Google Cloud

Enabling generative AI value: Creating an evaluation framework for your organization

September 17, 2024

Table of contents

Introduction

This document provides a framework for evaluating generative AI that can help organizations deploy more generative AI use cases, develop faster, and manage risks more effectively. **To drive successful generative AI implementations, organizations can tailor their approach by adopting a few simple evaluation practices and evolving their evaluations as capabilities mature.** This document describes tasks and recommendations that can help your organization evaluate and derive value from generative AI solutions.

The prevailing approach to generative AI evaluation

Generative AI has the potential to boost global GDP by 7%—nearly $7 trillion—and increase productivity growth by 1.5% over the next decade[1]. Given this promise, organizations are investing in improving workflow efficiency, enhancing quality, and launching innovative services. Even with this potential, however, realizing value requires tailoring solutions to specific use cases. To create these solutions, organizations must develop iteratively, as outlined in "A platform-centric approach to scaling generative AI." In this development process, the evaluation stages provide early and continuous feedback, which enables enhancements and increasing confidence that a solution is production-ready. Despite its critical role in generative AI development, implementing a practical evaluation framework can be complicated because of one or more of the following factors:

- Lack of high-quality evaluation data
- Limited technical knowledge of evaluations
- Insufficient time to tailor evaluations to a use case
- Too many options for models, orchestration frameworks, and architecture selections
- Difficulty in defining success and associated hill climbing

The good news, however, is that organizations have already adopted practices and cultures with their DevOps or MLOps approaches. Similarly, you can ease the evaluation burden and unlock value from your generative AI investments by tailoring an evaluation approach to your needs.

[1] Goldman Sachs: Generative AI could raise global GDP by 7%
[2] Google: What is Retrieval Augmented Generation
[3] Measuring Massive Multitask Language Understanding
[4] SWE-agent: Agent-Computer Interfaces Enable Automated Software Engineering, Yang et al 2024
[5] Language Models are Few-Shot Learners, Brown et al 2020

The following figure shows the end-to-end evaluation process and its five steps: discover, design, develop, deploy, and operate.

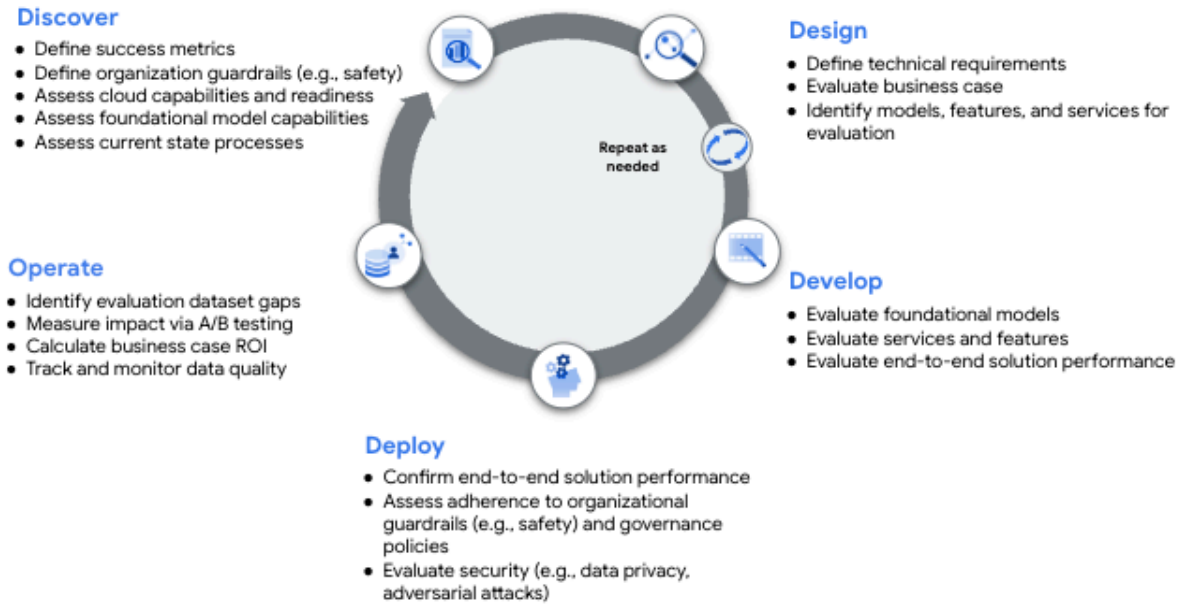

Figure 1: End-to-end evaluation steps

These five steps form a flexible evaluation framework that can be adjusted to meet your requirements, and repeated as needed. For example, as organizations start investing in generative AI, they often depend on their early evaluations of AI solutions to drive decisions. An organization's first wave of investment in generative AI typically focuses on use cases with an easy-to-measure impact, like customer assistants[2]. This measurement focus streamlines the business case for enterprise-wide adoption. Thus, it is important to adopt a user-centric evaluation approach that provides clear before-and-after comparisons.

Throughout development, AI solution builders—developers, product managers, AI engineers, IT managers, and others—must evaluate models, services, design choices, and overall solution performance. These evaluations typically consist of model performance, end-to-end performance, safety, latency, scalability, and cost. For instance, augmenting an application with a new retrieval-augmented generation (RAG)[2] pipeline requires evaluations of latency, service cost, model outputs, and retrieval relevance from the vector database or knowledge graphs. However, **evaluations require high-quality data, domain expertise, technical expertise, time, and the ability to automate the comparison of multiple architectural options. So, some organizations do not adopt an evaluation framework**—28% of enterprises don't evaluate at all[3]**.**

[2] Google Cloud: 101 real-world gen AI use cases from the world's leading organizations
[3] Scale AI: Zeitgeist: 2024 AI Readiness Report

Other organizations adopt evaluations but often focus on specific aspects, like model performance, using heuristic tests to assess this performance, or considering general-purpose metrics (such as reasoning abilities). However, these general-purpose metrics, such as MMLU[3] or GLUE[4], might lack the coverage of relevant scenarios and sensitivity to the nuanced demands of a use case. For example, on the general benchmark SWE-bench, GPT-4 solves 12.47% of bugs, whereas Claude Opus solves 10.46%. However, for astropy—an astrophysics-focused Python package—Claude solves 33.33% of bugs, and GPT-4 solves 16.66% of bugs. If solution builders focus on general coding ability instead of evaluating only astropy patching, their solution might be less effective on astropy tasks. Solution builders need to consider the entire solution's performance, of which model performance is only a dimension.

