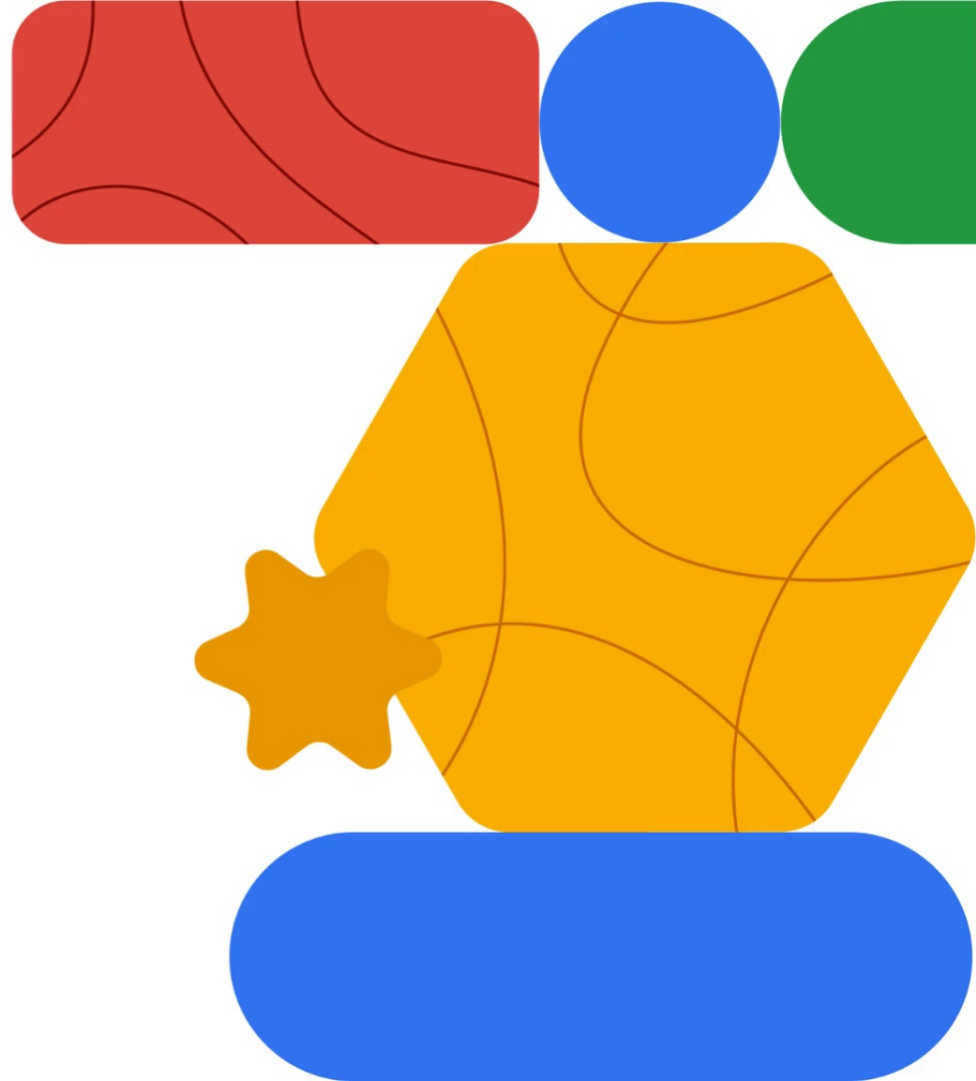
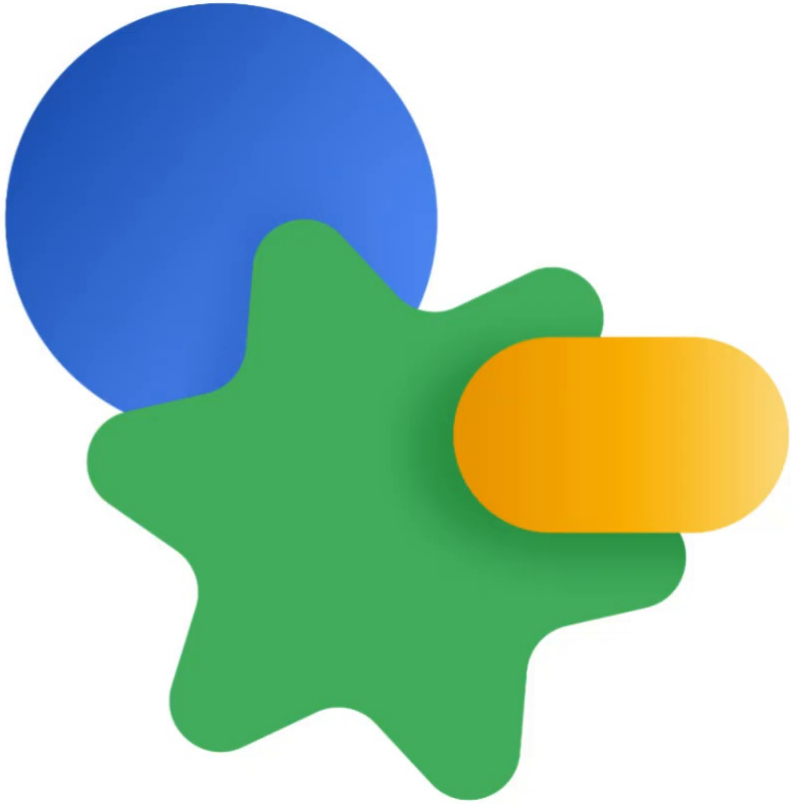
Google Cloud

