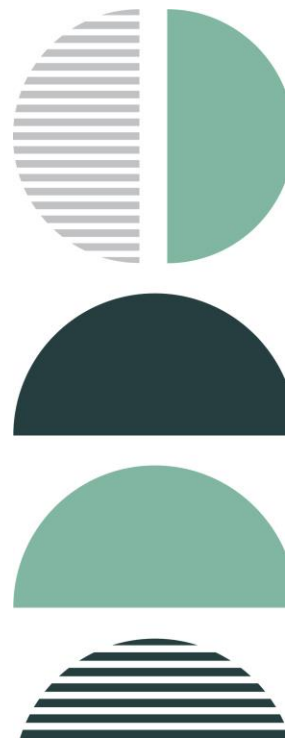
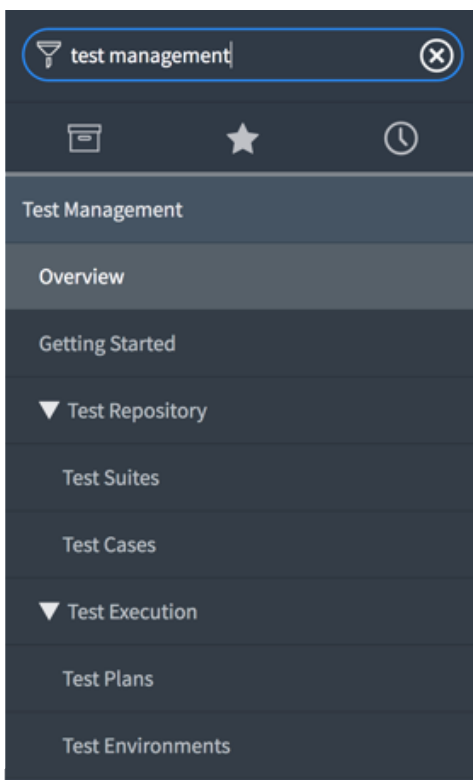
With manual testing, you give someone a list of instructions to follow with expected results, and they document whether or not they got the expected results. The ServiceNow [Test Management](#) application (shown in Figure 6) makes this even easier. If your instance doesn't include the Test Management menu, contact your account rep for more information.

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 re-test as necessary.





Practitioner insight: Customers with good test plans create very detailed and clear instructions for their tests. When writing test plans, assume your testers know nothing about the software or process they are testing.

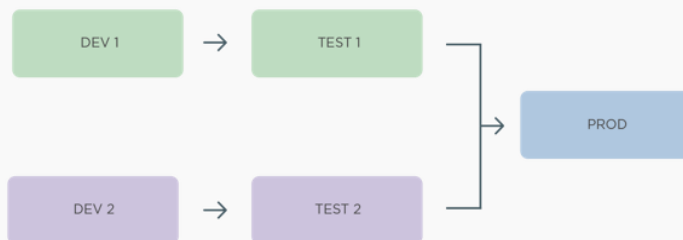
Each of your tests should include:

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

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

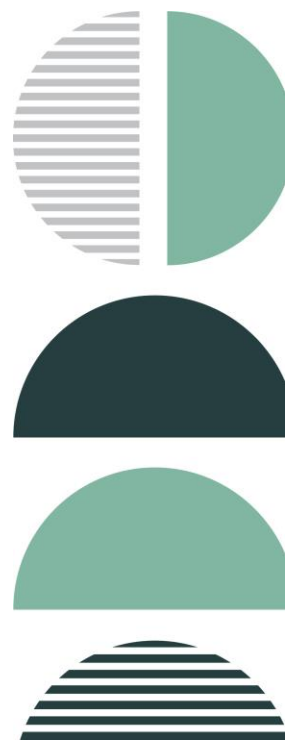
If you have a sandbox instance, consider cloning your production instance back to sandbox after you upgrade your dev instance to the target release and before you upgrade production. You can use this as a dev instance for emergency changes to production on the legacy release while your upgrade to the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

