

The AI Agents category represents 35% of the Salesforce Certified Agentforce Specialist Exam. This section focuses on the operational logic, security, and deployment of agents across various business functions.

Based on the sources, here is a list of the content and specific problems (exam-style scenarios) mapped to the sub-categories you provided:

1. Manage Deterministic Behaviour Using Filters and Variables

This category involves ensuring an agent follows strict business rules rather than relying solely on the probabilistic nature of a Large Language Model (LLM).

- Number of distinct scenarios found: Approximately 5.
- Aligned Problems:
 - Restricting Refunds: A scenario where an agent must only allow refund processing for customers with an "Active" status. The solution is creating a context variable for account status and applying a conditional filter.
 - Identity Verification: Ensuring an agent only shares sensitive account details after a "Verify Identity" action confirms verification. This is managed by setting the verification status as a variable and applying a filter to subsequent actions.
 - Triage Logic: Implementing a process where high-severity requests are escalated to humans while low-severity requests create a support case. This is achieved by populating a custom variable via a triage action and using filters.
 - Tiered Access: Limiting booking actions to "Premium" or "Elite" membership tiers using a context variable mapped to the membership field and a conditional filter.
 - Execution Sequence: Ensuring an agent retrieves available sessions, verifies eligibility, and then creates a booking in a specific sequence to avoid failures. This is managed using custom variables to store completion status for each step and applying conditional filters.

2. How an Agent Works and the Reasoning Engine

This focuses on the "brain" of Agentforce and how it translates user requests into actions.

- Number of distinct scenarios found: Approximately 7.
- Aligned Problems:
 - Reasoning Engine Function: The primary role of the reasoning engine is identifying agent topics and actions to respond to user utterances.
 - LLM Role in Planning: The LLM's role in understanding intent is to identify the best matching topic and actions and determine the correct order of execution.
 - Planner Action: When a user says "Show me all the customers in New York," the planner service uses the Query Records standard action.
 - Topic Selection Criteria: The reasoning engine uses the Classification Description and Scope of a topic to decide if it matches a user's request.
 - Action Selection Factor: After a topic is selected, an important factor the engine uses to select a specific action is the priority given to each action.
 - Handling Ambiguity: If an agent cannot understand a request, it responds with a preconfigured message based on the action type.

3. Select and Configure Topics and Actions based on Agent Types

This involves choosing the right configuration for different functional agents (Service vs. Sales vs. FAQ).

- Number of distinct scenarios found: Approximately 5.
- Aligned Problems:
 - Knowledge-Based Q&A: For an agent that answers questions using Knowledge articles, the Specialist should use the General FAQ topic and the Answers Questions with Knowledge

action.

- Assigning Actions: The correct method to assign actions is to assign them to a Topic first in Agent Builder.
- Custom Action Instructions: When creating custom actions, the Action Instructions are critical because they tell the LLM which action to use.
- Unique Descriptions: If an agent provides incorrect summaries, the Specialist should investigate if the Action Instructions are unique enough for the engine to distinguish between them.

4. Manage Agentforce Security and the Agent User

This covers the permissions required for agents to act on behalf of the organisation.

- Number of distinct scenarios found: Approximately 6.
- Aligned Problems:
 - Knowledge Access: If an agent fails to provide answers from Knowledge articles, it is likely because the Agentforce Service Agent user was not assigned the Allow View Knowledge permission set.
 - Object-Level Security: When a Service Agent cannot assist with a new custom object (e.g., "Product Replacement"), the assigned permission set must be updated to include Read access to that specific object.
 - SDR Visibility: If sales reps cannot find a newly deployed Sales Development Representative (SDR) agent, the cause is often that they are missing the Use SDR Agent permission set.
 - Least Privilege: Ensuring an agent cannot access sensitive medical research stored internally by following the principle of least privilege and avoiding granting permissions to that object.
 - Criteria-Based Access: Ensuring an SDR agent only accesses Leads in its assigned region using criteria-based sharing rules.

