

Agentic AI Use Case Worksheet

**A practical planning worksheet for defining, scoping, and measuring
an AI agent use case before design or implementation begins.**

1. Use Case Overview

The use case overview captures the basic story of the project: what the agent will be called, why it is needed, who will use it, and what value it should create. This section should be written in plain business language so that sponsors, product owners, process owners, technical teams, risk teams, and end users all understand the same problem.

1.1 Use Case / Project Name

The project name should be short, specific, and easy to remember. A good name should indicate the business process, the user group, or the outcome the AI agent supports. Avoid vague names such as “AI Assistant” or “Automation Bot” because they do not explain what the use case actually does.

- **Worksheet prompt:** What is the working name of this agentic AI use case?
- **Good format:** [Function or process] + [agent role] + [intended outcome].
- **Example:** “IT Incident Triage Agent” for an agent that reviews new incident tickets, classifies priority, suggests assignment groups, and drafts first-response updates.
- **Example:** “Supplier Risk Review Agent” for an agent that collects supplier information, checks risk indicators, and prepares a due diligence summary for review.

1.2 Problem You Want to Solve

Describe the current pain point clearly before describing the AI solution. The problem statement should explain what is difficult, slow, expensive, risky, inconsistent, or frustrating today. It should also describe the current process and why existing tools, people, or workflows are not enough.

- **Worksheet prompt:** What problem or opportunity are we addressing?
- **Current-state questions:** What happens today? Who performs the work? Which systems are involved? Where do delays, errors, or escalations occur?
- **Root-cause questions:** Is the problem caused by volume, missing information, inconsistent decisions, manual handoffs, unclear ownership, or slow response time?
- **Example problem statement:** “Service desk agents spend significant time reviewing incident descriptions, checking historical tickets, and deciding the correct assignment group. This delays first response and creates inconsistent prioritization for high-impact incidents.”

1.3 Target Users

Identify who will directly interact with the AI agent and who will benefit from its outputs. Target users may include frontline employees, managers, analysts, customers, support teams, auditors, or process owners. Be specific because different users need different levels of explanation, control, approval, and training.

- **Primary users:** People who use the agent directly, such as service desk analysts, HR recruiters, procurement analysts, project managers, or customer support representatives.
- **Secondary users:** People who consume the agent's outputs, such as managers, approvers, compliance teams, customers, or leadership teams.
- **Human reviewers:** People who validate recommendations before action is taken, especially when the agent affects risk, cost, customer experience, or compliance.
- **Example:** For an incident triage agent, primary users may be service desk analysts, secondary users may be resolver groups, and reviewers may be major incident managers for high-priority tickets.

1.4 Expected Outcome

The expected outcome explains what should improve after the agent is introduced. It should connect the AI capability to a business result, such as faster cycle time, better quality, lower operational effort, improved user experience, reduced risk, or more consistent decision-making.

- **Outcome type:** Efficiency, quality, compliance, customer experience, decision support, knowledge reuse, or risk reduction.
- **Expected business result:** State what will be better, faster, safer, cheaper, or more consistent.

- **Example:** “Reduce average incident triage time from 20 minutes to 5 minutes while improving assignment accuracy and ensuring high-risk incidents are escalated for human review.”
- **Tip:** Avoid outcomes that only describe activity, such as “use AI in the service desk.” Instead, describe a measurable improvement, such as “reduce manual ticket review effort by 50%.”

2. Define the Goal

The goal section turns the use case idea into a concrete objective. A strong goal gives the AI agent a clear purpose, defines the result the business expects, and sets measurable success criteria. This also helps determine the appropriate level of agent autonomy, the need for human approval, and the risks that must be controlled.

2.1 What should the AI agent accomplish?

Define the specific work the AI agent should perform. The description should include the inputs the agent receives, the decisions or recommendations it makes, the actions it may take, and the points where a human must review or approve the output. Agentic AI is strongest when the task can be described as a workflow, not just as a single answer-generation activity.

- **Inputs:** What information will the agent read? Examples include tickets, emails, customer records, policies, knowledge articles, project plans, logs, or supplier documents.
- **Reasoning task:** What judgment should the agent make? Examples include classify, prioritize, compare, summarize, recommend, route, draft, detect, or escalate.
- **Action:** What output should the agent produce? Examples include a recommendation, a completed form, a draft response, a risk summary, an updated status, or a next-step plan.

- **Human control:** Which actions require approval before execution?
- **Example:** “The incident triage agent should read new incident tickets, identify affected services, check urgency and impact, recommend priority, suggest the assignment group, draft the first response, and escalate possible major incidents to a human incident manager.”

2.2 What is the desired result?

The desired result should describe the business outcome after the agent completes its work successfully. This is different from the agent task. For example, “drafting a response” is a task, while “customers receive faster and more accurate updates” is a desired result.

- **Business result:** What improvement should the organization see?
- **User result:** How will the user’s work become easier, faster, or more reliable?
- **Customer or stakeholder result:** How will the experience improve for the people affected by the process?
- **Example:** “Service desk analysts can handle new tickets faster, resolver groups receive better-quality ticket assignments, and business users get earlier acknowledgement of high-impact incidents.”

2.3 How will you measure success?

Success measures should include outcome metrics, process metrics, quality metrics, and adoption metrics. The best measures compare the current baseline with the target improvement after implementation. For agentic AI, measurement should also include reliability and human-override indicators so the team can monitor whether the agent is producing trusted results.

