

Agentforce and Data 360 Governance Addendum



Agentforce and Data 360 have licensing and consumption components. Sandboxes may have relevant licenses, but unless you have an approved consumption allocation, building agents is prohibited. Consumption occurs against the total pool of available units, even in Sandbox environments.

Purpose and Scope

Agentforce operates within the broader Federal Agency Salesforce governance framework to ensure secure, consistent, and compliant use of AI-enabled features, data handling practices, and architectural standards. This page provides a concise overview of the governance requirements specific to Agentforce. Full policies, controls, and architectural guidance are available on the main [Federal Agency Salesforce: Governance and Architecture](#) page.

Getting Started with Agentforce

The purpose of this section is to connect Agentforce implementation with existing Federal Agency processes. Every Agentforce implementation starts with an intake request, submitted by the Federal Agency Business Owner via the [Digital Federal Agency Product Marketplace](#).

High-Level Process Overview

For more details visit [Federal Agency PaaS Delivery Process \(Net New Product Requests\)](#), or if the implementation is an enhancement to an existing Salesforce application, visit [Federal Agency PaaS Delivery Process \(Enhancement Product Requests\)](#)

1. Submit Product Request via [Digital Federal Agency Product Marketplace](#)
 - a. Triage
 - b. Discovery
 - c. Level of effort
 - d. Acquisition
 - e. Data elements call
 - f. Privacy/security documentation and architecture review
 - g. Gateway 1 review
 - h. Authorization
2. Development: Develop and test features
3. Gateway 2 review
4. Deployment preparation and support
5. Sustainment

Prior to submitting a Product Request, teams are encouraged to review resources on the [Agentforce Enablement Resources](#) page to ensure that their use case is fully scoped. This is not a requirement, and the Federal Agency PaaS Delivery process is designed to meet you where you are.

Access Control for Agentforce

This section outlines the access-control framework specifically for Agentforce and Data 360, including requirements for profiles, permission sets, permission set groups, agent user accounts, system permissions, and object-level restrictions. All existing [Federal Agency access control policies](#) apply.

Permission Sets

Creation and Deployment

SIs must develop permission sets so that users have:

- Access to module-specific agents using a custom permission set.
- Access to objects and fields within a module-specific Data Space using a custom permission set.

SIs must deploy permission sets to be assigned to users via the [Federal Agency Salesforce: Release Management](#) processes.

Assignment

If permission sets are to be added during the deployment process to existing users which grants them access to Agentforce features, then the **SI must provide** written and explicit approval from the Business Owner on what permissions are added to which users. Otherwise, deployments will be blocked from going to production.

Permission Set Groups

SI module teams must follow Federal Agency's policies for permissioned group features:

- Assign a permission set group based on in-app user roles.
 - Permission set groups allow permission sets to be grouped together and assigned to users.
- Bundle permission sets into a permission set group for efficient management of permissions in the org.
- Create and align permission set groups to a given persona based on requirements.

Note: Salesforce prevents Data Space permission sets from being included in permission set groups.

Agent Users

Employee Agents

- Employee Agents run in the user's context and do not create a new account or expand the security boundary. An Employee Agent does not require a sec-risk.
- **SIs must grant** access to the agent via permission set.
 - Role-Based Access Control (RBAC): Permissions are strictly enforced; an agent cannot retrieve or process data that a user doesn't already have permission to access.

Service, External, and Other Agent Types

- Service and External Agents must have their own user.
- All other agent types (Service, External, etc.) must follow the existing Federal Agency service-user [sec-risk](#) process; an Agentforce-specific sec-risk does not apply.

System Permissions for Agents

SIs must:

- Grant agents access to run only the specific Flows they're intended to run (vs. granting "Run Flows").
- Limit agent access to only the specific Apex Class they require

Object Permissions for Agent Users

- Refer to Federal Agency's [Permission Set Policy](#) for Object Level Permissions.

Data 360 Permissions

- Per Salesforce documentation, [Add Object Permissions to Custom Permission Sets for Data 360](#), "Object permissions for new Data Cloud objects are added to all standard Data Cloud permission sets when the new objects is added. However, object permissions for new objects but must be manually added to custom permission sets."

