

What are the best practices for customization?

Questions addressed:

Page 1:

- What criteria should I use to determine whether to customize?

Page 2:

- If I do decide to customize, what technical best practices should I follow?
- What best practices should I follow for custom integration requirements?

Page 3:

- How do I deal with customizations when I upgrade?

What criteria should I use to determine whether to customize?

Excess customization can build up technical debt and lengthen your upgrade cycle, inhibiting your ability to take advantage of new features. [Evaluate demands](#) for customization very carefully and only resort to customization where there is clear business value and no alternative to satisfying demand. Your demand board can use a simple, three-step scoring framework to help your evaluation:

Step 1: Identify a “value score” for the proposed customization.

Use a simple scoring framework to assess the business value associated with the request. Your demand board should review and approve this scoring framework. You can use the example below to get started.

Value score	Description
5 – Mandatory	Customization is required for regulatory and compliance purposes.
4 – Critical	Customization is a "must have" to realize a business value objective and/or adoption requirements.
3 – Medium	Customization supports realization of a business value objective and/or adoption, but workarounds are available.
2 – Low	Customization supports service experience for service consumers, process users, and/or developers but does not necessarily promote a business value objective or adoption.
1 – No value	Customization does not support improved service experience, value realization, or adoption.

***Note:** Tables are derived from our Success Playbook on [avoiding customization pitfalls](#).

Step 2: Identify a “complexity score” for the proposed customization.

Evaluate the type of customization proposed based on the potential risk it could have on performance and/or your time to upgrade. You can use the example below to get started.

Customization type	Complexity score	What business value do we need to see?
Extend an existing table in scope with some scripting	Low to medium	Medium
Build a new scoped application	Medium	Medium
Build a new global application	Medium to high	Critical
Change baseline business rules	High	Critical or mandatory
Build complex, custom integration	High	Critical or mandatory

Step 3: Evaluate the return for the proposed customization.

After you’ve identified your complexity score, you can determine whether the business value of the proposed customization is sufficient to warrant implementation.

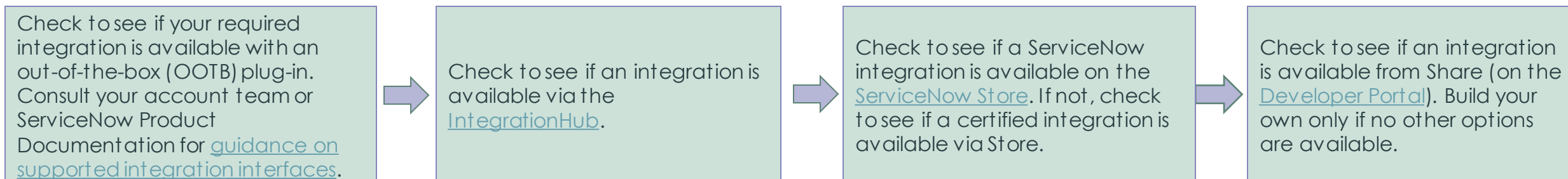
What are the best practices for customization? (Continued)

What technical best practices should I follow for customization?

1. **Avoid copying objects.** Instead, update objects in place wherever possible, except for Service Portal widgets and other items designed to be reused.
2. Default to “**add before edit.**” This means that you should, for example, add fields to forms rather than change the type of an existing field. When adding, avoid using the same names as out-of-the-box objects, methods, or classes. Keep the number of fields you add to a minimum – the more fields you have on a form, the longer it may take to load.
3. **Use the ServiceNow® no- and low-code capabilities** wherever possible, including use of UI policies (before writing client scripts), Flow Designer (over business rule scripts), IntegrationHub (before writing custom integrations), and other capabilities.
4. Use **scoped applications** as your default for any new custom development.
5. **Document all customizations.** Add comments to explain *why* you customized (including business justification), and ensure you review all comments prior to upgrading, to determine if you can revert to out-of-box.
6. **Create tests for all customizations.** You should ensure that you write [Automated Test Framework \(ATF\)](#) tests for all customizations where possible.
7. **Use HealthScan** regularly to [identify unnecessary customizations](#).

What best practices should I follow for custom integration requirements?

For more details on integrations review our checklist on: [Implement integrations with ServiceNow](#)



What are the best practices for customization? (Continued)

How do I deal with customizations when I upgrade?

ServiceNow upgrades will not overwrite customizations you have made but will mark them as skipped records in the ServiceNow [Upgrade Monitor](#). To make sure they're successfully ported to the upgraded instance, you must manually [process the skipped changes](#).

Assuming you've documented all your customizations—including business justification—take your documented inventory and compare it with the skipped records identified in the Upgrade Monitor. After filtering out low-risk changes that have resulted in skipped records (e.g., field labels or form layouts), you'll need to decide whether to:

- **Retain** each customization
- **Revert** to out-of-the-box
- **Merge** your customization with the base system to resolve conflict

For a detailed overview of this process, see our Success Quick Answer, [What's the process to review and address skipped changes?](#)

Related resources

- [Success Playbook – Avoid customization pitfalls so you can innovate and meet demand at scale](#)
- [Success Playbook – Perform ServiceNow upgrades quicker and more effectively](#)
- [Success Checklist – Manage to out-of-the-box](#)
- [No-Code Guide for the Now Platform®](#)

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