Metric Category	Example Success Measure	Why It Matters
Efficiency	Average triage time reduced from 20 minutes to 5 minutes	Shows whether the agent saves operational time
Quality	Assignment accuracy improves from 75% to 90%	Shows whether the agent improves decision consistency
User Adoption	70% of eligible analysts use the agent weekly	Shows whether users find the agent useful in real work

Human Oversight	High-risk recommendations are reviewed before execution	Shows whether controls are working for sensitive decisions
Reliability	Agent output is accepted without correction in 80% of standard cases	Shows whether the agent is trusted for repeatable tasks

2.4 Is the goal specific and measurable?

A goal is specific and measurable when it clearly states what the agent will do, for whom, under what conditions, and how success will be verified. If the goal is too broad, the team may build a generic assistant instead of a useful agent. If the goal has no metrics, it will be difficult to prove business value or decide whether the proof of concept should scale.

- **Weak goal:** “Use AI to improve incident management.”
- **Stronger goal:** “Use an AI agent to classify and route new priority 3 and priority 4 incident tickets within 5 minutes, with at least 90% assignment accuracy and human review for suspected major incidents.”
- **Specific:** It identifies the process, ticket types, and agent responsibility.

- **Measurable:** It includes timing and accuracy targets.
- **Controlled:** It defines when human review is required.

3. Map the Current Workflow

Workflow mapping helps the team understand how the task is performed today before deciding what the AI agent should automate, assist, or monitor. This step prevents teams from designing an agent around assumptions. It also reveals where human judgment, system handoffs, repeated checks, and approval points currently exist.

3.1 How is the task handled today?

Describe the current process step by step, from the moment the work is triggered to the moment it is completed. Include the people involved, systems used, documents referenced, approvals required, and outputs created. The purpose is not to blame the current process but to make it visible enough to improve.

- **Worksheet prompt:** What event starts the task?
- **Process trigger:** A ticket is created, a request is submitted, an email is received, a supplier is onboarded, a report is due, or a customer raises an issue.
- **Current systems:** List the tools used today, such as ticketing systems, CRM platforms, ERP systems, shared folders, spreadsheets, knowledge bases, email, chat, or workflow tools.
- **Current output:** Identify what is produced at the end, such as a routed ticket, completed review, status update, risk summary, approval recommendation, or customer response.

- **Example:** In incident triage, a service desk analyst reads the ticket, checks the affected service, searches previous incidents, confirms urgency and impact, selects priority, assigns the resolver group, and sends an acknowledgement to the user.

3.2 What steps are handled by humans?

Identify the parts of the workflow that require people to read, interpret, decide, communicate, approve, or correct information. Human steps are not automatically bad; many are important control points. The goal is to determine which human activities should remain manual, which can be assisted by the agent, and which can be automated safely.

- **Information gathering:** Humans collect missing data from forms, emails, documents, logs, dashboards, or other teams.
- **Interpretation:** Humans read unstructured text, understand context, and decide what the request means.
- **Judgment:** Humans decide priority, eligibility, risk level, next action, or escalation path.
- **Communication:** Humans draft messages, ask clarifying questions, update stakeholders, or document decisions.
- **Approval:** Humans approve high-risk, high-cost, customer-impacting, regulatory, or policy-sensitive actions.

- **Example:** In supplier onboarding, procurement may gather supplier documents, risk teams may assess financial or compliance risk, legal may review contract clauses, and business owners may approve final onboarding.

3.3 Which steps require decision-making?

Decision-making steps are the points where the workflow can move in different directions based on rules, judgment, context, or risk. These are important for agentic AI because they show where the agent may need to classify, recommend, route, escalate, or ask for human review.

- **Rule-based decisions:** Decisions that follow clear rules, such as “if impact is high and urgency is high, assign priority 1.”
- **Context-based decisions:** Decisions that depend on interpretation, such as whether a customer complaint suggests dissatisfaction, churn risk, or a routine support issue.
- **Risk-based decisions:** Decisions that consider potential harm, compliance exposure, fraud indicators, safety, customer impact, or financial loss.
- **Escalation decisions:** Decisions about whether the task should be handled normally, sent to a specialist, or escalated to management.
- **Example:** In a project status reporting use case, the agent may decide whether a milestone is on track, at risk, or delayed based on schedule variance, blocker comments, dependency status, and recent updates.

3.4 Where do delays, repetition, or bottlenecks occur?

Delays and bottlenecks reveal where agentic AI may create the most value. Look for steps that consume time, depend on manual searching, require repeated copying between systems, or wait for someone to interpret information. Also identify bottlenecks caused by approvals, unclear ownership, missing data, or high work volume.

Bottleneck Type	What to Look For	Agent Opportunity
Manual search	Users repeatedly search policies, tickets, emails, logs, or documents	Agent retrieves and summarizes relevant information
Repetitive classification	Users repeatedly categorize similar requests or incidents	Agent suggests category, priority, owner, or next step
Missing information	Work cannot proceed because required fields or documents are incomplete	Agent checks completeness and asks targeted clarification questions

Approval delay	Requests wait for review because approvers lack context	Agent prepares a decision summary with key evidence and risks
System handoff	Users copy information between tools or teams	Agent prepares structured updates or recommended handoff notes

4. Identify Agent Decisions

This section defines the decisions the AI agent may support or make during the workflow. A decision may be simple, such as selecting a category, or complex, such as recommending escalation based on multiple signals. Clearly defining agent decisions helps the team determine data needs, guardrails, confidence thresholds, and human review points.

4.1 What decisions would the agent need to make?

List the decisions that the agent must make or recommend at each stage of the workflow. Separate decisions into low-risk, medium-risk, and high-risk categories. Low-risk decisions may be automated, while high-risk decisions should usually remain recommendations that require human approval.

- **Classification decisions:** What type of request, case, ticket, document, or customer issue is this?
- **Priority decisions:** How urgent or important is the work?
- **Routing decisions:** Which team, role, queue, or system should handle the next step?
- **Recommendation decisions:** What action should be taken next?
- **Escalation decisions:** Should this be escalated because of business impact, risk, uncertainty, or policy requirements?