5. Identify when to use Employee, Service, or Sales Agents

This requires matching business requirements to the specific capabilities of each agent type.

- Number of distinct scenarios found: Approximately 5.
- Aligned Problems:
 - Employee Agent: Used for internal tasks like empowering a marketing team to find campaign data, generate creative content, and manage project tasks.
 - Sales Agent: Used for assisting sales staff in scheduling daily tasks and providing detailed explanations behind product prices and deals.
 - Service Agent: Used to resolve customer cases end-to-end, such as handling guest complaints and offering upgrades or hotel credit.
 - SDR Agent: Specifically used to qualify and nurture leads 24/7 and automatically book meetings for qualified prospects.

6. Connecting Agents to Various Channels

This covers the technical requirements for deploying an agent to Slack, websites, or email.

- Number of distinct scenarios found: Approximately 4.
- Aligned Problems:
 - Slack Connection: To connect an agent to Slack, the Specialist must create a connection between Salesforce and the Slack workspace and install the agent.
 - Digital Experience/Web: A required step to connect an agent to a Digital Experience site is creating an Omni-Channel flow that routes messages to the agent.
 - Channel Specifics: SDR Agents are typically noted as requiring deployment in the Messaging channel.
 - Email Integration: To connect an email template to a Service Agent, the Specialist should create an Email Configuration for the agent.

Analogy for Understanding: Think of the Reasoning Engine as a Master Chef (the brain). The Topics and Actions are his Recipe Book (the configuration). If he doesn't have the right permissions to the Pantry (Data Security), he can't get the ingredients. Deterministic behavior is like a Strict Health Code he must follow regardless of his creativity, and the Channels are the different Waitstaff (Slack, Web, Email) that deliver his meals to the customers.

The Prompt Engineering

category constitutes 20% of the Agentforce Specialist Exam, focusing on the creation, grounding, and management of templates that allow generative AI to interact with Salesforce data safely and consistently.

1. When to Use Prompt Builder

Prompt Builder is the appropriate tool when business requirements necessitate consistent, AI-generated content grounded in CRM data without requiring the user to manually type complex prompts.

- **Drafting Communications:** Use it for creating newsletters for trade shows or generating detailed product descriptions for marketing materials.
- **Summarisation:** It is ideal for generating guest summaries on contact record pages or providing a digest of sales order details.
- **Consistency:** It addresses concerns regarding the time-consuming nature of manual prompting and ensures consistency across sales and service teams.

2. User Roles and Permissions

Managing and executing prompt templates requires specific permission sets to ensure security and functional access.

- **Creating/Managing:** Users who need to create or edit templates, such as data scientists or admins, must be assigned the Prompt Template Manager permission set.
- **Executing Only:** For users (like sales reps) who should only run templates but not modify them, assign the Prompt Execute Template or Prompt Template User permission set.
- **Data Access:** Users will only see AI-generated outputs if they have permission to access the underlying fields used for grounding; otherwise, the draft may contain placeholders.
- **Generative AI Access:** Users must have the Generative AI User permission assigned to see features like the "sparkle" icon next to AI-enabled fields.

3. Considerations for Field Generation and Flex Types

The choice of template type depends on where the output is needed and the complexity of the inputs.

- **Field Generation:**

- Used to automatically populate a field (e.g., Description or AI Analysis) on a record page.
- **Critical Step:** After creating and activating the template, you must edit the Lightning page layout to associate the field with that specific prompt template.

- The org must be set to API version 59 or higher to enable these fields for generative AI.

- **Flex Templates:**

- Used when the prompt requires data from multiple unrelated objects (standard or custom) as inputs.
- They are versatile and can be called directly from Lightning Web Components (LWC).
- Ideal for complex scenarios like suggesting products from a large catalog.

4. Grounding Techniques

Grounding ensures the AI response is based on factual CRM data rather than general training.

