Govern your ServiceNow environment

What's in this Success Insight

This Success Insight is an introduction to ServiceNow® environment governance and provides insight into how you can start defining standards to manage your ServiceNow environments. It will help you answer the following key questions:

1. What is ServiceNow environment governance? Why is it important?
2. What concepts and practices do I need to consider when defining ServiceNow environment governance?
3. How do I define policies that support the specific environment governance needs at my organization?

Key insights

- The best ServiceNow environment governance defines instance structure, instance management, cloning practices, and application management.
- Start by defining a minimum viable set of standards as part of a starter management approach rather than trying to deliver a full, mature set of policies all at once.

1. What is ServiceNow environment governance? Why is it important?

ServiceNow environment governance defines the **standards and practices for how to right-size environments, replicate instances correctly, and ensure proper ServiceNow environment maintenance.** It's a key part of ServiceNow technical governance and should be developed and/or approved by your technical governance board.

ServiceNow environment governance sets the architecture and processes needed to establish the appropriate instance structure to support your organization's development, testing, and release processes. It also defines the administrative privileges for each instance, including who should have elevated permissions within an environment and what prerequisites are required to obtain those permissions.

2. What should be considered when defining ServiceNow environment governance?

Your ServiceNow environment governance should define standards for the following practices.

Instance structure

Instance structure defines your ServiceNow instance setup and support model. It is *the* foundational component of environment governance—your instance structure informs how all environment-related policies and standards should be designed. In fact, instance structure impacts all technical governance, including how you govern data, develop, and administer the platform.

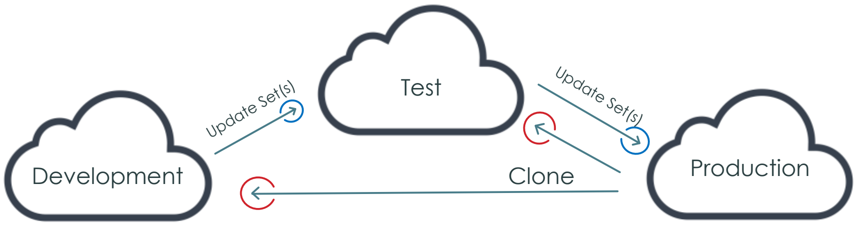
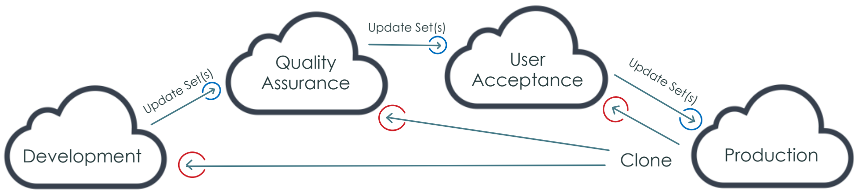
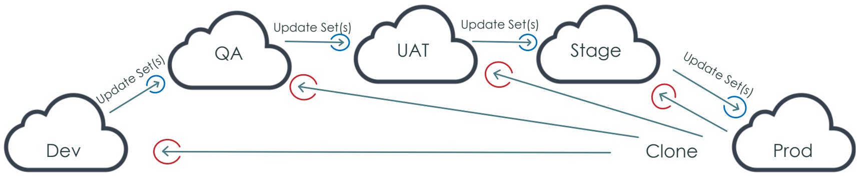
Start defining your instance structure by determining the instance stack (e.g., development, testing, quality, staging, training, production) that you'll need to successfully support your activities within the platform. As you define your instance structure, you should consider items such as the development process standards set by your organization, compliance and regulatory concerns, the effort involved in keeping instances in sync, and the number of development teams working with ServiceNow applications.

Define instance structure early on. This should ideally happen in an early pre-sales engagement with ServiceNow. It's not a go-live event.

Practitioner insight: Periodically revisit your instance structure as part of continuous improvement efforts to assess if you have the right setup to deliver the value you expect from ServiceNow.