Mature AI organizations do more rigorous evaluations. However, unlike their highly automated MLOps and DevOps practices, their generative AI evaluations are largely manual, which result in slow and resource-intensive evaluations. As a result, these organizations struggle to scale their evaluation frameworks. Highlighting this point, an executive at a large health insurer stated, "We do not have a standard evaluation process or metrics, making it hard to compare and govern models. As a result, we find ourselves in lengthy review cycles that hinder our agility." The lack of structure in the evaluation process can burden solution builders and slow the adoption of new use cases.

In practice, evaluations are executed with varying rigor and consistency. For instance, in human-driven evaluations, some individuals might prefer more verbose answers, which can lead to inconsistent evaluations. Highlighting another issue, a CISO at a large B2B software company said, "Our generative AI applications are changing so rapidly that things we approved six months ago would not be approved today, but they are already in production. Since we lack an ongoing review process, we must accept the heightened security risk." To begin, this whitepaper explains how evaluation unlocks value, how evaluation can evolve with your capabilities, and how you can start with evaluations.

The importance of evaluation on the path to value

Although generative AI can create outsized value for organizations, realizing that potential requires business leaders and solution builders to come together to build with the outcome in mind. As outlined in "A Platform-centric Approach to Scaling Generative AI," evaluation creates a feedback loop that lets developers and AI engineers enhance a solution. In this feedback loop, solution builders assess model performance and maximize ROI by using high-quality ground-truth data—data that reflects common inputs in production and corresponding ideal outputs—and a series of tests.

Starting this feedback loop requires building a high-quality ground-truth dataset that has multiple examples of ideal outputs for a given input. Even at the earliest stage of development (like a pilot), assembling these datasets from real-world or analogous data sources creates a tool that can be used repeatedly to measure success. **Without this tool, functional tests are more like vibe checks because they rely on predicted inputs that might not reflect production scenarios.** As a result, these evaluation datasets increase confidence in assessments and in the solution, which enables organizations to recognize success.

With a high-quality dataset, solution builders can test a solution on dimensions like safety, fairness, and accuracy. By testing these solutions with this data, gaps in performance can be exposed and addressed iteratively. An executive at a large government contractor emphasized this point, saying, "As part of our agile development process, domain experts evaluate which model meets our performance characteristics, enabling our development team to refine the solution further." By using human evaluators, such as experienced federal employees, this executive's team can solicit feedback. The team can then adjust the design and functionality of the solution to optimize its ability to produce accurate outputs across various inputs and contexts.

Performance is critical for generative AI applications because even a few inaccurate results can make a solution untrustworthy and decrease value. Consider an AI legal assistant: if the assistant incorrectly cites case law, an entire brief and its underlying workflow are called into question. Engineers can proactively identify these situations by evaluating and changing model parameters (like temperature[4]) or adding components (like function calls[5] or check grounding[6]). **Similarly, generative AI solutions must produce safe outputs that comply with organizational policies and regulations.** For instance, a retail chatbot should never answer medical or legal questions. To prevent this risk, guardrails must be assessed by ensuring that safety filters catch out-of-scope inputs, engineered system prompts return on-topic content, and malicious actors cannot elicit damaging responses.

As generative AI solutions scale, higher-quality solutions tend to have higher costs because of growing system complexity. For example, consider a solution designed to assist with long-horizon tasks (for example, creating an investment thesis from hundreds of documents) with nuanced analyses. This solution might require hundreds of API calls, but it can also drive an order-of-magnitude increase in human output. Builders need to ensure profitability by estimating the costs and benefits of production-grade solutions. When done well, product and IT managers can establish a clear path to long-term utility and cost-effectiveness.

---

[4] Temperature
[5] Function calls
[6] Check grounding: A method of fact-checking model outputs to ensure outputs are factual and represent the underlying data

Finally, solutions must be flexible enough to accommodate changing requirements (such as longer context windows or multimodality), **data distributions** (such as input data drift)**, and model upgrades** (such as new releases) **that improve performance and cost-effectiveness.** As such, AI engineers and developers need repeatable evaluation methodologies to ensure that both old and new features are working as intended—the generative AI version of regression testing.

By rigorously evaluating generative AI solutions against these four dimensions (performance, cost, scale, and safety), developers can establish a solid foundation for ongoing and repeatable ROI from generative AI.

Tailoring evaluation frameworks to your needs

Evaluation is a resource-intensive process that yields critical feedback, and this feedback enables solution builders to develop production-quality applications. However, it is important to adapt the rigor of evaluation to the specific use case. For example, a comprehensive evaluation of a customer FAQ chatbot would typically include compiling a dataset of customer inquiries and their correct responses, testing the chatbot's summarization and retrieval capabilities, conducting adversarial tests to confirm the effectiveness of guardrails, and assessing overall customer satisfaction with the provided responses. These steps are thorough, but they require significant amounts of expertise and time. In contrast, using the same amount of expertise and time to evaluate a generative contract-writing assistant might be more intensive than needed.

To determine the appropriate level of evaluation rigor, product and IT managers should consider the following factors:

- **Frequency:** Consider the likelihood of your solution behaving in an unintended way, such as producing incorrect outputs, leaking data, or generating inappropriate content. Solutions with extensive customization, like fine-tuning or adding adapters, are typically higher risk and require more thorough evaluations.