DUAL STACK UPGRADE PROCESS



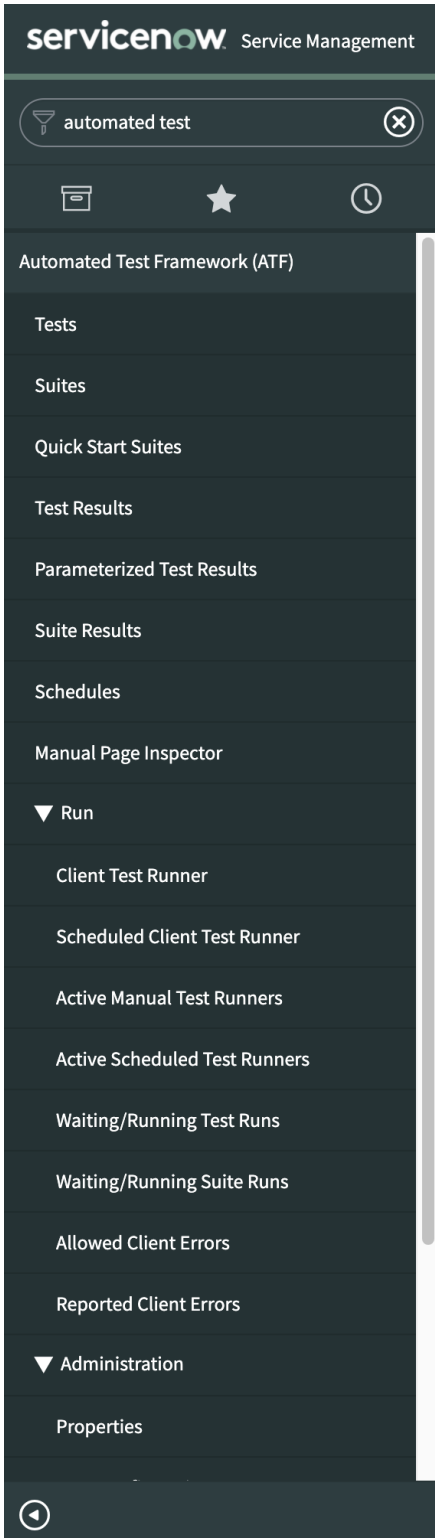
- ① Upgrade **DEV1** to target release and validate.
- ② Upgrade **TEST1** to target release and validate. During **DEV1** and **TEST1** upgrade, development on legacy release continues on **DEV2/TEST2/PROD**.
- ③ Publish any remaining legacy changes from **DEV2** to **TEST2** to **PROD**.
- ④ Upgrade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

All instances are now at the target release.





Practitioner insight: Always remember to update your test plans when you make configurations and customizations to your applications!

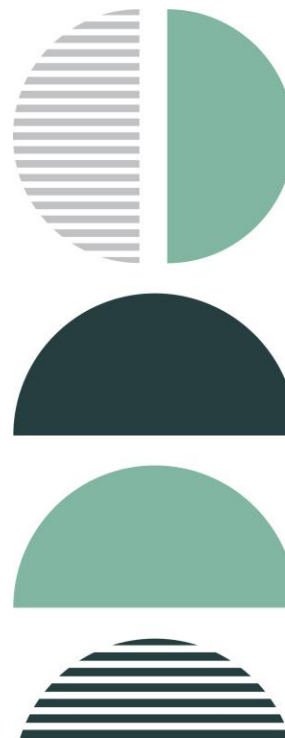


2. Automated test with ATF

Automated Test Framework (ATF), allows you to create and run automated functional tests, such as creating records, setting field values, checking results of field values, and so on. If a test fails, it indicates that the result doesn't meet the test you provided, but not what the underlying problem may be.

When you upgrade or modify an instance, run your tests to confirm that the instance still works as designed. You can re-use the same tests for subsequent upgrades as long as you keep the tests in sync with the configurations and customizations they test.

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



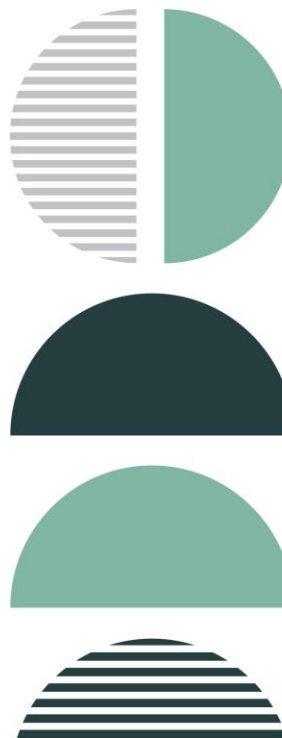


"As you test, prioritize your issue list and review it frequently to ensure your developers are working on the right things."