# Implementing your testing strategy in Conversational Agents



Google Cloud

# Unit testing



# esting in Conversational Agents

categories: Route testing and NLU testing

## Route testing

- Testing each route available on a single page
- Includes conditional routes and fallbacks scenarios

## NLU testing

- Testing recognition of user phrases on specific pages

## Routi

Refe
Test
Age
spe
of t
enti

# Routing tests in Conversational Agents

## Definition

Referred to as Partial Page Testing in Conversational Agents, that focuses on specific components or pages of the agent, rather than the entire conversation flow.

## Scope

Limited to individual pages in a flow.

## Purpose

- To test the functionality of specific components or logic within a flow.
- To quickly identify and fix issues in isolated parts of the agent.

# Routing tests in Conversational Agents

## Definition

Referred to as Partial Page Testing in Conversational Agents, that focuses on specific components or pages of the agent, rather than the entire conversation flow.

## Scope

Limited to individual pages in a flow.

## Purpose

- To test the functionality of specific components or logic within a flow.
- To quickly identify and fix issues in isolated parts of the agent.

# Routing tests in Conversational Agents

## Definition

Referred to as Partial Page Testing in Conversational Agents, that focuses on specific components or pages of the agent, rather than the entire conversation flow.

## Scope

Limited to individual pages in a flow.

## Purpose

- To test the functionality of specific components or logic within a flow.
- To quickly identify and fix issues in isolated parts of the agent.

# Routing tests in Conversational Agents

## Definition

Referred to as Partial Page Testing in Conversational Agents, that focuses on specific components or pages of the agent, rather than the entire conversation flow.