- **Example:** An incident triage agent may decide the likely service affected, the ticket category, the priority, the assignment group, whether the issue resembles a known incident, and whether a major incident manager should review it.

4.2 What information would it need to make those decisions?

An agent can only make reliable decisions when it has access to the right information, understands the context, and can verify evidence. Identify the minimum data needed for each decision, where that data comes from, how current it must be, and whether the information is structured or unstructured.

- **Business rules:** Policies, decision trees, thresholds, standard operating procedures, escalation rules, service-level agreements, and approval matrices.
- **Operational data:** Ticket details, customer records, system status, logs, project updates, supplier profiles, previous cases, or transaction history.
- **Knowledge sources:** Knowledge articles, FAQs, playbooks, process guides, regulatory guidance, product documentation, or historical resolutions.
- **Context signals:** Business impact, user sentiment, affected service, location, time sensitivity, customer tier, risk category, or dependency status.
- **Example:** To decide incident priority, the agent may need the affected service, number of impacted users, business criticality, urgency, known outage status, related incidents, and SLA rules.

4.3 Are the decisions predictable or variable?

Some decisions follow stable rules and are predictable. Others depend on context, interpretation, exceptions, or evolving conditions. This distinction is important because predictable decisions can often be automated with clear guardrails, while variable decisions usually require confidence scoring, explanation, and human review.

Decision Type	Description	Example	Recommended Agent Role
Predictable	Decision follows clear rules and stable conditions	Route password reset requests to the identity support queue	Automate when confidence is high
Semi-predictable	Decision follows rules but requires context or evidence checking	Recommend incident priority based on urgency, impact, and affected service	Recommend with explanation and allow human override
Variable	Decision depends on judgment, ambiguity, exceptions, or changing risk	Decide whether a supplier risk finding requires legal escalation	Support human decision-making rather than act independently

High-risk	Decision may affect customers, compliance, finance, security, safety, or reputation	Approve access to sensitive data or reject a high-value transaction	Require human approval and audit trail
------------------	---	---	--

4.4 What should happen when the agent is uncertain?

Uncertainty handling is one of the most important design decisions for agentic AI. The agent should not guess when the information is incomplete, conflicting, outdated, or too risky. Instead, it should follow a defined fallback path, such as asking a clarifying question, retrieving more evidence, lowering autonomy, escalating to a human, or documenting why it could not proceed.

- **Ask for clarification:** If required fields or context are missing, the agent should ask a specific question instead of making assumptions.
- **Retrieve more evidence:** If the decision depends on supporting information, the agent should search approved sources before recommending an action.
- **Use confidence thresholds:** If confidence is below the agreed threshold, the agent should avoid taking action automatically.

- **Escalate to a human:** If the case is high risk, unusual, ambiguous, or policy-sensitive, the agent should route it to the correct reviewer.
- **Explain uncertainty:** The agent should state what information is missing, conflicting, or unclear.
- **Example:** If an incident ticket says “system is down” but does not identify the affected application or number of users impacted, the agent should ask for missing details or assign it for human triage instead of automatically setting a high priority.

5. Identify Required Tools

Agentic AI becomes useful when the agent can access the right tools, retrieve the right information, and perform the right actions safely. This section identifies the systems, APIs, data sources, files, search capabilities, functions, and internal applications the agent will need. A tool should only be included if it directly supports the workflow, improves decision quality, or enables a controlled action.

5.1 What tools or systems will the agent need?

List every tool or system the agent must use to complete the assigned workflow. This includes systems the agent reads from, systems it writes to, tools it uses to calculate or analyze information, and applications it uses to trigger actions. The list should distinguish between mandatory tools, optional tools, and tools that may be added later.

- **Read-only tools:** Systems the agent can view but not change, such as knowledge bases, reports, logs, policies, dashboards, or historical records.
- **Action tools:** Systems where the agent can create, update, route, assign, notify, or trigger a workflow.
- **Analysis tools:** Tools used for calculations, classification, comparison, data validation, or pattern detection.
- **Communication tools:** Email, chat, ticket comments, notification systems, collaboration platforms, or customer messaging tools.

- **Example:** An incident triage agent may need access to the ITSM platform, configuration management database, monitoring alerts, knowledge base, incident history, service ownership directory, and communication channels for updates.

5.2 APIs

APIs allow the agent to connect with business systems in a structured and controlled way. They are useful when the agent needs to retrieve live records, update workflow items, check status, create tasks, send notifications, or call another service. For each API, document what the agent can access, what actions it can perform, and what authentication or approval is required.

- **Worksheet prompt:** Which systems must the agent connect to through APIs?
- **Common API uses:** Retrieve records, update status, create tickets, assign owners, check inventory, validate customer data, or trigger approval workflows.
- **Access design:** Define whether the agent has read-only, write, update, or approval permissions.
- **Control point:** High-risk API actions should require confirmation, policy validation, or human approval before execution.
- **Example:** A service desk agent may call an ITSM API to retrieve ticket details, update category and priority, recommend an assignment group, and add a draft response note for human review.

5.3 Databases

Databases provide structured information that the agent may need for decisions, validation, reporting, or workflow execution. Examples include customer records, product catalogs, asset inventories, employee directories, supplier master data, incident history, transaction tables, and audit logs. Database access should be limited to the minimum data needed for the use case.