Gabriela Pospis, Sr. QA IT Manager

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.](#)



Step 5: Streamline future upgrades

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

*Customization is defined as: modifying the out of the box behavior of ServiceNow with code.

Check your progress:

- Do you have a governance process to ensure that customizations are only done for true business needs?
- Do you have a process to document customizations when they occur so you understand how to manage them during upgrades?

If you answered “yes” to the questions above, you have completed the steps outlined in the workbook. If not, complete these actions to complete this step:

1. 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 always 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.

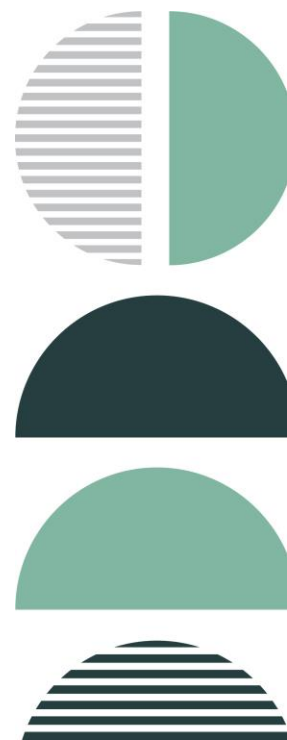
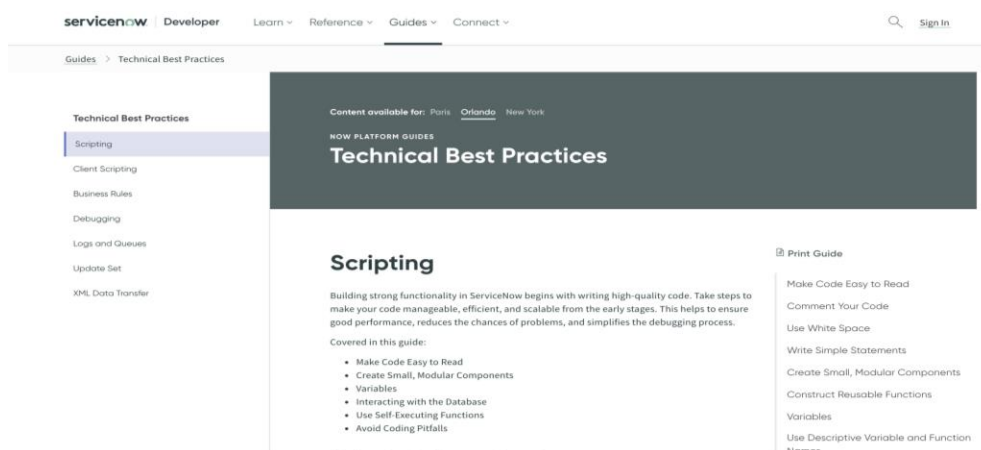
Note: Your definitions may vary on some items.

CONFIGURATION	CUSTOMIZATION
Add a new field to a form	Modify the service catalog UI page
Add a new client script to set the value of a field	Modify an OOTB business rule
Add a new business rule to run a calculation	Modify OOTB dictionary
Place Service Portal widgets on a page	Create new Service Portal widgets
Add a system web service to allow system integration	Client side direct DOM manipulation

2. Observe technical best practices while developing

We recognize that customers may have unique requirements and may need to do customizations to address a business process. That's great—it's part of the reason the Now Platform is architected the way it is.

The key is to make customizations prudently after examining all options. 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](#) on the Developer site to help protect you against common (and some not so common) issues.



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 expected result. However, when you are in 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 inefficient code. This is where the technical best practices can help future-proof your implementations.

3. Engage ServiceNow to conduct a Configuration Review

A [ServiceNow Configuration Review](#) (Config Review) can help find potential issues. This is a fixed-fee service available from ServiceNow.

The Config Review is performed by a ServiceNow certified customer outcomes consultant and addresses potential challenges to upgradability, manageability, scalability, and performance. It can be conducted during or after implementing ServiceNow. 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.

[Return to workbook checklist](#)

