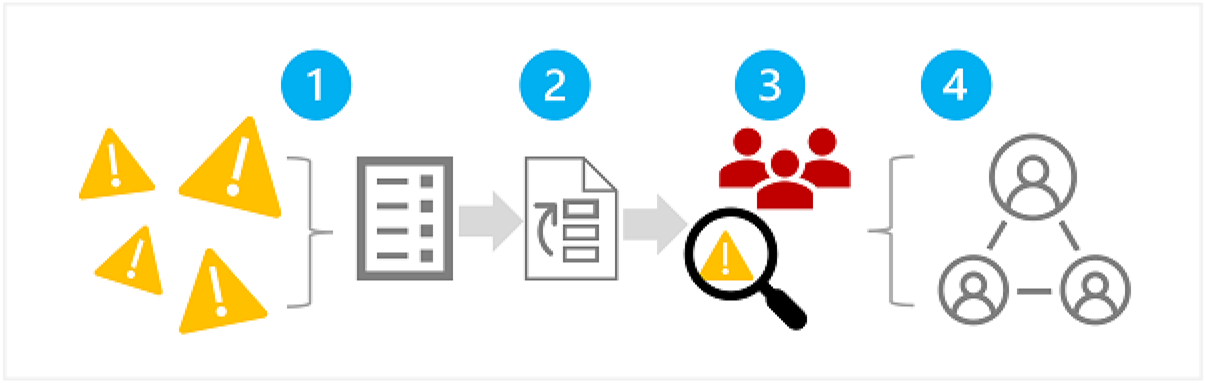
100 XP

Identify potential harms

5 minutes

The first stage in a responsible generative AI process is to identify the potential harms that could affect your planned solution. There are four steps in this stage, as shown here:



1. Identify potential harms
2. Prioritize identified harms
3. Test and verify the prioritized harms
4. Document and share the verified harms

1: Identify potential harms

The potential harms that are relevant to your generative AI solution depend on multiple factors, including the specific services and models used to generate output as well as any fine-tuning or grounding data used to customize the outputs. Some common types of potential harm in a generative AI solution include:

- Generating content that is offensive, pejorative, or discriminatory.
- Generating content that contains factual inaccuracies.
- Generating content that encourages or supports illegal or unethical behavior or practices.

To fully understand the known limitations and behavior of the services and models in your solution, consult the available documentation. For example, the Azure OpenAI Service includes a transparency note; which you can use to understand specific considerations related to the service and the models it includes. Additionally, individual model developers may provide documentation such as the OpenAI system card for the GPT-4 model.

Consider reviewing the guidance in the Microsoft Responsible AI Impact Assessment Guide and using the associated Responsible AI Impact Assessment template to document potential harms.

2: Prioritize the harms

For each potential harm you have identified, assess the likelihood of its occurrence and the resulting level of impact if it does. Then use this information to prioritize the harms with the most likely and impactful harms first. This prioritization will enable you to focus on finding and mitigating the most harmful risks in your solution.

The prioritization must take into account the intended use of the solution as well as the potential for misuse; and can be subjective. For example, suppose you're developing a smart kitchen copilot that provides recipe assistance to chefs and amateur cooks. Potential harms might include:

- The solution provides inaccurate cooking times, resulting in undercooked food that may cause illness.
- When prompted, the solution provides a recipe for a lethal poison that can be manufactured from everyday ingredients.

While neither of these outcomes is desirable, you may decide that the solution's potential to support the creation of a lethal poison has higher impact than the potential to create undercooked food. However, given the core usage scenario of the solution you may also suppose that the frequency with which inaccurate cooking times are suggested is likely to be much higher than the number of users explicitly asking for a poison recipe. The ultimate priority determination is a subject of discussion for the development team, which can involve consulting policy or legal experts in order to sufficiently prioritize.

3: Test and verify the presence of harms

Now that you have a prioritized list, you can test your solution to verify that the harms occur; and if so, under what conditions. Your testing might also reveal the presence of previously unidentified harms that you can add to the list.

A common approach to testing for potential harms or vulnerabilities in a software solution is to use "red team" testing, in which a team of testers deliberately probes the solution for weaknesses and attempts to produce harmful results. Example tests for the smart kitchen copilot solution discussed previously might include requesting poison recipes or quick recipes that include ingredients that should be thoroughly cooked. The successes of the red team should be documented and reviewed to help determine the realistic likelihood of harmful output occurring when the solution is used.

ⓘ **Note**

Red teaming is a strategy that is often used to find security vulnerabilities or other weaknesses that can compromise the integrity of a software solution. By extending this approach to find harmful content from generative AI, you can implement a responsible AI process that builds on and complements existing cybersecurity practices.

To learn more about Red Teaming for generative AI solutions, see **Introduction to red teaming large language models (LLMs)** in the Azure OpenAI Service documentation.

4: Document and share details of harms

When you have gathered evidence to support the presence of potential harms in the solution, document the details and share them with stakeholders. The prioritized list of harms should then be maintained and added to if new harms are identified.

Next unit: Measure potential harms

Continue >