- Record Snapshots: When using this feature, be aware that it automatically filters out empty data, such as fields without values.
 - Related Lists: To ground a template with a related list, the list must be present on the parent object's page layout for the prompt to retrieve data correctly.
 - Data Cloud Integration: Web activities tracked in Data Cloud can be used for grounding by adding them as an enrichment related list to a unified Contact record.
 - PII Security: When grounding with sensitive data, it is a best practice to mask sensitive fields and index only non-PII data.
5. Process for Creating, Activating, and Executing
- The lifecycle of a prompt template follows a specific technical workflow.
- Creation Process: Select the template type -> Develop the prompt in the workspace -> Select resources for dynamic CRM grounding -> Choose the LLM model -> Test and validate generated responses.
 - Activation: Once a prompt template version is activated, it is immutable; no further changes can be saved to that specific version.
 - Standard Templates: If a standard Salesforce template does not meet requirements, you should clone the existing template and modify the clone.
 - Execution via Flow: You can invoke a prompt template within a screen flow or template-triggered prompt flow to collect user inputs (like an order number) and display the AI summary.
6. Best Practices for Writing Effective Prompts
- Effective prompt design focuses on providing the LLM with clear structure and boundaries.
- Role-Playing: Always instruct the LLM to role-play as a specific character (e.g., "You are an experienced support agent") to provide better context.
 - Few-Shot Prompting: Include desired output examples within the instructions to ensure the LLM follows the correct format and avoids unnecessary conversational text.
 - Structured Format: Use clear sections for Role, Task, Context, Constraints, and Format.
 - Iterative Design: Prioritise clear, concise instructions and use iterative feedback to refine mismatches in generated information.

Analogy for Understanding: Think of Prompt Builder as a Professional Blueprint. The Template Type (Field Generation or Flex) is the style of the house. Grounding is the actual site survey and land data—without it, the architect (LLM) is just guessing based on generic houses they've seen before. Permissions are the security badges that determine who can view the plans versus who can actually change them. Lastly, Best Practices are the specific building codes that ensure the house is exactly five rooms (constraints) and painted blue (format).

The Data Cloud for Agentforce category represents 20% of the exam, focusing on how Data Cloud powers an agent's ability to retrieve and process information from various sources to provide accurate, grounded responses.

1. Agentforce Data Library and its Types
- The Data Library acts as a curated repository that allows an agent to provide answers based on verified documents rather than general knowledge.
- Key Considerations: Once a data source is chosen for a library, it cannot be changed later. Additionally, an agent can have only one data library assigned to it at a time.
 - Knowledge-Based Type: This uses Salesforce Knowledge articles. The identifying fields

help the engine locate the correct article, while the content fields enrich the AI's response with specific details.

- File Upload Type: This allows for the ingestion of proprietary documents (like PDFs). Once configured, the system indexes these files into Salesforce File Storage.
- Automatic Components: Saving a Data Library automatically creates a data stream, a search index, and a retriever within Data Cloud.

2. Improving Responses with Unstructured Data

Using unstructured data (like PDFs or HTML files) requires specific preparation to ensure the AI can "read" and retrieve the information efficiently.

- The Preparation Process: Handling unstructured data involves a cycle of loading, chunking, vectorising, and storing content so it is search-optimised for a vector database.
- Chunking Strategies: When uploading thousands of HTML knowledge files, use section-aware chunking to ensure the agent retrieves accurate responses quickly.
- Indexing for Accuracy: If product documentation changes frequently, the best approach to keep retrieval accurate is to rebuild the search index.
- Past Cases Scenario: To answer questions based on successfully resolved past cases, a Specialist should create an Unstructured Data Model Object (UDMO) based on the Case object and create an index on it.

3. Retrievers in Data Cloud

Retrievers perform contextual searches over indexed repositories to fetch the most relevant documents for grounding.