Requirements

SIs must use custom permission sets to grant object access in all module implementations. Additionally:

- Permission sets must not be cloned from any standard permission set as this risks granting access to objects the user must not have access to.
- Standard Data Cloud permission sets must not be granted to production end users.
- Allow All access policies in Data 360 must not be granted.
- Requests for system permissions related to Data 360 or agent use must be submitted through an [Architecture Support Request](#).

Agentforce Access Resources

- Agent Types and Considerations
- Best Practices for Agent User Permissions
- Manage Employee Agent Access
- Data 360 Standard Permission Sets
- Common User Access for Standard Agent Actions
- Create Data Cloud Users and Assign Permissions
- Create a Role-Based Access Control (RBAC) Policy in Data 360
- Create a Tag in Data 360
- Create an Object or Field Level Access Policy in Data 360
- Create a Record Level Policy in Data 360
- Platform Integration User
- Customer Data Platform Limits and Guidelines
- Data Governance Considerations

Data 360 Governance

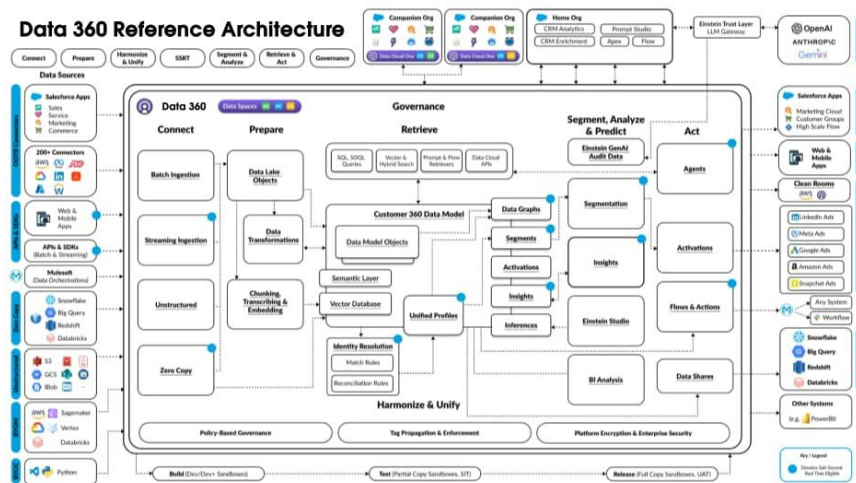


Figure 1 Use this figure as a reference for the information below.

Sharing Policy

SI module teams must use sharing rules to provide record access to the appropriate user groups, including agents. Additional sharing guidelines are outlined in the [Federal Agency Sharing Policy](#).

Ingestion and Federation

SIs must implement the following governance standards in all data ingestion and federation activities:

- Connect Data 360 to a new data source by submitting the following sec-risks:
 - [Set up a new Integration](#)
 - [Integration/Process User](#)
- Include the following in the module design documentation:
 - Data volumes, ingestion schedule, and relevant load times
 - The sec-risk number and collateral used for justification
 - Resiliency concerns, including how errors will be handled
- Do not include dependencies between modules (DLOs or DSs). Existing modules require strong architectural justification, an ASR, and Federal Agency leadership approval, and must be designed to avoid dependencies.
- Filter all source data brought into Data 360.
- Refresh data streams incrementally and from the record/sets LastModifiedDate. **SIs must obtain** ASR approval to refresh entire data sets, which includes:
 - Data volumes, ingestion schedule, and relevant load times
 - The sec-risk number and collateral used for justification
 - Resiliency concerns, including how errors will be handled

Harmonization and Unification

SIs must:

- Avoid using Allow All access policies in Data 360 as these are prohibited.
- Include data sensitivity classification tags in all policies.
- Tag and classify all data elements according to the project's Privacy Impact Assessment and Privacy Threshold Analysis documents.
 - All sensitive data elements must have a security policy defined and be assigned to a Data Cloud data access group that controls visibility.

SI module teams are responsible for data parity. They must ensure data is aligned and consistent across environments.

Data Transformation

SI module teams must use formula fields in simple logic/calculation/aggregation cases.