- **Worksheet prompt:** What structured data does the agent need to query?
- **Data examples:** Customer tier, service ownership, SLA category, risk rating, asset criticality, supplier status, entitlement rules, transaction history, or employee role data.
- **Data quality check:** Confirm whether the database is accurate, current, complete, and trusted for decision-making.
- **Privacy check:** Identify whether the database contains sensitive, personal, regulated, or confidential information.
- **Example:** A supplier risk review agent may query supplier master data, previous assessment results, contract status, country risk level, and payment history before preparing a risk summary.

5.4 Files and documents

Many agentic AI use cases depend on unstructured or semi-structured documents.

These may include policies, standard operating procedures, contracts, project plans, knowledge articles, meeting notes, risk assessments, requirement documents, invoices, training materials, or audit reports. The agent should be able to find, read, summarize, compare, and cite relevant content from approved document sources.

- **Worksheet prompt:** Which files or document libraries should the agent use?
- **Document types:** Policies, procedures, playbooks, forms, templates, knowledge articles, design documents, checklists, reports, or meeting summaries.
- **Version control:** Confirm which document is the approved source of truth and how the agent will avoid outdated drafts.
- **Retrieval need:** Decide whether the agent must search documents, extract specific fields, summarize sections, compare versions, or generate a draft output.
- **Example:** A project management agent may use the project charter, risk register, status reports, dependency log, and meeting notes to prepare a weekly executive update.

5.5 Web or search tools

Search tools help the agent locate information across approved sources. This may include internal enterprise search, knowledge bases, document repositories, web

search, vendor portals, public regulatory sites, or product documentation. Search tools are useful when the agent needs current information, supporting evidence, or cross-document context.

- **Worksheet prompt:** What search sources are allowed for this use case?
- **Internal search:** Enterprise documents, emails, chat history, tickets, knowledge bases, lessons learned, or approved repositories.
- **External search:** Public regulations, vendor documentation, market data, vulnerability advisories, standards, or support pages.
- **Search governance:** Define whether external web results are allowed, how sources are validated, and when the agent must cite or reference evidence.
- **Example:** A compliance support agent may search internal policies and approved regulatory websites to summarize obligations, but it should not rely on unverified blogs for final recommendations.

5.6 Python functions or internal applications

Some use cases require calculations, data transformation, file parsing, validation, scoring, or specialized business logic. In these cases, the agent may need to call Python functions, automation scripts, workflow engines, internal applications, or reusable service components. These tools should be well-tested, permission-controlled, and logged because they can directly affect business outputs.

- **Worksheet prompt:** Does the agent need to calculate, transform, validate, generate, or update anything using internal logic?
- **Python function examples:** Risk scoring, SLA breach prediction, duplicate detection, data cleansing, report generation, text extraction, or anomaly detection.
- **Internal application examples:** Workflow approval systems, project dashboards, HR systems, procurement platforms, finance tools, monitoring platforms, or case management applications.
- **Control point:** Any function that changes business data should include logging, permission checks, and rollback or correction procedures.
- **Example:** A finance reconciliation agent may call a Python function to compare invoice records against payment data, flag mismatches, and generate an exception report for review.

6. Choose the Agentic Workflow

The agentic workflow describes how the AI agent will approach the task. Some use cases only need tool use, while others need planning, reflection, or multiple specialist agents working together. The right workflow should match the complexity, risk, repeatability, and decision requirements of the use case.

6.1 Tool Use

Tool use means the agent can call approved systems or functions to retrieve information, perform calculations, update records, or trigger actions. This workflow is appropriate when the task is clear, the tools are reliable, and the agent's role is to connect information and execute controlled steps.

- **Best for:** Ticket updates, knowledge retrieval, database lookups, report generation, record validation, or notification workflows.
- **Design question:** Which tools can the agent call, and what permissions should each tool have?
- **Guardrail:** Separate read-only tool actions from actions that change data or notify stakeholders.
- **Example:** An incident triage agent uses a ticketing tool to read the incident, a knowledge base search tool to find similar issues, and an ITSM tool to draft recommended updates.

6.2 Planning

Planning means the agent breaks a larger task into smaller steps before acting. This is useful when the workflow has multiple stages, dependencies, decision points, or required outputs. Planning helps make the agent's work more transparent because users can see what the agent intends to do before it executes the next step.

- **Best for:** Research tasks, project updates, investigation workflows, compliance reviews, procurement assessments, and multi-step service requests.
- **Design question:** Should the agent show its plan to the user before executing?
- **Guardrail:** Require confirmation before the agent performs irreversible, sensitive, or externally visible actions.
- **Example:** A supplier risk review agent first plans to collect supplier documents, check sanctions and risk indicators, summarize findings, compare against policy thresholds, and prepare an approval recommendation.

6.3 Reflection

Reflection means the agent checks its own output before presenting or acting on it. It may verify whether the answer follows instructions, whether required data is missing, whether the recommendation is supported by evidence, and whether the output meets quality or compliance rules. Reflection is especially useful when errors could reduce trust or create operational risk.

- **Best for:** Draft responses, risk assessments, executive summaries, policy interpretation, customer communications, or decision recommendations.
- **Design question:** What should the agent check before sending or saving the output?
- **Reflection checklist:** Completeness, accuracy, policy alignment, evidence support, tone, formatting, missing assumptions, and escalation triggers.
- **Example:** Before sending a customer update, the agent checks whether the message includes the issue summary, current status, next update time, owner, and approved wording.

6.4 Multi-Agent Teamwork

Multi-agent teamwork uses more than one specialized agent to complete a broader workflow. Each agent has a defined role, such as research, analysis, drafting, validation, or approval preparation. This approach is useful when the work requires multiple types of expertise or when separating responsibilities reduces risk.

- **Best for:** Complex workflows where different tasks require different knowledge sources or decision logic.
- **Design question:** Which specialist agents are needed, and how will handoffs work between them?