- **Risk:** Determine the potential outcomes if the solution malfunctions. High-risk use cases, which typically influence critical decisions in sectors like finance or healthcare, demand rigorous evaluation. Conversely, solutions with lower impact, such as ad-copywriting assistants, might require less stringent scrutiny.

- **Application type:** The solution's deployment surface also affects its evaluation. Solutions that naturally have a human-in-the-loop (like ad-copywriting assistants) require less rigorous evaluation because humans still influence the final output, compared to autonomous solutions (like customer support assistants).

Product managers can determine the appropriate level of rigor by carefully considering these factors. By using a prioritized approach to evaluation, like the approach suggested by the EU AI Act[7], builders can minimize the burden of evaluation while ensuring that high-risk solutions are appropriately tested and delivered quickly. This strategy accelerates feature deployment and also enhances safety and effectiveness across varying levels of risk.

The evaluation journey

After solution builders have defined success, organizations' evaluation datasets and capabilities should at least parallel the maturity of their solutions. This growth should occur in phases: pilot, production, and scale. In the pilot phase, AI solution builders can use lightweight evaluation frameworks that use smaller evaluation datasets, end-to-end human evaluations, and general-purpose and public benchmarks, which are used to down-select models for consideration. As they transition use cases to production, solution builders should conduct more thorough evaluations that involve a mix of human and computational tests. Finally, as solutions scale, organizations should standardize the evaluation process and integrate evaluation into governance processes, like AI go-live reviews.

Building and evaluating a contract assistant: A practical example

To illustrate the evaluation journey, consider a procurement team that is overwhelmed with managing requests to track and analyze previous purchasing agreements, missed obligations, and savings opportunities from renegotiations and supplier relations. In response, the team developed a strategic roadmap that involves a generative AI application designed to enhance responsiveness, increase efficiency, and provide flexible support. The roadmap has three phases:

1. **Pilot:** Use a preselected model to build a chatbot to answer frequently asked questions, provide immediate responses to common inquiries, and deflect inquiries from the procurement team
2. **Production:** Customize interactions with individuals, offer real-time support, and create personalized how-tos.
3. **Scale:** Automate end-to-end procurement workflows, such as managing tickets, creating requests, and collecting information for approvals.

[7] EU Artificial Intelligence Act

Before the development process, the team defined success for each phase of the solution and identified the following requirements accordingly:

- **Technical requirements:**
  - **High throughput:** Handle up to 100 requests per minute
  - **Integrations:** Integrate with existing contract-management software and e-procurement software
  - **Latency:** Return content in under five seconds and first subtoken in under two seconds
  - **Model performance:** Answer questions with 95% accuracy
  - **Data residency and compliance:** Comply with multinational laws by storing data in specific localities as needed

- **Business requirements:**
  - **Cost:** Cap costs per interaction at 10 cents
  - **Translation accuracy:** Create translations that convey the nuances of contractual terms in different language than the original contract
  - **Format:** Present information in bullets
  - **Length:** Generate initial translations that should be less than 2,000 characters
  - **Chat length:** Generate follow-up questions and answers that should be less than 1,000 characters
  - **Style:** Use a casual business style that uses conversational language and simple sentences; do not provide answers in legalistic terms
  - **UX Quality**: Provide users with a multiturn chat experience that suggests follow-up answers
  - **Performance Metrics:** Target near-perfect uptime to ensure constant availability

The following figure shows the different steps in the pilot, production, and scale phases of the evaluation process for the contract assistant:

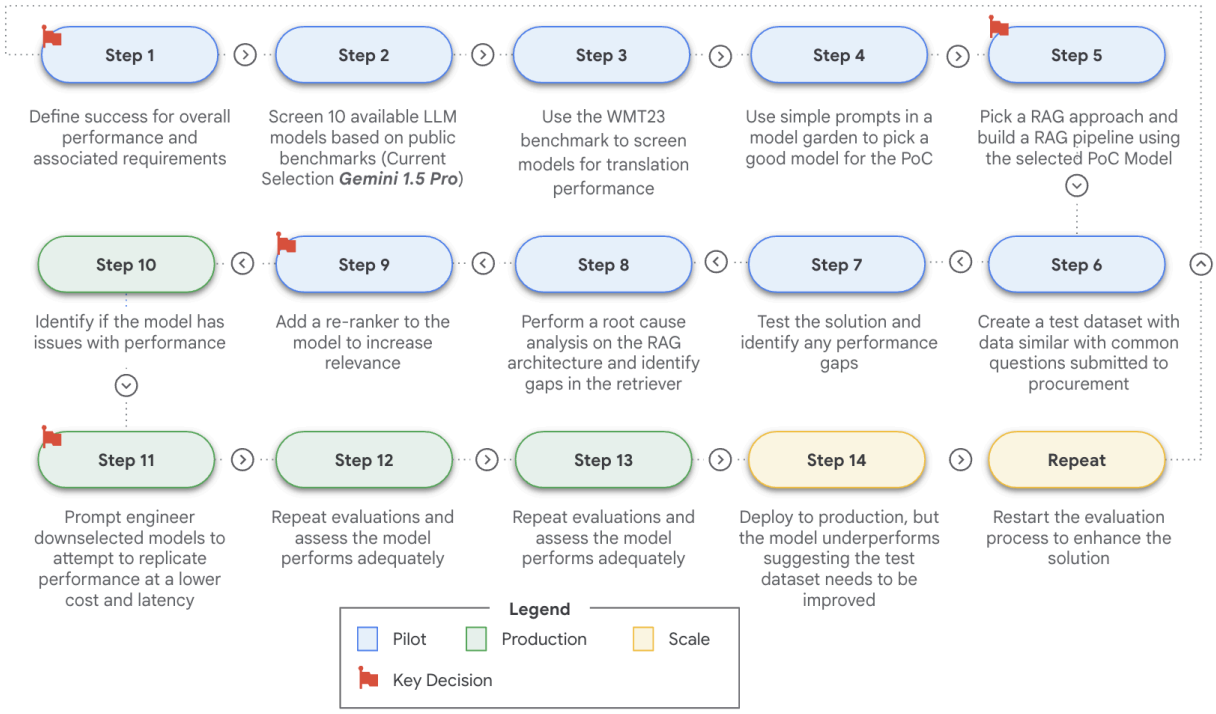

Figure 2: Step-by-step evaluation of contract assistant

This evaluation process involves multiple steps that the team must perform and key decisions that they must make throughout the pilot, production, and scale phases. The following sections describe the evaluation steps for these phases.