## Scope

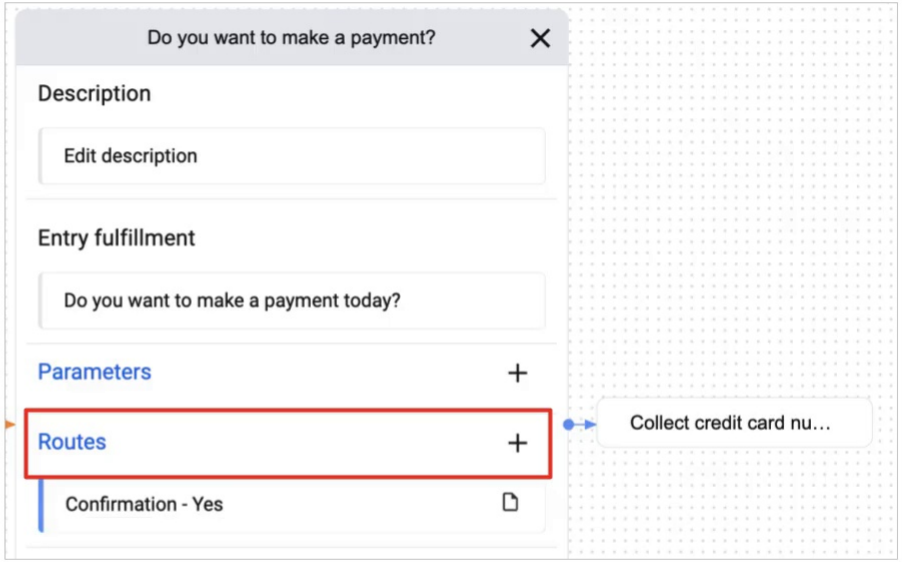
Limited to individual pages in a flow.

## Purpose

- To test the functionality of specific components or logic within a flow.
- To quickly identify and fix issues in isolated parts of the agent.

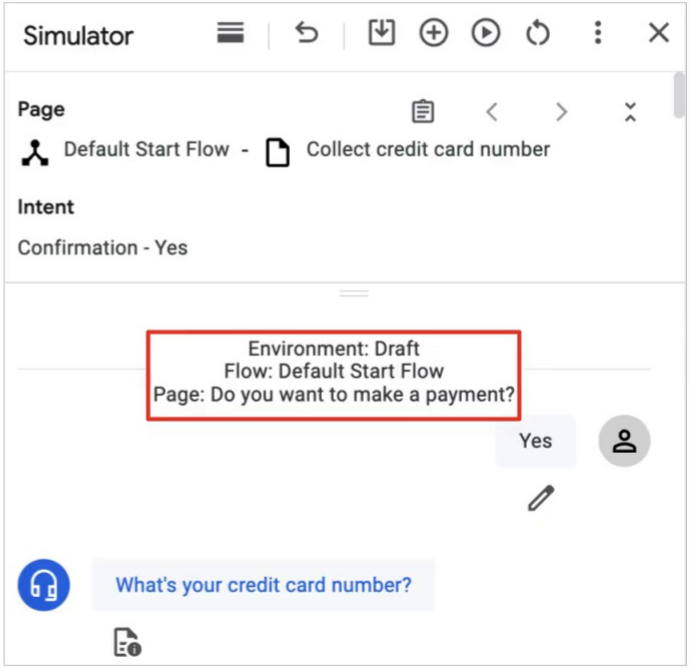
# Routing tests in Conversational Agents

For each route on a page, you need to create an explicit test case.



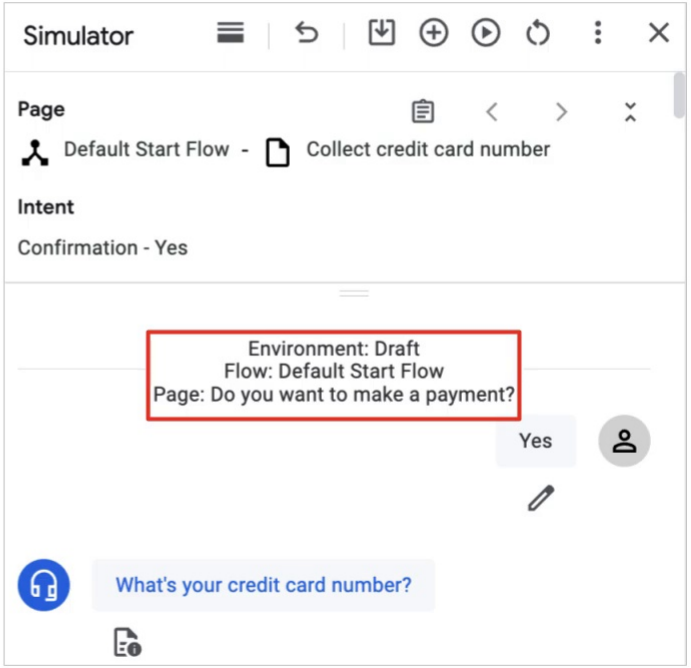
# Routing tests in Conversational Agents Simulator

To reduce turns included in test case, use a different input page as the start.