If you decide that you need to include additional instances into your instance stack, you should always weight the value of doing so against cost to support and maintain those additional instances over time.

This table shows some example instance structures with details on the benefits and potential concerns of each.

Structure (and the instances involved in each)	Benefits and considerations
Three instance stack (development, testing, and production)  Test Update Set(s) Update Set(s) Development Clone Production	**Benefits –** Including a testing instance allows end user testing to occur without being affected by current development activities. Including fewer instances streamlines the code promotion, cloning, and environment maintenance processes. **Considerations –** Quality assurance testing must occur in the development instance, which could be affected by active development. Also, three instances may not be enough for customers with advanced testing requirements or with parallel development streams.
Four instance stack (development, quality assurance, user acceptance, production)  Update Set(s) Quality Assurance User Acceptance Update Set(s) Update Set(s) Development Clone Production	**Benefits –** By separating the testing instance into quality assurance and user acceptance instances, all testing can occur without being affected by active development activities and it provides a better testing process for end users. **Considerations –** In large environments with multiple applications, the migration to production could be the first time all update sets are migrating together.
Five instance stack (development, quality assurance, user acceptance testing, staging, and production)  Update Set(s) Update Set(s) QA UAT Stage Update Set(s) Update Set(s) Dev Clone Prod	**Benefits –** All testing activities can occur without being affected by the active development activities. The addition of a staging instance allows the testing of the full release package. In some cases, the staging environment can also serve as a production reference environment for troubleshooting issues that occur in production. **Considerations –** This method may increase time to value since you need extra time to support migrating to staging and increasing the number of instances for clone downs.
* The default structure is a two instance stack with a production and sub-production (development) instance. However, given the utility of including a testing instance, the two instance stack is used only rarely.	

Supplementary instances

In addition to the primary instance stack, it may be beneficial to include supplementary instances to support the business, such as:

Instance type	Benefit
Training	A dedicated training instance allows training for end users to occur in a dedicated environment so it doesn't impede the development cycle.
Sandbox	A sandbox lets you test upgrades and perform POCs without affecting the development cycle.
Innovation	An innovation instance lets you explore innovation and improvement projects, including adopting new features without disrupting other development.

Instance management

Once you've determined your instance structure, define policies for how you'll manage each instance included in your structure. Defining these policies upfront will help your organization in many ways including:

- Reducing risk by making sure that changes to instances happen using a defined process and within predefined maintenance windows
- Reducing the mean time to resolve (MTTR) incidents or outages to your instance by clearly defining support procedures and teams
- Preventing incidents and outages by defining the process to modify system properties or enable plugins

Each instance management policy should define at least each area in this table.

Area	Questions to answer for each area
Instance overview	• What is the instance's purpose and intent? • What activities should we conduct on the instance? • What high-level limitations or restrictions are in place for the instance? • Who is the instance's intended audience?
Support procedures	• What is the support procedure to resolve incidents against the instance? • What support groups are involved in resolving incidents against the instance? • What escalation paths are necessary to resolve incidents with higher complexity? • When and how should we escalate to ServiceNow support if necessary?
Critical availability	• What blackout windows exist that prevent us from performing maintenance on the instance? • What compliance or regulatory controls prevent when or how you conduct maintenance?
Maintenance procedures	• What maintenance windows exists for performing routine maintenance and/or change requests against the instance? • What enterprise maintenance windows exist that impact the instance?
Change management requirements	• Does this instance fall under change management control? • What is the change management process for conducting change to the instance? • What change types should we use when creating change requests against the instance?
Advanced permissions	• Who should be granted admin and security admin permissions within the instance? The security admin role is the most elevated administration role. • What are the prerequisites to obtain admin and security admin permissions within the instance? • How will you monitor and maintain permissions for the instance over time? • What elements of this security hardening guide apply to this instance? How should they be set?
System properties	• What review process is necessary before modifying system properties in the instance? • Who should approve modifications to system properties within the instance?
Plugins	• What is the process to activate plugins within the instance? • Who should be consulted before activating plugins within the instance? • Who should approve activating the plugins in the instance?