Evaluations in the pilot phase

The first step in adopting generative AI often involves a few pilots, where the primary goal is to validate hypotheses and generate learnings. This emphasis might encourage solution builders to use a frontier model (like Gemini 1.5 Pro) that maximizes performance over a more cost-effective model (like Gemini 1.5 Flash). **Although organizations might use state-of-the-art (SOTA) models, it is crucial to consider performance for the specific use case.**

To begin, organizations can use public benchmarks to quickly assess a model's potential. When organizations select metrics, asking questions about a model's role in the use case is important. These questions could include the following:

- Does the model reason across documents?

- Does it answer basic questions?
- Does it interact with multiple types of data?

After solution builders confirm the model's efficacy, they can compare performance across models by referencing public benchmarks that are underpinned by standard test datasets. Some of these benchmarks and tests include Big-Bench Hard[8] for reasoning, MMMU[9] for multimodal modeling, and WMT23[10] for language translation. Although these benchmarks provide information on model capabilities, they often measure performance on unrelated tasks (like academic multiple-choice questions) and do not fully represent production workloads. However, these metrics can serve as pragmatic filters for initial screening, and they offer a lightweight evaluation framework for selecting a model.

These benchmarks can provide general, directional information, but they are not a replacement for task-based evaluations on real-world data. Task-based evaluations look at a single step, such as generating a summary of a large document. Though these task-based evaluations are time-consuming, evaluating a solution's performance on a couple of tasks during the pilot phase (such as summarization and retrieval) can provide quick feedback on its performance. For each of these evaluations, AI engineers need to check the quality of their evaluation dataset and select a series of metrics (like accuracy or coherence) that have more straightforward definitions for criteria. By capturing this feedback throughout the development process, AI engineers and developers can track improvements in performance and identify when to deploy to production. Before solution builders deploy a pilot, they should partner with domain experts to conduct an end-to-end evaluation.

In the contract assistant example, the team needed to select a model quickly because of their short pilot timeline. They shortlisted five models based on the availability in their cloud environment and ease of access. Because the team's roadmap called for answering common contracting questions about contracts written in multiple languages (such as English or Chinese), they created a RAG application. To choose a model, they looked at the WMT23 benchmark to pick the three top-performing models on language translation. Though this assessment did not give them definitive comparisons for their use case, it suggested potential top performers. Before they compared each model head-to-head, the team compiled common questions and answers from the procurement team. They then compared models by using these questions and answers with some simple prompt engineering. To assess the models' accuracy, the team used their evaluation dataset to test accuracy by comparing the ideal translated summary—in terms of style and length—to the generated response. Additionally, they tested a few common questions collected from procurement for the multiturn chat experience and measured the coherence across responses. Based on these evaluations, the team selected

[8] Challenging BIG-Bench Tasks and whether Chain-of-Thought can Solve them
[9] A Massive Multi-discipline Multimodal Understanding and Reasoning Benchmark for Expert AGI
[10] WMT23

Gemini Pro 1.5 for its performance and native integration with their environment. Additionally, they consulted multiple experts to conduct a human evaluation to validate the translations, summaries, and response style. By working with multiple experts, the team reduced the potential for bias in the evaluations. From this process, the team found that the short answers needed additional detail to be effective.

To identify the underlying issue, they performed a root-cause analysis on each component of the RAG architecture, including the embeddings, retriever, and LLM generation. The following figure shows a high-level view of the chatbot's architecture and the various evaluation steps that the team performed on each component:

1. **RAG service:** Evaluate RAG retrieval for relevance and response time
2. **Internal and external data sources:** Conduct integration tests and check for data quality
3. **Gemini 1.5 Pro:** Assess model capabilities by using public benchmarks and manual task specific testing
4. **Contract assistant solution:** Assess solution performance by using human evaluators