# Routing tests in Conversational Agents Simulator

To reduce turns included in test case, use a different input page as the start.



# NLU tests in Conversational Agents

## Definition

NLU tests can be run in Conversational Agents test cases, but it's recommended to test them in a process outside of Conversational Agents to allow testing of more utterances.

## Scope

Scope means it's limited to individual pages in a flow.

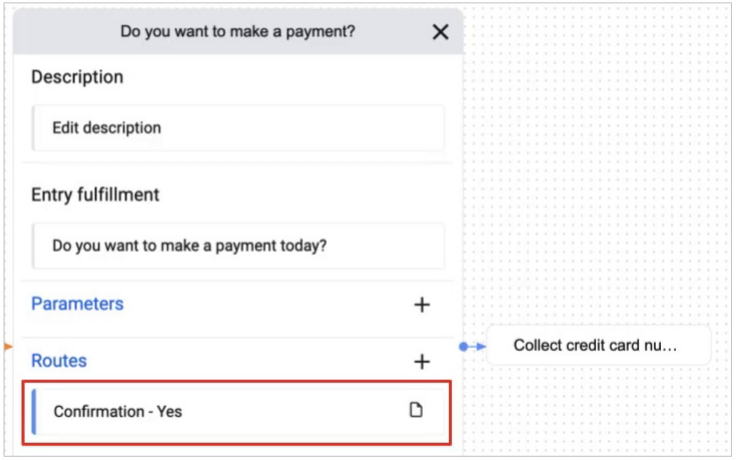
## Purpose

The purpose is to test the efficacy of user utterances getting tagged to the correct intent on each page.

# NLU tests in Conversational Agents

**NLU testing requires a wide variety of user utterances to be tested for their intent in Conversational Agents**

| NLU Testing | | |
|---|---|---|
| **Page** | **Utterance** | **Expected intent** |
| Do you want to make a payment? | Yes | Confirmation - Yes |
| Do you want to make a payment? | Sure | Confirmation - Yes |
| Do you want to make a payment? | Of course | Confirmation - Yes |



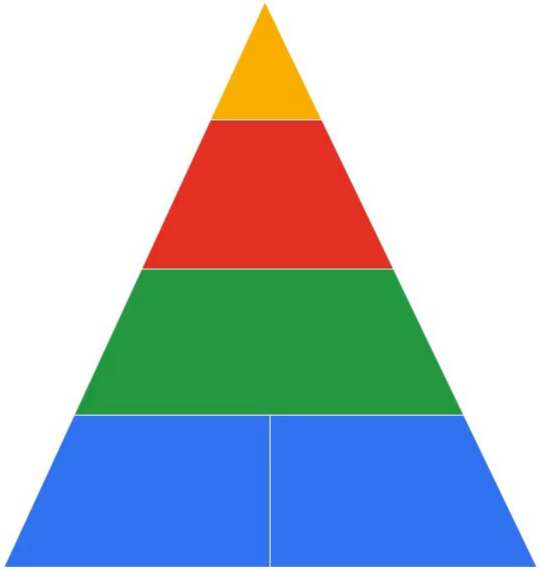
Google Cloud

# Unit test driven development

# The importance of test driven development

**Objective**

The end to end development life cycle of a conversational agent needs to be supported by a **robust test strategy.**