- Individual (Custom) Retrievers: These provide a primary advantage by allowing the configuration of filters, specific fields, and the number of results returned.
- Essential Configuration: When creating a custom retriever, you must select the search index, specify the associated Data Model Object (DMO) and data space, and define filters.
- Latency and Relevance: To ensure minimal latency and avoid irrelevant results, you should prioritise defining filters to limit the scope of each search efficiently. For example, a custom retriever can be filtered by "publication date" or "product line" to ensure only current information is used.

4. Search Types: Keyword, Vector, and Hybrid

The search type determines how the engine matches a user's query to the stored data.

- Keyword Search: Best used for exact term matching on structured fields, such as retrieving a specific policy clause number from a document library.
- Vector Search: Highly effective for handling misspellings (e.g., if a customer misspells a package name) because it looks for semantic similarity rather than character-for-character matches.
- Hybrid Search: This combines keyword precision with semantic flexibility. It is the preferred choice when queries mix specific technical terms with broader user intent, balancing precision and contextual disambiguation.
- Pre-Filtering: In a hybrid search scenario, you can apply pre-filtering within a custom retriever (e.g., only returning results where WarrantyStatus = 'Active') to ensure accuracy.

Analogy for Understanding: Think of Data Cloud as a Massive Library. The Data Library is a specific curated shelf you've set up for the agent. Chunking is like taking a 500-page book and breaking it into logical chapters so the agent doesn't have to read the whole book to find one fact. The Retriever is the Librarian who goes to find the book. If you use Keyword Search, the Librarian looks for an exact title match. If you use Vector Search, they look for books "about" a certain topic. Hybrid Search is when

the Librarian looks for books that are "about" your topic but also have a specific keyword on the cover.

The Development Lifecycle category represents 20% of the Salesforce Certified Agentforce Specialist Exam. This section focuses on the rigorous testing, safe deployment, and continuous monitoring required to maintain effective AI agents.

1. Testing an Agent using Agentforce Testing Center

The Agentforce Testing Center is the primary environment for validating agent reliability and accuracy before they reach customers.

- Bulk Testing with CSVs: To efficiently test a large and repeatable number of utterances, the Specialist should recommend creating a CSV file with test cases and uploading it to the Testing Center.,,
- Data Integrity and Environment: Running tests in a production environment risks modifying real CRM data,. Therefore, the best practice is to use the Testing Center only in sandbox environments replicated from production.,
- Defining Success Criteria: When configuring test cases, specifying the Expected Topic API Name as the expected output ensures that the tests accurately reflect the agent's intended functionality.,
- Coverage and Realism: To cover a broad range of user phrasing, the Testing Center can generate AI-generated synthetic test utterances based on natural language variations.,
- Performance Metrics: Testing should capture specific metrics such as response times, accuracy and relevance of answers, and resolution success to monitor overall correctness.,

2. Considerations for Deploying from Sandbox to Production

Moving an agent from a development environment to a live production org requires careful coordination of metadata and configurations.

- Deployment Tools: Specialists can deploy flows, Apex, and all agent-related items using either change sets or the Salesforce CLI/Metadata API.,
- Inclusion of Dependencies: A successful deployment must ensure all dependencies are included, configuration settings are aligned with production, and a plan for version management is in place.,
- Apex Code Coverage: Any Apex classes invoked by an Agent Action must have at least 75% code coverage from unit tests before they can be deployed to production.,,
- Activation Status: An agent is not automatically active after deployment; it must be manually activated in production, regardless of its status in the sandbox.,
- Metadata Components: When using the CLI, developers must include specific components like the genAiPlannerBundle in their package.xml to ensure topics and actions are transferred.,
- Custom LLM Names: If using a "Bring Your Own LLM" (BYOLLM) setup, ensure the name of the LLM matches in both sandbox and production to avoid deployment errors.,

3. Managing and Monitoring Agent Adoption

Once deployed, agents must be continuously monitored to ensure they are meeting business goals and providing accurate information.