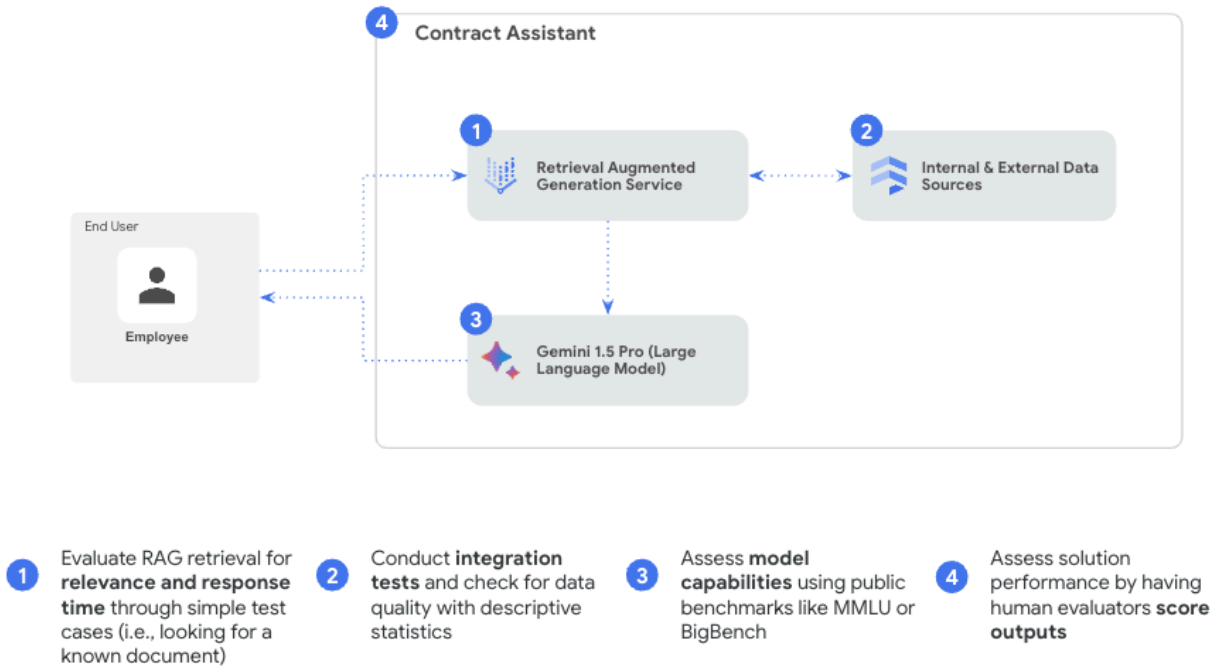

Figure 3: High-level architecture and evaluation steps for the example contract-FAQ chatbot

To analyze the embeddings, the team visualized a series of outputs to check whether there was enough differentiation and found that the model was separating documents properly. Next, they checked the retriever and realized that it was not returning enough relevant documents. Finally,

they checked the generation process with a human evaluation, which showed that the LLM and prompts were functioning properly.

From this evaluation, **the procurement team recognized that the document retriever could be improved, so they incorporated a re-ranker service to reorder the documents.** The team's experience illustrates the importance of conducting both a brief screening for a model with public benchmarks and human evaluations of the final solution.

Evaluations in the production phase

As solution builders transition from pilot to production, the rigor of evaluations should increase. Before evaluating a production use case, IT and product managers should determine the appropriate level of rigor as described in section 2, The importance of evaluation on the path to value. Specifically, they should consider private benchmarks[10] to accurately determine performance for use cases that require a higher level of rigor for evaluation. Initially, product managers and AI engineers should decompose the use case into tasks; for instance, an IT support chatbot would have **retrieval and summarization tasks.** However, although each task [11] contributes to the application's functionality, some can have a more significant impact. For instance, in a medical chatbot, generating accurate summaries is more critical than replicating the tone of a physician, so task analyses might prioritize summarization.

After tasks are prioritized, AI engineers can create a dataset for each task and select a series of evaluation metrics, including computational metrics (like accuracy) and human-driven metrics (like expert ratings). Creating an evaluation dataset typically requires a bottom-up collection of inputs and outputs similar to what users might expect in production. **For instance, to evaluate the performance of a customer support agent that is augmenting current call center operations, an organization can create realistic test scenarios by combining actual customer queries, knowledge articles, and successful agent responses. The test scenarios include straightforward inputs and their matching outputs. The test scenarios can also include inputs and outputs that include typical human errors, such as misspellings and misunderstandings.** This approach provides the solution with a more realistic test, which can enrich evaluation and feedback, and also create a reliable path to scale.

To build test datasets, consider the following guardrails:

[11] Private benchmarks: Internally developed dimensions used to assess the performance of a system for specific use cases

- **Identify common queries:** Use techniques like k-means[12] or hierarchical clustering[13] to create a distribution of queries by similarity. This process helps identify common query sets.
- **Sample from each distribution:** Select a representative series of queries from each cluster to provide a robust test without exhaustive effort.
- **Cleanse the data:** Cleanse the data by removing extraneous information, such as incorrect suggestions from a customer support agent. However, this data should not be entirely cleansed; it should include misspellings and other minor complications to ensure that the solution is robust across various inputs.
- **Annotate the data:** Add labels to the data, such as response quality ratings or classifications, to improve feedback for fine-tuning or prompt engineering.

Although this type of data is ideal for testing, it might take substantial effort to obtain. Therefore, test data needs to be sourced from related examples and requires additional expert validation. After solutions are deployed to production environments, continuously capturing common data points and identifying edge cases can help create a rich evaluation dataset for future enhancements. After the solution builders define metrics and create datasets, they can run multiple metrics with the same input-output pairs because most metrics will use the same datasets for calculations. Capturing this data enhances future evaluations, but significant data cleansing and annotating are required to see value gains. However, these choices are resource intensive and should be evaluated as such.

An executive at a large health insurer illustrated this point by stating, “Through multiple years of AI and now generative AI investment, we have built up large, annotated clinical datasets. As we adopt generative AI, these are used to support evaluations, which help us fine-tune our retrievers and guide our prompting.”

High-quality data improves solutions by providing feedback from real-world data. This feedback can be used to select a model, enhance prompts, or drive fine-tuning decisions. Similarly, proprietary datasets might be used to create private benchmarks or task-based metrics for actions like summarization. With this data, AI engineers can use evaluation frameworks, such as the Gen AI Evaluation Service in Vertex AI to score each model, drive fine-tuning, guide prompt engineering, or automate human evaluation. These frameworks can be used to calculate pointwise and pairwise metrics. These frameworks can also provide quick feedback to developers, which enables faster development iterations.

[12] K-means: A clustering technique that groups data points to a set number of groups based on their similarity, which is represented by distance in a vector space

[13] Hierarchical clustering: A clustering technique that iteratively groups similar data points, which creates a series of increasingly large clusters

Solution builders might also want to use an open-weight model[14] and deploy it within their infrastructure for specific scenarios. Before solution builders deploy one of these models, an IT manager must assess its security posture and its permissible uses according to the model's license. To ensure that the model is secure, the IT manager performs scans, like malware and pickle scanning[15], to prevent security breaches. Similarly, IT managers should also work with in-house counsel or legal advisors to ensure that the model can be used for their use case.