A/B tests

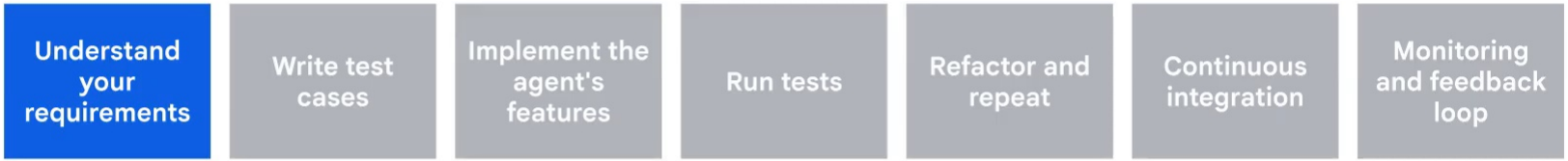
E2E/UAT tests

Integration system tests

Unit tests

# Test driven development

Process



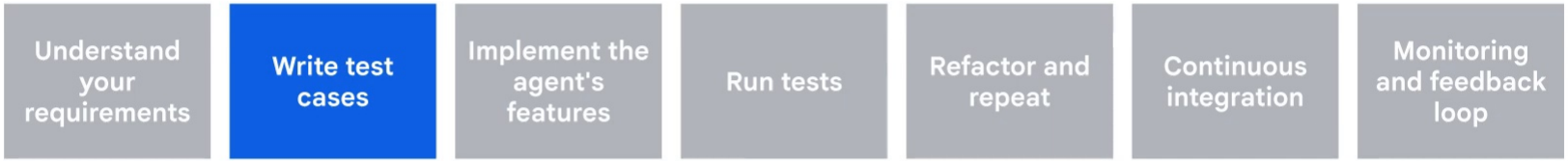

Explanation

**What:**
Before you start writing test cases, clearly define what you expect your agent in Conversational Agents to do. This includes understanding the different types of user inputs (intents) and the expected responses or actions from the agent.

**How:**
Define your CUJs.

# Test driven development

Process



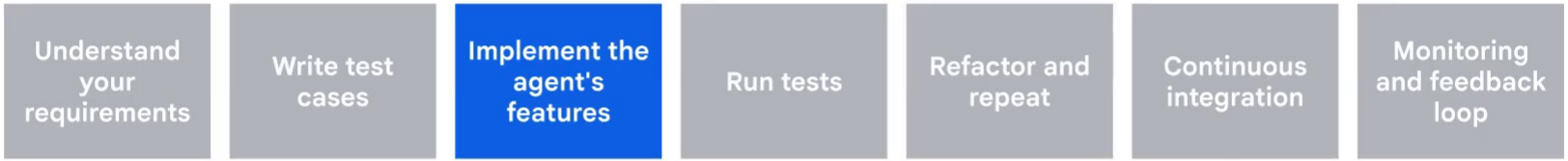

Explanation

**What:**
Start by writing test cases before developing the actual features. For an agent in Conversational Agents, these test cases can include different scenarios of user-agent interactions. You might write tests for how the agent should respond to specific intents, how it handles context switching, or how it manages fallbacks.

**How:**
Map your CUJ tasks to your test cases via a Requirements Traceability Matrix (RTM).

# Test driven development

Process



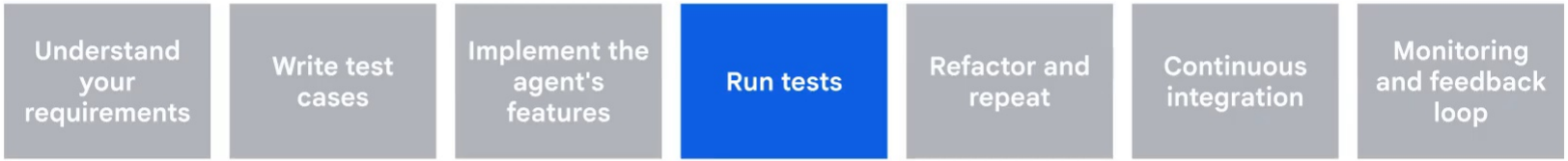

Explanation

**What:**
With your tests in place, begin implementing the agent's features. This involves defining intents, training phrases, parameters, responses, and fulfillment logic in Conversational Agents.