Practitioner insight

In a large instance when multiple development teams are working together across multiple initiatives, policies regarding system properties should be considered to keep one team from impacting another team's work.

Practitioner insight

Define administration privileges differently across different instances. You can allow more users access to administration in sub-production instances but should restrict this access in production instances.

Teams that are building scoped applications on the Now Platform® should consider setting up application administration as seen in our product documentation.

Cloning practices

The main goal of the cloning process is to make sure that each environment aligns as closely as possible with the production environment. This gives you greater certainty that the activities performed on a given environment are translatable to the production environment.

ServiceNow recommends cloning your production instance to your sub-production instances a minimum of twice per year to keep your instances in sync with one another. A good practice is to clone your instance stack after every major release because this is usually the time when there are the fewest non-committed updates in the stack. To accomplish this, you need a detailed cloning practice to ensure that cloning can be carried out effectively and efficiently, without impacting the day-to-day work that goes on within each instance.

Consider these things when you develop your cloning policies and strategy:

- **Impact on the development cycle –** Cloning can impact active development if you don't take the correct precautions and communicate the appropriate actions. Create a detailed plan of the cloning process and share it with your platform resources to ensure that active development is not lost during the clone. It's also a good practice to build clone time into your release and development schedule to create a repeatable and predictable process for development and platform teams.

  Developers should mark update sets that are works in progress as complete and export all update sets that have not been yet been deployed to production as XML files. If necessary, rename update sets with incomplete work to identify them appropriately (for example, you might add the suffix **[incomplete]**). Also make sure you export any required data records and items that weren't captured in update sets before cloning. Import and commit update sets from XML after cloning and create new update sets that represent the continuation of work. Merge these with the imported update sets to make sure that all work not yet deployed to production is properly captured in update sets.

See our System Clone product document page for more details.

- **Inclusion and exclusion of data –** Many instances contain data in the production instance that's considered sensitive. Since most sub-production instances have users with elevated permissions, it's important to determine which data should and should not be included in the clone to prevent spillage. This table is an example of how to allow or restrict certain data access and use per instance.

Item	Dev instance	Testing instance	Staging and production instance
Companies	Allowed	Allowed	Allowed
Account	Allowed	Allowed	Allowed
Customer	Not allowed	Not allowed	Allowed
Manufacturer	Allowed	Allowed	Allowed
Services	Allowed	Allowed	Allowed
CIs	Not allowed	Not allowed	Allowed
Locations	Not allowed	Not allowed	Allowed
Models	Allowed	Allowed	Allowed
Products	Allowed	Allowed	Allowed
Contracts	Not allowed	Not allowed	Allowed
Users	Not allowed	Not allowed	Allowed
Catalog items	Allowed	Allowed	Allowed
Tickets	Not allowed	Not allowed	Allowed
SLAs	Allowed	Allowed	Allowed
Assets	Not allowed	Not allowed	Allowed
Licenses	Not allowed	Not allowed	Allowed

- **The source and targets of clones –** Determine the sequence of cloning from production through development so you're less likely to affect active efforts in sub-production instances such as user acceptance testing (UAT), training, or development. There are two ways this is most commonly done:

1. You can clone the production instance to the next lower instance (that is, staging or testing), and use that instance as a source for all other clones.
2. You can clone the production instance to all other instances.

Practitioner insight

Customers that clone their instance stack after every major release and at least twice per year experience fewer conflicts when promoting configurations into production and have smoother upgrades.

Application management

It's also important to understand each application's use, it's purpose, and the details about how to govern and manage each application as part of your ServiceNow environment. Defining these policies upfront will help your organization in many ways, including:

- Reducing risk by making changes to applications using a defined process and within predefined maintenance windows
- Reducing the mean time to resolution (MTTR) of incidents or outages to your applications by clearly defining support procedures and teams
- Preventing incidents and outages by defining the process to modify system properties or enable plugins.