Finally, for **high-volume or real-time use cases (such as customer service agents), minor performance differences in latency, cost, and safety can significantly impact end-user experience**. For instance, in a customer service context, a latency of 10 seconds per query could extend complex calls by minutes, which would then increase median response times and impact the overall experience. Therefore, technical evaluations are required to ensure that solutions can scale to meet production demands.

In the IT assistant example, after the initial pilot, the team planned to develop a solution that scaled to more users and added new functionality. They conducted a rigorous evaluation of the increased demand and new features as part of the development process. Specifically, the team selected the following metrics:

- **Accuracy:** The similarity between an ideal response and the system's response
- **Groundedness:** The alignment between an answer and the source data
- **Coherence:** The ability of a system to generate a logical, organized, and coherent response
- **Brevity:** The ability to create a response of the correct length
- **Comprehensiveness:** The ability of a response to include all essential details

The following diagram shows the components of the contract-assistant solution after it was scaled to handle more users and offer additional features, and the evaluation steps for various components:

1. **Agent orchestrator (such as Vertex AI Agent or LangChain):** Evaluate accuracy, latency, and failure rates
2. **Enterprise procurement systems:** Conduct integration and configuration checks
3. **Gemini 1.5 Flash (LLM):** Test model performance by using public benchmarks and evaluate coherence and accuracy by using the Gen AI Evaluation Service in Vertex AI
4. **Real-time interaction and personalization:** Conduct human evaluation of the effectiveness of personalization

[14] We prefer to use the term "open-weight" because it more accurately reflects the current landscape, where many open-source models provide access to the pre-trained weights, but do not include the underlying training data.
[15] Pickle scanning: A way of scanning objects in a common Python storage format to identify any malware that is hidden in the model

5. **Successful responses:** Evaluate dataset for response quality and compliance with privacy requirements

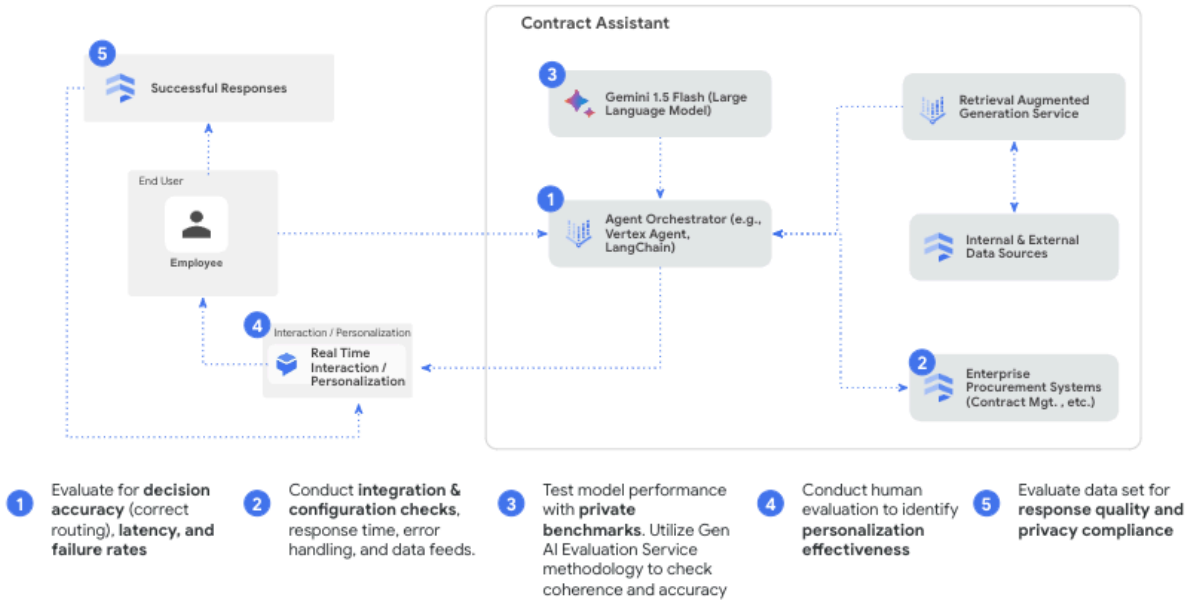

Figure 4: High-level architecture and evaluation steps for the example contract assistant

Their analytical process discovered that the performance of Gemini 1.5 Pro was comparable to using Gemini 1.5 Flash with additional prompt engineering. Gemini 1.5 Pro had slightly better performance, but Gemini 1.5 Flash had substantially lower costs. Consequently, the team switched to Gemini 1.5 Flash, improving the return on investment (ROI).

Before deploying to production, the team also conducted an extensive end-to-end evaluation using the Gen AI Evaluation Service in Vertex AI, which allows the use of LLMs to judge the solution's outputs. Similar to human evaluation, the LLM-as-a-judge approach calculated pairwise and pointwise metrics, identifying poorly performing cases that lacked detail. Based on the findings from the LLM-as-a-judge evaluation, the team adjusted the RAG architecture to provide adequate context by increasing the chunk size and adding adjacent chunk context. This optimized performance and ensured reliability after the solution was deployed to production.

This approach helps ensure that solutions requiring extensive scrutiny are robust, efficient, and aligned with organizational goals as they move from pilot testing to full-scale production.