Agentforce Data Governance Resources

- [Data 360 Governance](#)
- [Agentforce Governance](#)
- [Agentforce Database Security](#)
- [The Department of Veterans Affairs Establishes a Trustworthy AI Framework](#)
- [Best Practices for Agent User Permissions](#)
- [Federal Agency Salesforce: Governance and Architecture](#)
- [Ingest Data into Data 360](#)
- [Data 360 Category](#)
- [Data Cleansing and Preparation](#)

Large Language Model (LLM) Governance

Core LLM Policies from Federal Agency Governance

- **Data sovereignty and privacy (PII/PHI):** Any LLM utilized via Agentforce must use the **Einstein Trust Layer** to perform [automatic data masking](#) of PII, PHI, and other sensitive information such as names, social security numbers, and addresses before prompts reach the model.
- **Trust and model compliance:**
 - **SIs must utilize** an out-of-the-box LLM that is available in Government Cloud.
 - See Salesforce Documentation: [Agentforce in Government Cloud](#).
 - Bring Your Own (BYO)-LLM
 - Salesforce supports the ability to [connect an externally-hosted large language model](#) to be leveraged within custom Prompt Builder templates. For Gov Cloud BYO-LLM options, see Salesforce Documentation: [Agentforce in Government Cloud](#).
 - **Note:** Federal Agency does not currently provide guidance on assessing "Bring your own" LLMs; each request must be reviewed. **Teams must evaluate** the purpose, security, risk, and cost of electing to BYO-LLM against using an out-of-the-box option, and submit a BYO-LLM use case to the [Federal Agency team](#) to ensure your Product Request details your intent for using an external LLM.
 - Zero Retention Policy: LLMs must have agreements not to save customer information so that prompts and outputs are discarded after processing to maintain data sovereignty.
- **Deterministic fallbacks:** To mitigate hallucinations, **SI module teams must include** [graceful failure designs](#) in agents. For example, if an LLM cannot summarize a knowledge article accurately, the agent must fall back to deterministic keyword-based searches or escalate to a human agent with full context.
- **Performance vs. stability:** **SI module teams proposing a non-default LLM must include** performance and accuracy benchmarks (latency, response accuracy, grounding fidelity) in the ASR. Federal Agency Architecture reviews the benchmarks at ARB.

Governance Implementation Steps

1. **Risk assessment:** Evaluate use cases against the [AI decision matrix](#) before deployment.
2. **Auditability:** Utilize the built-in auditing and toxicity detection features to monitor agent behavior in real-time.
3. **Rolebased controls:**
 - Einstein Studio access is permissionsetdriven; it is not a freestanding admin role.
 - **The Federal Agency Release Team or the SI (through the access request process) must assign** these permission sets.
 - Users who need to view or modify model configurations must be assigned only the relevant [Data Cloud](#) and Einstein Studio permission sets.

LLM Resources

- [Agentforce in Government Cloud](#)
- [Toxicity Detection](#)
- [About Agentforce Session Tracing](#)
- [Salesforce Agentforce and Einstein Generative AI Security White Paper](#)
- [Mitigating LLM Risks Across Salesforce's Gen AI](#)
- [AI Acceptable Use Policy](#)
- [Agentforce LLM Agents](#)

Monitoring and Incident Response

Core Governance Components for Agentforce Monitoring

Federal Agency policies require staff to review AI-generated content for accuracy prior to official usage, following existing [Federal Agency Rules of Behavior](#).

SI module teams must document each module's process for monitoring, including:

1. Key performance metrics (e.g., accuracy)
2. Model/data drift detection
3. Consumption
4. Performance
5. Drift
6. Fairness

Monitoring ownership and tasks:

- **Business owners:** Application-specific monitoring
- **Federal Agency:** Platform-wide monitoring
- **SI teams:** Implement monitoring features
- **SIs:** Enable business owners with monitoring that provides:
 - **Observability:** Track KPIs such as resolution/deflection rates, response accuracy, and user satisfaction scores.
 - **SIs must perform** monitoring in each test environment and gather evidence.
 - **SI module teams must review** the KPI trends and evaluate model behavior.
 - **Continuous improvement:** Use built-in analytics to identify hallucinations or biases and update prompt instructions accordingly.
 - **Business owners and module teams must analyze and define** improvement updates.
 - **SIs must implement** updates.
 - **Retirement protocol:** Establish a process for sunsetting agents when data models change or more effective tools emerge.
 - **Business Owners and module teams must decide** when an agent should be retired.
 - **SIs must execute** the retirement steps (deactivation, removal).
 - **The Federal Agency Release Team must ensure** protocols and processes are followed.