To do this, define policies for how you'll manage each application on ServiceNow. Each application management policy should define at least the areas shown in this table.

servicenow

Area	Questions to answer for each area
Ownership and stewardship	• Who is the owner of the application? • Who are the stewards of the application, if any? • How can I contact the owner if necessary?
Support process	• What is the support procedure to resolve incidents against the application? • What support groups are involved in resolution of incidents against the application? • What escalation paths are necessary to resolve incidents of higher complexity? • When and How should you escalate to ServiceNow support if necessary?
Critical availability	• What blackout windows exist that prevent you from performing maintenance on the instance? • What compliance or regulatory controls prevent when you conduct, or how you conduct maintenance?
Maintenance	• What maintenance windows exists for performing routine maintenance and or change requests against the instance? • What enterprise maintenance windows exist that impact the instance?
Community and fulfillers	• What users are dependent, and in what way are they dependent on this application? • How do users access the application? • Who are the fulfillers of the application? • Who are approvers within the application, if any? • What critical functionality is supported by this application?
Security rules	• What specific security rules are in place for this application? • What compliance and regulatory controls affect this application? • What elements of this security hardening guide apply to this application? How should they be set?
Standing operational and organization decisions	• What operational or organizational decisions were made regarding this application?
Data management	• What data volume do you expect? • What is the data lifecycle? You can learn more about data • How will you manage data archiving and retention requirements? Review our Success Insight on defining ServiceNow data governance for more information on data management.
* Keep in mind that some of this information should be tracked and managed within the CMDB.	

3. Who should be involved in defining ServiceNow environment governance?

Your ServiceNow platform owner and ServiceNow technical governance board should be accountable for managing the creation of all technical governance and policy related to ServiceNow, including environment governance.

If your organization doesn't have a ServiceNow technical governance board, see our resource on getting started with ServiceNow governance.

Consider involving these roles when you define your ServiceNow environment governance if they're not already participants on your technical governance board.

Role	Description
Enterprise architect	The enterprise service architect creates one single language between people, processes, and technology that allows seamless sharing and enriching of data across IT functions. The EA could be consulted when you define the instance structure, instance management policies, and application management policies. The EA should at least be informed of this work.
Platform owner	The platform owner is typically accountable for managing the creation of all environment governance policies.
Platform architect	The platform architect is typically responsible for the creation of environment governance policies while consulting members of the organization, including enterprise architects, security administrators, data owners, and application owners.
Security administrator	Security administrators are consulted during the creation of environment governance policies to safeguard that the instances are protected according to organizational security policies and guidelines.
Development leads/SMEs	Development leads/SMEs should be consulted when you define the instance structure, instance management policies, and cloning processes because the output of these activities will affect the development flow and process.

4. What’s the starter or minimum viable approach to ServiceNow environment governance?

Ideally, ServiceNow environment governance should define practices for instance structure, instance management, cloning practices, *and* application management. That’s a lot to get right from the start. We recommend starting with a minimum viable policy that defines the most important standards and controls for your specific needs first instead of trying to define everything all at once.

Take these steps to get started:

1. Connect with your ServiceNow platform owner and, if established, your ServiceNow technical governance board. They should be accountable for managing the development of ServiceNow environment governance policies at your organization.
2. Work with your platform owner to select one to three of the policy components listed in section 2 (for example, “instance structure”) that you think are the most urgent and important to enact at your organization. In our experience, at least instance structure and instance management are most important to include in a minimum viable policy.
3. Consult with your technical governance board to approve of the components you selected. If necessary, invite any subject matter experts to help your technical governance board consider what’s needed in initial ServiceNow environment governance.
4. Build out specific policy standards for your selected components.
5. Expand on your minimum viable approach to include standards for the remaining components in section 3.