Evaluations in the scale phase

Although evaluation is integral to any implementation, it is time consuming because of several time-intensive steps and the need for use-case-specific data. Evaluation can slow down generative AI adoption for organizations that want to adopt multiple use cases. As such, teams need tools and practices to accelerate the evaluation process, including prioritizing use cases, using a platform or evaluation service, and integrating it with third-party governance solutions. **By reducing the evaluation rigor for a low-risk and low-impact use case—like an ad-copywriting assistant—solution builders can free up additional resources for higher-risk, high-impact use cases, such as personalized content-creation agents.**

Even with prioritization, evaluation can be time consuming. As organizations scale, they can create standardized and automated evaluations in addition to existing, use-case-specific evaluations. By adding these evaluations, they can use a generative AI platform to automate evaluations and streamline the evaluation process. These platforms offer features such as the following:

- **Prompt versioning**: Manages iterations of prompts, enabling rollbacks and root cause analysis
- **Evaluation-dataset creation**: Facilitates the generation of datasets tailored for specific testing scenarios
- **Model evaluation**: Provides tools to systematically assess the performance and accuracy of AI models
- **Human-preference evaluation**: Incorporates user feedback to refine AI responses according to human preferences
- **Safety filtering**: Implements safeguards to prevent generating harmful or inappropriate content
- **High-quality retrieval**: Simplifies the evaluation of RAG systems by offering configurable search capabilities (such as semantic search) that need evaluation of outputs instead of internal components (such as embeddings, a retriever, and a re-ranker).
- **Check grounding**: Increases confidence that AI responses are well-founded and factually correct

Like DevOps or MLOps strategies, **adopting an evaluation framework and platform can alleviate the burden on development teams and accelerate deployments**. An executive from a leading healthcare organization stated, "Our platform enables our teams to select from suggested metrics and submit directly to our governance council. By making evaluations easy for teams, they can implement faster and maintain a higher level of compliance."

In the IT organization example, the team matured the assistant into an agent while other teams deployed multiple generative AI applications, such as a CRM assistant and a legal assistant. **Consequently, their organization invested in an end-to-end evaluation platform to ease the burden of solution creation.** The following figure shows the high-level architecture for the scaled IT agent, and the steps used to evaluate various components:

1. **LLM planning chain and standard operating procedures:** Evaluate accuracy of the plan, adherence, and cost efficacy of the planning chain
2. **Tools (such as Vertex Code Interpreter):** Evaluate its ability to call the correct function or tool and to execute accurately
3. **Enterprise processes:** Conduct integration and configuration checks
4. **Storage (such as working memory, intermediate storage, and others):** Measure quality and accuracy of stored variables
5. **Scaled contracting-agent solution:** Create scorecards for continuous evaluation and improvement

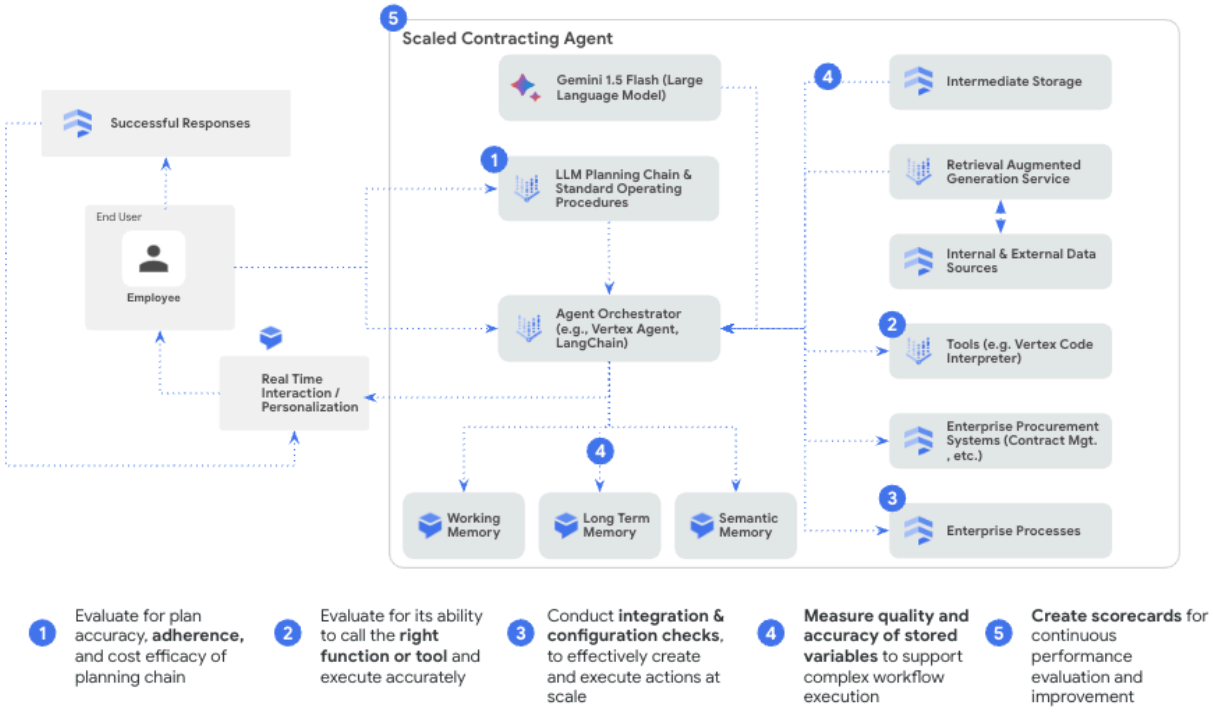

Figure 5: High-level architecture and evaluation steps for the example scaled IT agent

As these teams added more agentic capabilities, the figure shows that they used the platform's prebuilt usage tests to confirm that the model correctly interpreted and responded to the context. **Further, the team used the platform to facilitate long-horizon task checking, which evaluates how an agent guides employees through extended processes**, such as

automating contract approval processes. The team then consolidated evaluations into a comprehensive scorecard, which aided decision-making.

The platform is also integrated with the organization's governance process. It automatically generates submissions that include evaluation criteria for review by the organization's governance committee. This integration helps ensure that each step is documented and aligns with the organization's governance standards, which improves compliance.

Ultimately, organizations can effectively scale generative AI systems by adopting advanced evaluation frameworks and integrating them with standardized processes and governance **systems**. Doing so helps ensure both efficiency and adherence to quality and compliance standards.