**How:**
Standard Conversational Agents development and using the simulator as you build the feature.

# Test driven development

Process



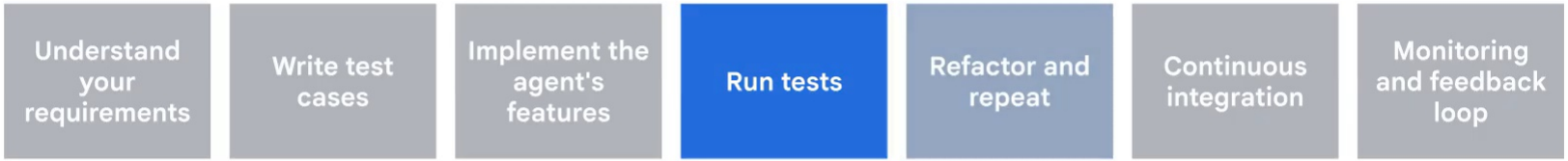

Explanation

**What:**
After implementing a feature, run your tests to see if the agent behaves as expected.

**How:**
The standard approach is to use the native menus in Conversational Agents. You can also use automated scripts for running test cases via the API.

# Test driven development

Process



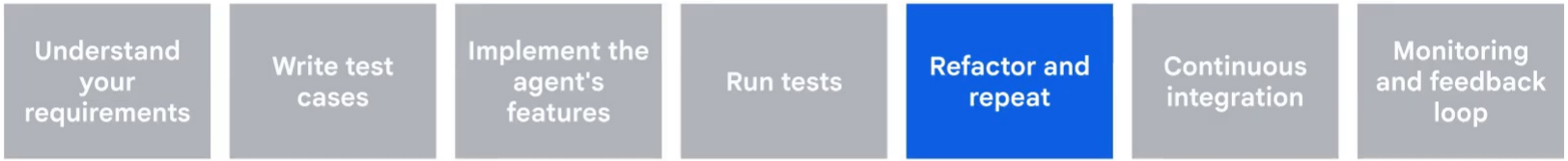

Explanation

**What:**
After implementing a feature, run your tests to see if the agent behaves as expected.

**How:**
The standard approach is to use the native menus in Conversational Agents. You can also use automated scripts for running test cases via the API.

# Test driven development

Process

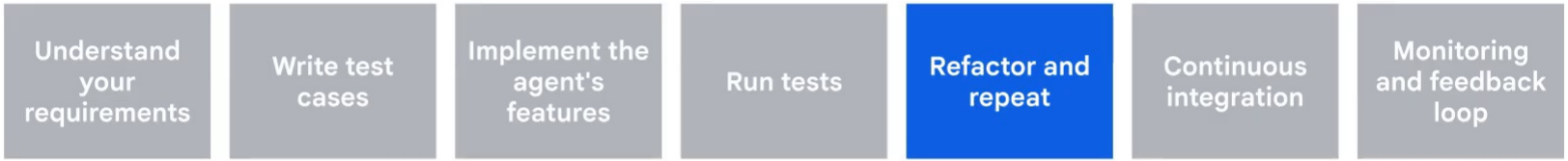



Explanation

**What:**
If tests fail, modify your agent's configuration and logic until all tests pass. Then, refactor your code to improve efficiency, readability, and maintainability. After refactoring, run the tests again to ensure nothing broke during the process. Repeat this cycle as you continue to develop new features.

# Test driven development

Process

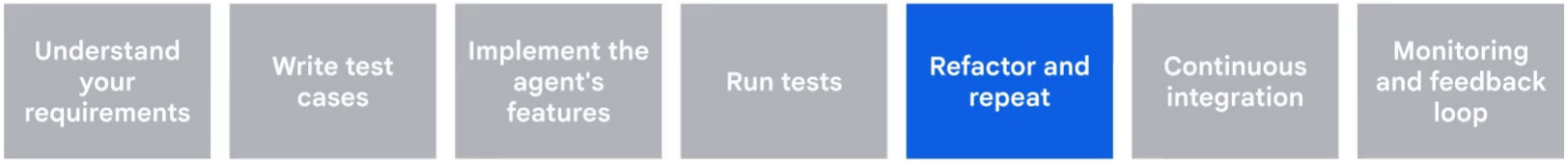



Explanation

**What:**
If tests fail, modify your agent's configuration and logic until all tests pass. Then, refactor your code to improve efficiency, readability, and maintainability. After refactoring, run the tests again to ensure nothing broke during the process. Repeat this cycle as you continue to develop new features.