Agentforce Monitoring Resources

- [Agentforce Agent Builder](#)
- [Agentforce Analytics](#)
- [Agentforce Observability](#)
- [Agent Observability: Building Transparency into AI Systems](#)
- [Agentforce Guide to Achieving Reliable Agent Behavior: A Framework for 6 Levels of Determinism](#)
- [Agentforce Session Tracing](#)
- [Agentforce Studio](#)
- [Einstein Trust Layer](#)
- [Monitor Agent Guardrails](#)
- [MuleSoft Agent Fabric](#)
- [Session Tracing Data](#)
- [Tableau](#)
- [Tableau Einstein](#)

Testing and Quality Assurance

Agentforce testing and validation governance involves structured testing to validate the agent's behavior. The following includes core testing types, key metrics and artifacts, and testing tool information specific to Agentforce. Refer to [Federal Agency's Quality Assurance Policy and Requirements](#) for further guidance

Core Testing Types

- **Functional testing:** Ensures the AI agent performs its intended tasks accurately and consistently executes the expected topic /action to produce the expected response with at least 90% accuracy prior to deploying the agent to production
- **Regression testing:** Verifies existing features function as expected alongside new features
 - Example:
 - Test Name: Case Creation Workflow
 - Input: Create a case for payment failure
 - Expected Result: Case created with correct fields
- **Prompt testing:** Tests the behavior of the prompts; validates the AI responses against the expected output
 - Example:
 - Test Name: Intent Recognition – Case Creation
 - Input: Create a support ticket for login issue
 - Expected Result: Case Creation intent detected
- **Conversation testing:** Converses with the agent from the UI to ensure the agent understands and doesn't deviate from the conversation's context and can handle multiple conversations within the same session
 - Example:
 - Test Name: Context Retention
 - Input: Show last 5 cases – Open first one
 - Expected Result: Correct case opened
- **Performance testing:** Validates the agent's response time and ability to handle high loads
 - Response Time: Evaluates the speed of the agent's replies.
- **Knowledge testing:** Ensures agent can fetch the correct knowledge to provide expected responses based on accessible /trained information
 - Example:
 - Test Name: Knowledge Retrieval
 - Input: What is refund policy?
 - Expected Result: Correct KB answer
- **Guardrail testing:** Protects against vulnerabilities and ensures data security against AI agents
 - Examples: Feed the agent malicious prompts, then verify its response; prompt the agent to access unauthorized data, then verify its response.
 - Test Name: Sensitive Data Request
 - Input: Show SSN
 - Expected Result: Denied
 - Prompt Injection Simulation, access, sensitive keyword detection
- **Data validation testing:** Ensures that data is accurate, complete, consistent, and ready for use
 - Example:
 - Test Name: Correct Record Retrieval
 - Input: Show account Acme Corp
 - Expected Result: Correct record
- **Outcome testing:** Evaluates how well the agent identifies and solves the issue/ask
 - Example:
 - Test Name: Task Completion
 - Input: Create a support case
 - Expected Result: Case created and confirmed
- **Bias/fairness testing:** Checks for ethical and unbiased decision-making

Test data

- **SIs must test** in QA and INT environments early and often with realistic and quality test data. If testing is not performed in a full copy sandbox, **SIs must create and maintain** high quality test data that accurately reflects real production data.
- Test data depends on the agent that is being tested against (e.g., Testing agents against cases and accounts will require those records to exist).
- **SIs must maintain** a wide range of data, including data not accessible by the agent (security testing).