- Usage and Usability: To monitor an agent's usability and the assignment of actions, Specialists should Run Agent Analytics.,,
- Troubleshooting Performance: If an agent is performing poorly or failing to trigger actions correctly, the Event Logs provide access to all user interactions, including errors and

incomplete plans,,,

- Analyzing Trends: To identify patterns in user queries and requests, the User Utterances dashboard or Agent Event Logs should be used,,,
 - Conversation Insights: Enabling the 'Enrich event logs with conversation data' setting allows Specialists to view full session data, including user inputs and agent responses, for the past 7 days,,,
 - Root Cause Analysis: If an agent provides unsatisfactory answers using grounded data (like PDFs or Knowledge), Specialists should examine the prompt instructions and the content of the chunks shown in the resolved prompt output.,,
-

Analogy for Understanding: Think of the Development Lifecycle as Launching a New Satellite. The Testing Center is the Virtual Simulator where you run thousands of flight scenarios (utterances) to see if it stays on course without crashing into the earth (production data). Deployment is the Launch Day Checklist—if you forget the fuel (dependencies) or your calculations aren't 75% verified (Apex coverage), it won't reach orbit. Finally, Monitoring is the Mission Control Room, where you watch the telemetry (analytics and event logs) to make sure the satellite is actually communicating with the ground correctly.

The Multi-Agent Interoperability category represents 5% of the Salesforce Certified Agentforce Specialist Exam. It focuses on the protocols and interfaces that allow agents to communicate with external tools, each other, and custom applications.

1. Model Context Protocol (MCP)

The purpose of the Model Context Protocol (MCP) is to provide a standardised way for agents to dynamically discover and consume external tools or APIs.

- Sub-Category Alignment: This sub-category addresses how an agent can extend its capabilities by reaching out to external specialised services.
- Key Use Cases:
 - External Tool Integration: An agent uses MCP to connect to a company's external product recommendation predictive model via APIs to generate dynamic suggestions.
 - Specialised Task Execution: A legal assistant agent uses MCP to dynamically find and use a document classification API to analyse specific case files.

2. Agent-to-Agent (A2A) Protocol

The Agent-to-Agent (A2A) protocol is designed to provide a standardised framework for cross-vendor agent discovery and communication.

- Sub-Category Alignment: This sub-category focuses on the "teamwork" aspect of AI, allowing different agents to interact.
- Purpose and Function:
 - Collaboration: A2A is the appropriate communication choice when multiple agents need to collaborate to solve a complex user request.
 - Interoperability: It ensures that agents, potentially built by different vendors, can find one another and exchange information within a governed framework.

3. Agent API

The Agent API provides a technical framework for external systems to interact directly with an Agentforce agent.

- Sub-Category Alignment: This focuses on custom integrations where standard Salesforce channels may not be used.

- Appropriate Scenarios for Use:
 - Custom Web Interfaces: If an organisation wants to integrate an agent into a custom website so customers can interact through a proprietary chat interface, the Agent API provides the necessary framework.
 - Secure Multi-Org Handoffs: When moving a user session between different Salesforce organisations, the Agent API is used to start the downstream agent's session and pass sensitive data, such as a verified customer ID, as a read-only context variable. This ensures data remains secure and persistent across handoffs without exposure to Large Language Model (LLM) modification.
-

Analogy for Understanding: Think of Multi-Agent Interoperability as a Global Business Summit.

- The Agent API is the Front Desk; it's how an outside visitor (a custom website) checks in and communicates with the building.
- MCP is the Contractor Directory; when an agent needs a specific job done that they can't do themselves (like a legal analysis), they use the directory to find and hire an external specialist.
- A2A Protocol is the Universal Translator and Badge System used by the summit attendees; it allows a Salesforce agent and a third-party agent to recognise each other's credentials and work together on a project.