- **Possible agent roles:** Intake agent, research agent, analysis agent, risk reviewer, drafting agent, quality checker, escalation agent, or coordinator agent.
- **Guardrail:** Define one coordinating agent or human owner so responsibility does not become fragmented.
- **Example:** A compliance review workflow may use one agent to collect evidence, one agent to compare evidence against policy, one agent to identify gaps, and one agent to draft the final review summary for human approval.

6.5 Other / Not Sure

If the team is not sure which workflow pattern is appropriate, start with the simplest design that can deliver measurable value safely. Avoid designing a complex multi-agent system when the use case only requires retrieval, summarization, or structured updates. Complexity should be added only when the task requires planning, specialist roles, evidence checking, repeated tool use, or layered review.

If Your Use Case Has...	Start With This Workflow	Reason
Simple lookups and updates	Tool Use	The agent mainly needs controlled access to systems or data

Multiple steps and dependencies	Planning	The agent should sequence work and explain the order of actions
High need for quality checking	Reflection	The agent should review completeness, evidence, and policy alignment
Different specialist tasks	Multi-Agent Teamwork	Specialized agents can separate research, analysis, drafting, and review
Unclear scope or early idea stage	Other / Not Sure	Begin with a low-risk prototype and refine the pattern after learning

7. Define Human Involvement

Human involvement defines how much autonomy the agent should have and where people must remain in control. This section helps the team prevent over-automation, protect sensitive decisions, and create a clear operating model for review, approval, escalation, and accountability.

7.1 What can the agent do independently?

Define the tasks the agent can complete without asking for human approval. These should normally be low-risk, repeatable, evidence-based, and reversible actions. The agent may act independently when the task follows clear rules, uses trusted data, and does not create significant financial, operational, legal, security, or customer impact.

- **Safe independent actions:** Retrieve information, summarize approved documents, classify low-risk requests, draft responses, prepare checklists, or populate non-sensitive fields.
- **Low-impact workflow actions:** Add internal notes, suggest next steps, create draft tasks, flag missing information, or recommend a queue without final assignment.
- **Reversible actions:** Actions that can be corrected easily, such as creating a draft record, generating a report, or adding a proposed status update.

- **Example:** An incident triage agent may independently read a new ticket, identify missing details, search the knowledge base, summarize similar incidents, and draft a recommended first response.

7.2 Where is human approval required?

Human approval is required when the agent's action could affect people, money, operations, security, compliance, or reputation. Approval points should be defined before deployment, not added after an incident occurs. Each approval point should specify who approves, what they review, what evidence they need, and how the approval is recorded.

- **Approval trigger:** The situation or threshold that requires a person to review the agent's recommendation.
- **Approver role:** The person or team responsible for approving, rejecting, or modifying the recommendation.
- **Evidence package:** The supporting information the agent must provide, such as data sources, reasoning summary, confidence level, and risk indicators.
- **Audit trail:** The record showing what the agent recommended, who approved it, when it was approved, and what action was taken.
- **Example:** A supplier risk agent may recommend approval for a low-risk supplier, but a human risk manager must approve suppliers with sanctions alerts, high financial exposure, or missing due diligence documents.

7.3 Which actions should always require approval?

Some actions should always require human approval, regardless of the agent's confidence level. These actions usually involve irreversible changes, sensitive data, legal or compliance obligations, financial movement, security access, public communication, or significant customer impact.

Action Type	Why Approval Is Needed	Example
Financial action	May create cost, payment, refund, loss, or budget impact	Approve invoice payment, issue refund, or change supplier payment terms
Security action	May expose systems, data, identities, or access privileges	Grant privileged access or disable a security control
Customer-impacting action	May affect service, eligibility, experience, or communication	Send final customer response or change customer account status
Compliance or legal action	May create regulatory, contractual, or audit exposure	Approve an exception to policy or interpret a contractual requirement

Operationally disruptive action	May affect service availability, production systems, or business continuity	Trigger production change, restart service, or escalate to major incident status
--	---	--

7.4 Who will review the agent's output?

Assign review responsibility based on the type of output and the risk level of the use case. Reviewers should understand the process, know how to challenge the agent's recommendation, and have authority to approve, reject, or request changes. For higher-risk workflows, more than one reviewer may be required.

- **Business process owner:** Reviews whether the agent output aligns with the business process and desired outcome.
- **Subject matter expert:** Reviews technical, domain-specific, legal, compliance, security, or operational accuracy.
- **Risk or compliance reviewer:** Reviews sensitive decisions, policy exceptions, regulatory exposure, and audit requirements.
- **End user or frontline reviewer:** Confirms whether the output is useful, understandable, and practical in daily work.

- **Example:** For a project status reporting agent, the project manager may review the update, the PMO lead may validate executive messaging, and the risk owner may review major risks before the report is shared.

8. Risk and Guardrails Check

The risk and guardrails check helps teams identify what could go wrong before the agent is released. Agentic AI can create value, but it can also introduce new risks if it uses poor data, takes the wrong action, accesses sensitive information, or operates without enough oversight. Guardrails should be practical, testable, and connected to the specific use case.

8.1 What could go wrong?

Start by listing failure modes in plain language. Consider incorrect outputs, missing context, outdated information, inappropriate actions, privacy concerns, security issues, bias, over-reliance by users, poor escalation, and lack of auditability. Each risk should be tied to a possible impact and a control that reduces the likelihood or harm.

- **Incorrect recommendation:** The agent classifies, prioritizes, routes, or summarizes incorrectly.
- **Missing information:** The agent acts without enough context or fails to ask for clarification.
- **Outdated source:** The agent relies on an old policy, obsolete knowledge article, or outdated operational data.
- **Unauthorized action:** The agent performs an action beyond its approved permission level.