5. How should I define specific ServiceNow environment governance policies?

While this Success Insight covers what to consider when you define ServiceNow environment governance, each organization needs to define the specific policies and standards required at their organization. You’ll need to account for any special needs required by:

- **Industry –** Companies operating in highly regulated industries (government, finance, and healthcare, for example) will likely require governance policies that more strictly enforces environment standards and controls. For example, if your developers need a certain certification or clearance to develop with real data (PII, location, etc.), is there a lower level environment that individuals could develop in for UI and workflow?
- **Existing enterprise governance and/or environment standards at your organization –** Many organizations already have defined environment standards. Your ServiceNow environment governance will need to align with (and usually be subordinate to) an existing, enterprisewide policy at your organization.

- **IT architecture –** Your IT architecture may impact how you need to design ServiceNow environment governance. Specific policies should account for how your IT environments are designed. This is especially true if ServiceNow is highly integrated with other systems.

When possible, we recommend working with your partner or with Now Platform architects to define specific ServiceNow environment governance policies that will work for your organization. Reach out to your account manager to connect with ServiceNow architects to help with this.

Additional resources

- ServiceNow governance resources on the Customer Success Center
- Get started with ServiceNow governance
- Define ServiceNow technical governance policies
- Defining ServiceNow governance golden rules
- Define ServiceNow data governance
- Consolidate production instances resources on the Customer Success Center

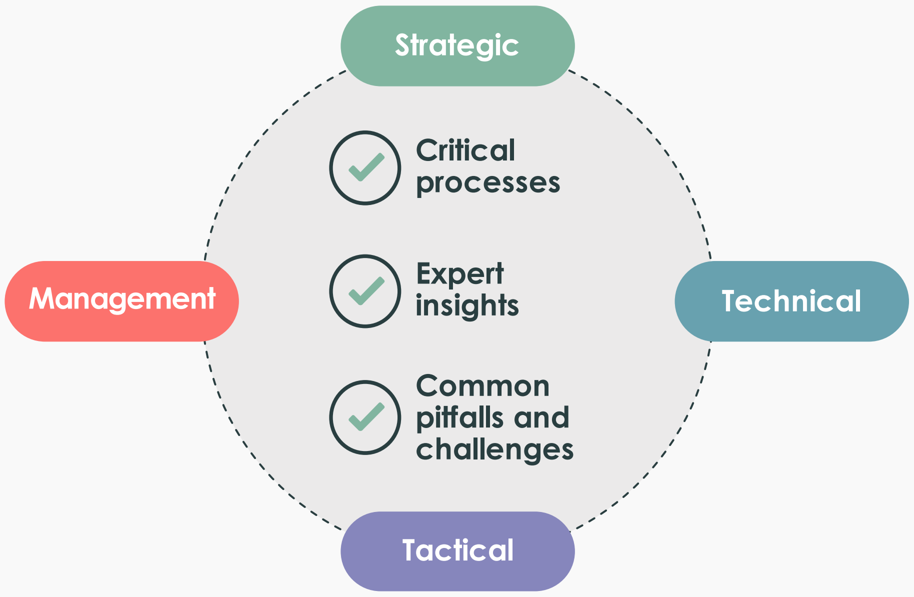
If you have any questions on this topic or you would like to be a contributor to future ServiceNow best practice content, please contact us at best.practices@servicenow.com.

servicenow

Customer Success Best Practices

ServiceNow's Best Practice Center of Excellence provides prescriptive, actionable advice to help you maximize the value of your ServiceNow investment.

Definitive guidance on a breadth of topics



Designed for:

- Executive sponsors
- Platform owners and teams
- Service and process owners

Created and vetted by experts

- **Best practice insights** from customers, partners, and ServiceNow teams
- Based on **thousands of successful implementations** across the globe
- Distilled through a **rigorous process** to enhance your success

Proven to help you transform with confidence

- Practical
- Actionable
- Value-added
- Expert-validated

Get started today.

Visit Customer Success Center.

Contact your ServiceNow team for personalized assistance.