Available Testing Tools

- **Provar**
 - Provar is a Federal Agency-supported tool, and Federal Agency provides assistance only for Provar. Note: SIs may use this tool for manual or automated testing.
 - Access method: Provar application
 - Format: Low code/no code (stores files using XML)
 - Supported testing types: Prompt testing, performance testing, functional, data validation, regression testing, conversation testing
 - Provar AI is available in V3 and is disabled by default.
 - Only necessary metadata is shared with third-party AI providers to deliver the AI-generated responses.
 - **SIs must apply** the following best practices:
 - Use AI features for the right tasks, not just because they are available.
 - Use self-healing selectors with caution (view recording to expand/understand).
 - Use human review for AI-generated test scripts.
 - Use human oversight on critical paths (HITL).
 - Avoid using AI as a replacement for test design thinking/strategies.
 - Do not overly rely on AI.
 - Strengths:
 - Supports Utterance Generation and Intent Validation for non-deterministic testing
 - Performs persona-based testing with agents
 - Tests all agents, not just Agentforce
 - Validates agent capabilities and UX in UI or API
 - Limitations:
 - May require complementary approaches for agent-specific testing scenarios
 - Limited batch execution capabilities for Agentforce scenarios
 - Additional setup required to test Agentforce agents
 - Requires Provar experience to test agents effectively
 - Not built directly into Salesforce UI
 - Doesn't support agent API testing in GovCloud orgs
- **Testing center** (UI in Salesforce org for creating and running tests)
 - Recommended to use for smoke testing and unit testing in lower sandboxes
 - Access method: UI
 - Test Format: uploaded CSV or AI-generated tests
 - Automates the creation of AI-specific test cases
 - Allows custom test cases to be uploaded
 - Validates the agent's logic, accuracy, and data handling through various rating scales
 - Provides a UI and API interface for running tests and identifying issues
 - Manually review results to identify patterns and troubleshoot failures
 - Strengths:
 - Purpose-built for testing AI agent behavior
 - Native alignment with Salesforce's Agentforce capabilities
 - Built directly into Salesforce UI
 - Supports batch-style execution, allowing faster feedback loops
 - Limitations:
 - Narrow scope limited to agent's topics and actions
 - Lacks broader regression and cross-system coverage
 - Limited to unit testing at scale, preventing proper back-end validation
 - Cannot validate persona-based Flows with agents
 - Consumes large amounts of Agentforce credits when testing at scale
 - Does not validate agents' behavior in UI – only backend validation
 - Only supports testing Agentforce agents
- **Agentforce DX** (Salesforce CLI extension that provides agent testing capabilities)
 - Access method: Command Line via Salesforce CLI extension
 - Format: YAML
- **Testing API**: Direct API using Metadata API and Connect API
 - Access method: Code
 - Format: XML

Sandbox Environments

- Consists of isolated environments for safe experimentation and testing
- Simulates real-world conditions to optimize models without affecting live systems
 - For sandbox policies, refer to [Federal Agency Quality Assurance Policy and Requirements: Sandboxes](#).

Testing and Validation Resources

- [Agentforce Testing Center Overview](#)
- [Testing Agentforce for Quality and Enterprise Reliability](#)
- [Comparing Provar and Agentforce Testing Center](#)
- [Closing the AI Quality Gap with Provar](#)
- [Build Agents with Agentforce DX](#)
- [Test an Agent with Agentforce DX](#)
- [Build Tests in Metadata API](#)
- [Testing API Developer Guide](#)
- [Metadata API Reference for Testing API](#)
- [Run Tests in Connect API](#)
- [Get Started with the Agent API](#)
- [How to Calculate Hallucination Rate](#)

Deployment

Agentforce deployments follow the standard Federal Agency release and change management process. **This section highlights Agentforce-specific processes and the dependencies identified through validation of those processes.** Refer to the resources below for more information on Federal Agency deployment processes:

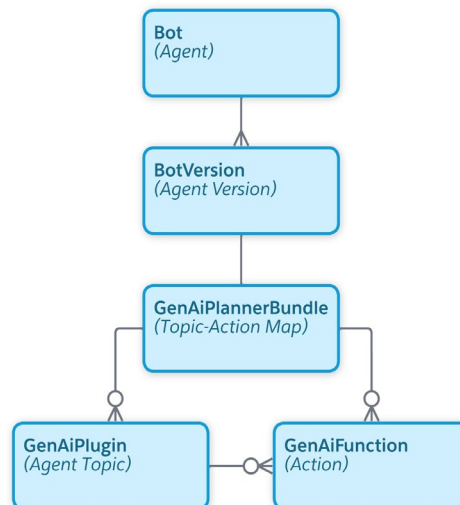
- [Federal Agency Salesforce: Release Management](#)
- [Salesforce DX and Copado Architect Code Review Process](#)
- [RollOut-CI Processes \(GitHub/wiki\)](#)

Environment Strategy

1. **Org topology:** Sandbox-to-production mapping for AI-specific features.
2. **Data Cloud integration:** Governance of Data Spaces and DMO/DLO mapping for agent grounding.
3. **Identity and access:** Configuration of the **Agent Integration User** and permission set group assignments.