- **Over-trust:** Users accept the agent's output without review, even when the case is unusual or high risk.
- **Example:** An incident triage agent may assign a business-critical outage to the wrong resolver group if the ticket description is vague and the agent does not verify affected service ownership.

8.2 What data will the agent access?

Document every data source the agent will read, retrieve, summarize, or use for decisions. Data access should be designed using the principle of least privilege: the agent should only access the minimum information required for the approved workflow. Classify data sensitivity and define how the agent will handle personal, confidential, regulated, or business-critical information.

- **Data inventory:** List systems, files, databases, APIs, logs, emails, chat history, documents, or knowledge bases used by the agent.
- **Data sensitivity:** Identify whether the data is public, internal, confidential, restricted, personal, regulated, or security-sensitive.
- **Data freshness:** Define how current the data must be for the agent's decision to be valid.
- **Data minimization:** Remove unnecessary fields, mask sensitive values, or limit access to approved records.

- **Example:** A HR support agent may need policy documents and employee lifecycle workflow rules, but it should not access compensation or health-related information unless the use case specifically requires it and approvals are in place.

8.3 What permissions does it need?

Permissions define what the agent is allowed to read, create, update, delete, send, approve, or trigger. Permission design should separate information access from action authority. The agent may need broad read access to perform useful retrieval, but action permissions should be narrow, logged, and reviewed regularly.

Permission Type	What It Allows	Recommended Control
Read-only	View records, documents, logs, or knowledge articles	Limit to approved sources and log access where required
Create draft	Create draft notes, recommendations, reports, or messages	Require human review before publishing or sending

Update records	Change fields such as status, category, assignment, or priority	Allow only for low-risk fields or require approval for sensitive fields
Trigger workflow	Start approvals, notifications, escalations, or automation steps	Use thresholds, confirmation prompts, and audit trails
Delete or irreversible action	Remove records, revoke access, cancel transactions, or block services	Avoid autonomous use; require explicit human approval

8.4 What actions could cause financial, operational, or security risks?

Identify actions that could cause harm if performed incorrectly, too early, too often, or without enough context. These actions should receive stronger guardrails, approval requirements, testing, monitoring, and rollback plans. The team should also define who is accountable when an agent-assisted action contributes to an incident or business issue.

- **Financial risks:** Incorrect payment, refund, discount, procurement approval, budget update, vendor change, or billing adjustment.

- **Operational risks:** Wrong routing, delayed escalation, incorrect priority, failed handoff, service disruption, or poor incident response.
- **Security risks:** Unauthorized access, exposed credentials, policy bypass, incorrect entitlement change, data leakage, or insecure automation.
- **Compliance risks:** Incorrect regulatory interpretation, missing evidence, incomplete audit trail, or unauthorized policy exception.
- **Reputation risks:** Poor customer communication, inaccurate executive update, inconsistent decision, or public-facing mistake.
- **Example:** A banking operations agent should not independently reverse a transaction, approve a high-value exception, or send a fraud-related customer notice without human approval.

8.5 What limits or guardrails should be applied?

Guardrails are the rules, controls, and limits that keep the agent operating safely. They should be explicit enough to test and monitor. Good guardrails do not simply say “be careful”; they define what the agent can do, what it cannot do, when it must ask for help, and how exceptions are handled.

- **Permission guardrails:** Limit the agent to approved systems, roles, tools, and action types.

- **Data guardrails:** Restrict sensitive data access, mask confidential values, and prevent unnecessary data exposure.
- **Decision guardrails:** Use confidence thresholds, escalation rules, and human approval for high-risk decisions.
- **Action guardrails:** Block irreversible, external-facing, financial, security, or customer-impacting actions unless approved.
- **Content guardrails:** Require approved tone, policy-aligned language, source references, and reviewer approval for formal communication.
- **Monitoring guardrails:** Track usage, errors, overrides, escalations, approval outcomes, and user feedback.
- **Example:** An incident triage agent may update low-risk categorization fields automatically, but it must escalate to a human before declaring a major incident, notifying executives, or triggering recovery actions.

9. Agent vs. Traditional Automation

This section helps the team decide whether the use case truly needs an AI agent or whether a simpler automation, workflow rule, script, or robotic process automation approach would be enough. Agentic AI is most appropriate when the task requires judgment, tool selection, flexible sequencing, context interpretation, or adaptation to changing conditions.

9.1 Do the steps change based on what happens?

If the workflow always follows the same sequence, traditional automation may be sufficient. If the next step depends on what the agent discovers, interprets, or receives from a tool, an agentic workflow may be more appropriate. Changing paths are a strong signal that the system needs reasoning or decision support rather than only fixed instructions.

- **Fixed workflow:** Every request follows the same steps in the same order.
- **Adaptive workflow:** The next step changes depending on missing information, risk level, user response, system status, or policy conditions.
- **Agent indicator:** The system must decide whether to search, ask a question, escalate, summarize, route, or take a different action based on context.

- **Example:** A password reset request may follow a fixed workflow, while incident triage may require different paths depending on affected service, business impact, outage status, and available evidence.

9.2 Does the system need to choose between different actions?

An AI agent is useful when the system must select the best action from multiple possible options. This may include choosing whether to classify, route, escalate, draft, notify, retrieve more information, or stop and ask for approval. If the action is always known in advance, a traditional rule or workflow may be simpler and safer.

- **Traditional automation fit:** “When form is submitted, create a ticket and send confirmation.”
- **Agentic fit:** “When request is submitted, interpret the request, identify missing information, choose the correct workflow path, draft a response, and escalate if risk is high.”
- **Decision examples:** Assign to team A or team B, ask for clarification or proceed, approve automatically or request human review, summarize or trigger investigation.
- **Example:** A customer support agent may decide whether to answer from a knowledge article, request account verification, escalate to billing, or send the case to a specialist.

