First, determine what initiates the workflow. Consider a record creation/update in Dynamics CRM.Understand what the Azure Function accomplishes.Define the intended outcome.

Establish a Test Environment with a Dynamics 365 Sandbox, and azure function slot, and mock for external applications or tools.

Specify Test Situations

Happy path or positive scenario with the legitimate data and validation to verify the function handles the data appropriately

Error scenarios for handles errors or mistakes with incorrect validations or wrong data

Test retry procedures or error handling by simulating external dependency failures like API outages

Conducting Tests

Starting the test and checking how function is being used using azure application insights to validate logs and tracing.

Tools

When making API calls to Dynamics CRM, use tools such as Postman.

Use frameworks such as xUnit/NUnit to implement automated tests (for.NET-based testing).

Selenium (in the case of a front-end user interface). Use Azure Application Insights to monitor and debug Azure Function execution. For continuous integration and deployment, use Azure DevOps Pipelines or Power Platform Build Tools. WireMock for mocking external services.