Scaling evaluation to meet ambitions

A robust evaluation framework gives AI solution builders the confidence to deploy and scale generative AI applications across their organizations. As an example, the following figure shows each implementation phase (discover, design, develop, deploy, and operate) and the various evaluation activities that a team can perform:

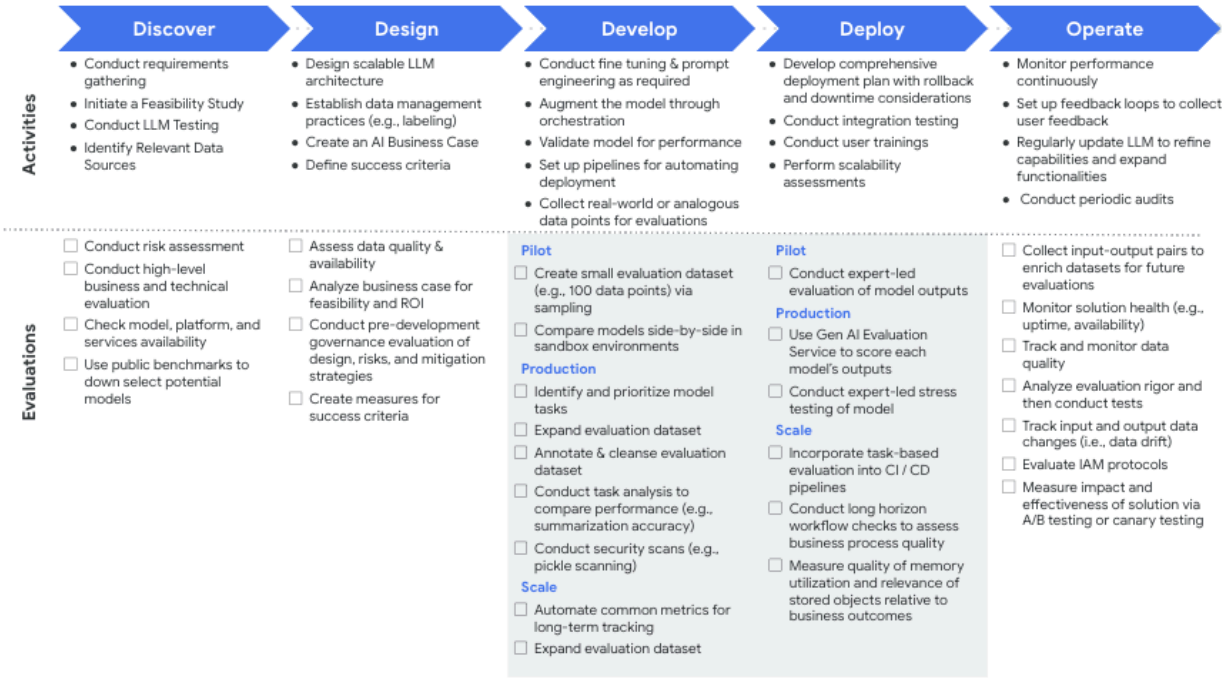

Figure 6: End-to-end evaluation checklist detailing high-level activities for each implementation phase

The figure shows the following activities for each phase:

- **Discover:** Conduct risk assessments, conduct high-level business and technical evaluations, check model platform and service availability, and use public benchmarks to help select a model
- **Design:** Analyze data quality and availability, analyze the business case, conduct a pre-development governance evaluation, and create measures of success
- **Develop:** Create evaluation dataset, compare model performance, identify and prioritize model tasks, annotate and cleanse the evaluation dataset, conduct task analysis, conduct security scans, automate long-term metrics for tracking, and expand the evaluation dataset
- **Deploy:** Conduct expert-led output analysis, use the Gen AI Evaluation Service to score outputs, conduct expert-led stress test of model, incorporate task-based evaluation into CI/CD pipelines, conduct long-horizon workflow checks for business process quality, measure the quality of memory use and the relevance of stored objects relative to business outcomes
- **Operate:** Collect input-output pairs to enrich datasets, monitor solution health, track and monitor data quality, analyze and test evaluation rigor, track data drift, evaluate Identity and Access Management (IAM) protocols, and measure the effectiveness of the solution

To start its evaluation journey, your organization should reflect on its current state of generative AI adoption. For organizations that are just beginning, you should focus on implementing basic screening tests by using public benchmarks, technical characteristics, or fundamental cost analysis. If your organization has already moved models into production, you should consider using a generative AI platform or evaluation tool to start implementing task analyses. Finally, if your organization is scaling generative AI or deploying agentic applications, you should focus on building standard evaluation datasets and metrics that can be directly integrated into your governance process. **As your organization's capabilities mature, your evaluation approach should evolve to meet the challenges of new or larger risks from increased users and new use cases.**

Incorporating a rigorous and adaptable evaluation framework is about more than immediate gains; it's about future-proofing your organization. As generative AI continues to evolve, so will the complexity and potential of the applications it enables. By adopting evaluation frameworks, you can empower your AI engineers and developers with feedback to enhance your solutions and give the business the confidence to deploy solutions at scale. **As a result, if your organization builds a solid foundation for evaluations, it is better positioned to realize the value of your generative AI investments—without significant overhauls to your architectures and processes.**

However, even with this solid foundation, you should focus on continuous improvement. You can improve the relevance and efficacy of your AI solutions by regularly revisiting and refining the evaluation criteria, staying abreast of the latest technological advancements, and incorporating feedback from all stakeholders.

Ultimately, **a well-structured generative AI evaluation framework is a cornerstone for successful AI deployment and scaling.** It mitigates risks, helps ensure high performance, and aligns AI initiatives with strategic goals. As your organization embarks on this journey, you should embrace a mindset of continuous learning and adaptation, which positions your organization to capitalize on the power of generative AI. Doing so helps to enhance operations, pave the way for future innovations, and sustain competitive advantage.