9.3 Does it need to use multiple tools?

Agentic AI can be valuable when the task requires the system to choose and coordinate multiple tools. However, multiple tools alone do not automatically justify an AI agent. The key question is whether the system must decide which tool to use, when to use it, how to interpret the result, and what to do next.

- **Simple integration:** A fixed automation calls the same tools in the same order every time.
- **Agentic tool use:** The agent selects tools based on context, such as searching documents, checking a database, calling an API, or generating a draft.
- **Tool coordination need:** The agent must combine results from several tools to recommend the next action.
- **Example:** A project reporting agent may use a project plan, risk register, meeting notes, dependency tracker, and status dashboard before creating an executive update.

9.4 Could a fixed script handle the task instead?

Before choosing an AI agent, check whether a fixed script, workflow rule, form validation, dashboard, or robotic process automation can handle the task reliably. Traditional automation is often cheaper, easier to test, easier to govern, and more predictable. If the task is deterministic and does not require interpretation, a fixed automation may be the better solution.

- **Use traditional automation when:** Rules are stable, inputs are structured, exceptions are rare, and the same output is needed every time.
- **Use an AI agent when:** Inputs are unstructured, context changes, decisions require interpretation, and the workflow needs flexible next steps.
- **Hybrid option:** Use traditional automation for predictable steps and an agent for interpretation, summarization, or exception handling.
- **Example:** A script can send reminders for overdue tasks, but an agent may be useful when it must interpret meeting notes, identify blockers, update project risks, and draft stakeholder communications.

9.5 Does the potential benefit justify the added complexity?

Agentic systems can require additional design, security review, testing, monitoring, user training, and governance. The team should compare expected benefits against added complexity. A use case is stronger when the agent reduces significant manual effort, improves decision quality, increases speed, lowers risk, or enables work that is difficult to scale with traditional automation.

Decision Question	If Yes	If No
-------------------	--------	-------

Does the workflow need judgment or interpretation?	Consider an AI agent	Traditional automation may be enough
Do steps change based on context?	Agentic workflow may add value	Fixed workflow may be simpler
Are inputs unstructured or incomplete?	AI agent may help interpret and clarify	Rules or forms may be sufficient
Are risks high?	Use human approval and strong guardrails	More autonomy may be acceptable
Is the benefit measurable?	Proceed to feasibility check	Refine the business case before building

10. Feasibility Check

The feasibility check confirms whether the use case can realistically be designed, tested, secured, funded, and operated. A promising idea should not move directly to development until the team understands data availability, tool readiness, model capability, security requirements, cost, and testing approach.

10.1 Is the required data available?

Confirm whether the agent can access the data needed to make decisions and generate reliable outputs. The data should be available, accurate, current, complete, and legally usable for the intended purpose. If key data is missing or unreliable, the use case may need data cleanup, process changes, or a smaller pilot scope.

- **Availability:** Does the data exist in systems, databases, files, or approved repositories?
- **Quality:** Is the data accurate, complete, consistent, and current enough for decisions?
- **Access:** Can the agent access the data securely and with the correct permissions?
- **Usage rights:** Is the data allowed to be used for this AI-supported workflow?

- **Example:** An incident triage agent will be difficult to build if tickets do not contain affected service, impact, urgency, symptoms, or historical resolution notes.

10.2 Are the required tools available?

Check whether the required tools, APIs, applications, databases, search systems, or internal functions are available for integration. The team should also confirm whether the tools are stable, documented, secure, and supported by system owners.

- **Tool readiness:** Are APIs, connectors, search tools, or automation functions available?
- **Integration effort:** How much engineering work is needed to connect the agent to required systems?
- **System ownership:** Who approves access, monitors usage, and supports the integration?
- **Reliability:** Are the tools dependable enough for the use case's operating hours and service expectations?
- **Example:** A procurement agent may not be feasible if supplier records are spread across disconnected spreadsheets with no approved system interface.

10.3 Does the chosen model support the required capabilities?

Assess whether the selected model or platform can handle the required reasoning, tool use, context length, language support, retrieval, structured output, and safety controls. The model should be tested against realistic examples, not only simple demonstrations.

- **Capability fit:** Can the model classify, summarize, reason, compare, draft, retrieve, plan, or call tools as required?
- **Context handling:** Can it use enough information from documents, records, conversations, or logs?
- **Structured output:** Can it produce consistent formats such as JSON, forms, tables, reports, or ticket updates?
- **Reliability:** Does it perform consistently across normal, edge, and ambiguous cases?
- **Example:** A contract review agent may require strong long-document comprehension, clause comparison, source referencing, and escalation when contractual language is unclear.

10.4 Are security requirements understood?

Security requirements should be defined before the agent is connected to tools, systems, or sensitive information. The team should understand authentication, authorization,

data protection, logging, monitoring, retention, auditability, and incident response requirements. Security design should cover both what the agent can access and what it can do.

- **Identity and access:** How will the agent authenticate, and what identity will be used for actions?
- **Least privilege:** Does the agent have only the minimum permissions needed?
- **Logging and audit:** Are tool calls, recommendations, approvals, and actions recorded?
- **Data protection:** How will sensitive information be masked, stored, transmitted, and retained?
- **Example:** A security operations agent should not take containment actions unless approval rules, logging, rollback, and escalation procedures are clearly defined.

10.5 Is the expected cost acceptable?

Estimate the cost of building and operating the agent. Cost may include model usage, tool integrations, infrastructure, data preparation, security review, testing, monitoring, human review time, training, maintenance, and support. The use case should have a realistic benefit case that justifies these costs.