# Test driven development

Process

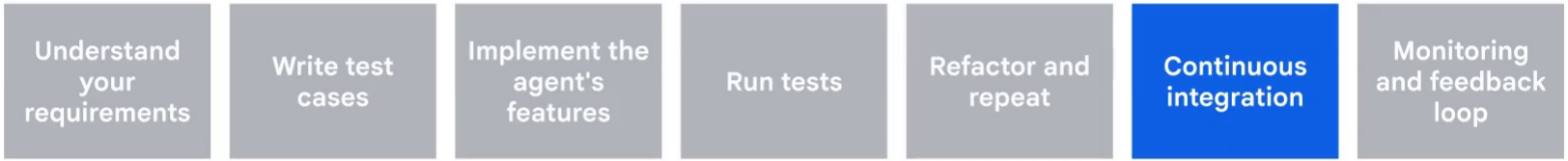



Explanation

**What:**
If tests fail, modify your agent's configuration and logic until all tests pass. Then, refactor your code to improve efficiency, readability, and maintainability. After refactoring, run the tests again to ensure nothing broke during the process. Repeat this cycle as you continue to develop new features.

# Test driven development

Process

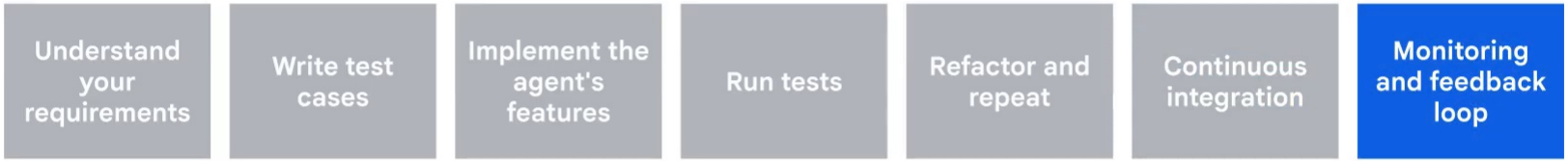



Explanation

**What:**
For more advanced setups, integrate your TDD process with a continuous integration system. This will allow you to automatically run tests whenever changes are made to the agent, ensuring consistent quality and functionality.

# Test driven development

Process



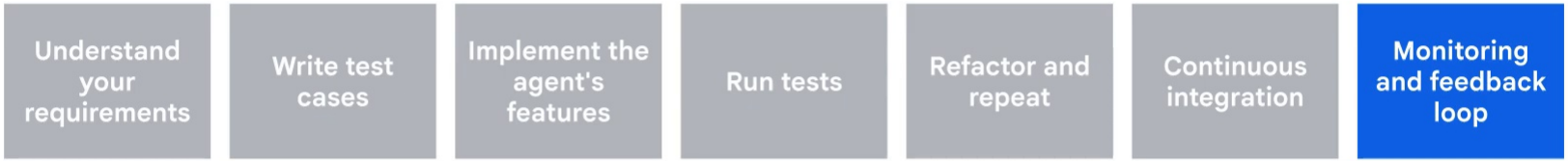

Explanation

**What:**
After deployment, continuously monitor the agent's performance. Collect user feedback and conversation logs to identify areas for improvement. Write new tests for these improvements and follow the TDD cycle to implement them.

**How:**
Document Key Learnings (KLRs) derived from test results to share with other developers.

# Test driven development

Process




Explanation

**What:**
After deployment, continuously monitor the agent's performance. Collect user feedback and conversation logs to identify areas for improvement. Write new tests for these improvements and follow the TDD cycle to implement them.

**How:**
Document Key Learnings (KLRs) derived from test results to share with other developers.


Google Cloud