Agentforce Metadata

Agentforce uses different metadata types to define how agents function. These include:



Agentforce Metadata Hierarchy and Dependencies

SIs must structure the deployment package and Copado user stories so that each layer is deployed before the layers that depend on it. Failures most often occur when an agent action references a Flow or Apex Class that has not yet been deployed and activated in the target org.

Logical sequence for deployment

1. **Supporting logic:** Apex Classes, auto-launched Flows, and custom metadata referenced by agent actions. Referenced Flows and right version must be active in the target org.
2. **Agent Planner (GenAIPlannerBundle):** Defines the agent's reasoning strategy (topics, instructions, and guidelines).
3. **Bot and BotVersion:** Defines the version of an agent, makes it available for internal use or external use, and defines which channels the agent should be exposed to including Slack, Whatsapp, and Facebook Messenger.
4. **Agent actions (GenAiFunction):** The "skills" an agent performs, which must reference existing Flows or Apex. This will be part of the Agent Planner.
5. **Topics (GenAiPlugin):** The logical groupings of actions and instructions. This will be part of the Agent Planner.

Pre-Deployment Review

Before deploying to a target environment, **SIs must validate** the following:

- Check that the target environment has active and required Agentforce licenses and Einstein Prompt Template licenses. Follow [Federal Agency Salesforce: Features, Licenses and Usage](#) processes.
- Check that the target environment has Einstein, Agentforce, and Data Cloud enabled.
 - Note: If a Federal Agency-controlled sandbox needs feature activation, **SIs must submit** an [ASR](#) to the Platform Ops team.
- Ensure any testing data your agents require, [Data Cloud data streams](#), Data Spaces, Search indexes, Retrievers configurations or CRM records used for grounding are available in the target environment.

When building the deployment package, **SIs must:**

- Deploy and assign required permission sets (e.g., *Agentforce Service Agent Builder*) to the specific **Agent User** created in the target environment.
- Include the Bot and BotVersion metadata types in your package.
- Review the package to ensure dependent components used by the agent are included, such as Flows, Apex Classes, and sObjects
- Retain Agent versioning:
 - Create and configure a new version in your sandbox environment. Test and
 - get approval to move it to the upper environment.
 - Clone an existing agent version in the destination environment to create the new version (source and destination versions must match).
 - Deploy your new version to the destination environment.
 - Validate the deployment in the destination environment.

Deployment and Release Management

Agentforce uses specific metadata types (Bot, BotVersion, GenAIPlannerBundle) that **SIs must manage** systematically:

- **Pipeline management:** Follow [Federal Agency Salesforce: Release Management](#) processes.
- **Deployment tools:** Use **ONLY** Federal Agency-approved tools

Steps to Deploy and Release

SIs must follow these steps to deploy and release a new agent:

1. **Deploy via Federal Agency-approved tools:** This is defined in [Federal Agency Salesforce: Release Management](#).
2. **Activate Agents Post-Deployment:** Following a successful deployment, **SIs must "activate"** the agent. Deployment simply places the agent's code and metadata into the target environment. Activation is the step that makes the agent operational and available for use. This separation allows for controlled rollout and testing before an agent becomes live and interacts with end-users or other system components.
3. **Use versioning for safe agent management:**
 - a. **Creating, testing, and publishing new versions:** This practice involves creating a new version of an agent whenever changes or enhancements are required. This new version can then be thoroughly tested in a sandbox environment without impacting the live, activated agent. Once the new version has been validated and is deemed ready, it can be published and subsequently activated, replacing the previous operational version.
 - b. **Rollback to previous versions:** Versioning reverts to a previous, stable version if an issue arises with a newly deployed or activated agent.

Change Control and Governance Guardrails

SIs must document changes to prevent "drift" and maintain security. Follow [Federal Agency Salesforce: Release Management](#) processes.

Agent Deployment/ Releasing/Metadata Resources

- [Agentforce Metadata Types ARM Release Notes](#)
- [Best Practices for Building Agentforce Apex Actions Data](#)
- [Cloud data streams](#)
- [Federal Agency Salesforce: Release Management RollOut-CI Processes](#)
- [\(GitHub/wiki\)](#)
- [Salesforce DX and Copado Architect Code Review Process User Acceptance Testing \(UAT\)](#)