- **Build cost:** Discovery, design, engineering, integrations, prompt design, workflow configuration, and testing.
- **Run cost:** Model calls, API usage, compute, monitoring, storage, and support effort.
- **Governance cost:** Risk review, compliance review, audit logging, approval workflows, and ongoing control testing.
- **Human cost:** Reviewer effort, exception handling, training, and change management.
- **Example:** A high-volume triage use case may justify agentic AI if it saves significant analyst time, but a low-volume quarterly task may be better handled with a checklist or template.

10.6 Is a testing approach defined?

Testing should prove that the agent performs safely and reliably across normal cases, edge cases, ambiguous cases, and high-risk scenarios. Testing should include expected outputs, unacceptable outputs, tool-use behavior, permission boundaries, human approval flows, and fallback behavior when the agent is uncertain.

- **Test scenarios:** Include typical cases, missing-data cases, conflicting-data cases, high-risk cases, and policy-exception cases.

- **Expected outputs:** Define what the agent should recommend, ask, draft, update, or escalate.
- **Failure tests:** Confirm that the agent refuses, escalates, or asks for clarification when it should not proceed.
- **Reviewer validation:** Have subject matter experts compare agent outputs against accepted decisions or historical cases.
- **Example:** An incident triage agent should be tested on routine tickets, vague outage reports, duplicate incidents, security-related incidents, and possible major incidents.

11. Final Decision

The final decision summarizes whether the idea should move forward as an AI agent, be implemented as traditional automation, or return for more investigation. The decision should be based on value, workflow complexity, data readiness, tool availability, risk level, governance needs, and testing feasibility.

11.1 Good candidate for an AI agent

Select this option when the use case requires interpretation, flexible decision-making, tool use, planning, reflection, or human-in-the-loop judgment. A good AI agent candidate should have a clear business objective, accessible data, defined tools, measurable success criteria, manageable risk, and a realistic testing approach.

- **Strong indicators:** Unstructured inputs, changing workflow paths, multiple tools, contextual decisions, and measurable improvement potential.
- **Readiness indicators:** Required data and tools are available, guardrails are defined, reviewers are assigned, and testing scenarios are prepared.
- **Example decision:** “Proceed with an AI agent prototype for incident triage because the task requires interpretation, tool use, prioritization, routing, and human approval for high-risk cases.”

11.2 Better suited to traditional automation

Select this option when the workflow is fixed, the rules are stable, the inputs are structured, and the required output is predictable. Traditional automation may be faster, cheaper, easier to govern, and safer when the task does not require complex reasoning or flexible tool selection.

- **Strong indicators:** Clear rules, fixed sequence, structured data, limited exceptions, and no need for interpretation.
- **Better-fit options:** Workflow automation, scripts, RPA, form validation, rule engines, dashboards, or scheduled reports.
- **Example decision:** “Use traditional automation for reminder emails because the trigger, recipient, timing, and message format are predictable.”

11.3 Needs more investigation

Select this option when the use case has potential but important information is missing. The team may need to clarify the workflow, clean data, confirm tool access, define approvals, estimate cost, assess security, or create test cases before making a build decision.

- **Common gaps:** Unclear business value, missing data, unavailable tools, undefined ownership, unresolved security concerns, or no testing approach.

- **Next step:** Run a short discovery sprint, interview users, map the process, validate data sources, and test a small sample manually.
- **Example decision:** “Investigate further before building because required supplier data is incomplete and approval ownership has not been confirmed.”

Final Decision Option	Use When	Recommended Next Step
Good candidate for an AI agent	The task requires judgment, adaptation, tool use, and measurable business value	Build a controlled prototype with human review and guardrails
Better suited to traditional automation	The process is fixed, rules are stable, and outputs are predictable	Implement as workflow automation, script, RPA, or rules engine
Needs more investigation	Data, tools, risks, ownership, or value are not clear enough	Run discovery, validate assumptions, and revisit the decision

Conclusion

An Agentic AI Use Case Worksheet helps teams make better decisions before investing in design and implementation. It ensures that the use case has a clear problem, measurable goal, mapped workflow, defined decisions, required tools, appropriate human involvement, practical guardrails, and a realistic feasibility assessment. The strongest candidates for AI agents are not simply tasks where AI sounds attractive; they are workflows where adaptive reasoning, tool use, contextual decision-making, and controlled autonomy create measurable value.

Use this worksheet as a living document. Revisit it after discovery interviews, prototype testing, risk review, user feedback, and operational monitoring. As the use case matures, the team should refine success measures, guardrails, tool access, human approval points, and final decision criteria so the agent remains useful, safe, and aligned with business outcomes.

AGENTIC AI DEVELOPER CERTIFICATION

**AGENTIC AI DEVELOPER
CERTIFICATION BASED ON REAL-
WORLD AGENT FRAMEWORKS TO
DESIGN, BUILD, AND DEPLOY
AUTONOMOUS AI SYSTEMS.**



ABOUT GSDC CERTIFICATION



LIFETIME VALIDITY

GSDC Certification is an globally accredited certification with lifetime validity.



EBOOK

Extensive and exclusive Ebook created by world's experts to help you with understanding core concepts.



CREATED BY EXPERTS

GSDC certifications are created and authored by world's leading experts in the field.



LEARNING MATERIALS

Get access to learning materials such as videos, ebooks, templates, and practice exams, which will help you clear the certification exam.

LEARNING OBJECTIVE

- Improve problem-solving with real-world examples
- Accelerate learning through hands-on resources
- Prepare for advanced AI developer roles
- Boost credibility among AI employers

Enroll now with the
code **LEARN20** To
avail **20%** discount

Enroll Now



www.gsdccouncil.org