automate business process/workflow

workflow

Business processes modeled in software are often called workflows.

workflow options from azure

Azure includes four different technologies that you can use to build and implement workflows that integrate multiple systems:

- Logic Apps
- Microsoft Power Automate(flow)
- WebJobs
- Azure Functions

Design-first technologies

it has gui to help define the workflow.

- Logic Apps
- Microsoft Power Automate.

  There are four different types of flow that you can create:

  - Automated: A flow that is started by a trigger from some event. For example, the event could be the arrival of a new tweet or a new file being uploaded.
  - Button: Use a button flow to run a repetitive task with a single click from your mobile device.
  - Scheduled: A flow that executes on a regular basis such as once a week, on a specific date, or after 10 hours.
  - Business process: A flow that models a business process such as the stock ordering process or the complaints procedure.

Design-first technologies compared

As you can see from the following table, Microsoft Power Automate is more appropriate for use by non-technical staff. If your workflow designers are IT professionals, developers, or DevOps practitioners, Logic Apps are usually a better fit:

	Microsoft Power Automate	Logic Apps
Intended users	Office workers and business analysts	Developers and IT pros
Intended scenarios	Self-service workflow creation	Advanced integration projects
Design tools	GUI only. Browser and mobile app	Browser and Visual Studio designer. Code editing is possible
Application Lifecycle Management	Power Automate includes testing and production environments	Logic Apps source code can be included in Azure DevOps and source code management systems

Code-first technologies

- webjobs. it is a part of the Azure App Service that you can use to run a program or script automatically.
    1. Continuous
    2. Triggered
    3. The WebJobs SDK only supports C# and the NuGet package manager.
- azure functions

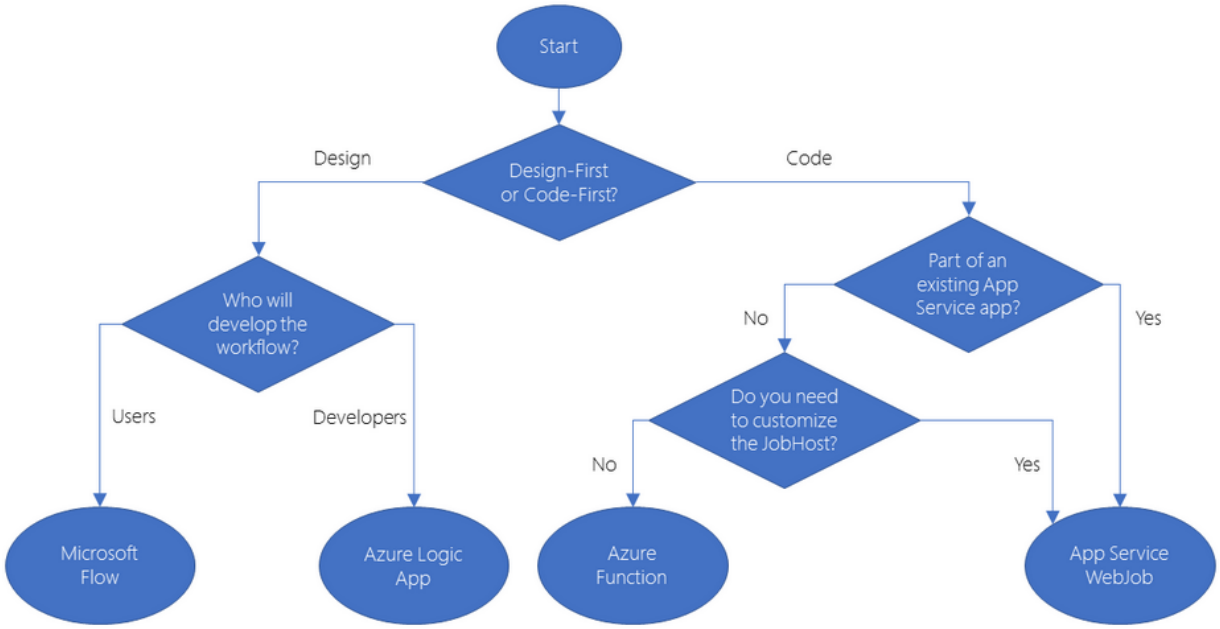
webjobs vs azure functions

In most cases, the simple administration and more flexible coding model provided by Azure Functions may lead you to choose them in preference to WebJobs. However, you may choose WebJobs for the following reasons:

You want the code to be a part of an existing App Service application and to be managed as part of that application, for example in the same Azure DevOps environment. You need close control over the object that listens for events that trigger the code. This object in question is the JobHost class, and you have more flexibility to modify its behavior in WebJobs.

	Azure WebJobs	Azure Functions
Supported languages	C# if you are using the WebJobs SDK	C#, Java, JavaScript, PowerShell, etc.
Automatic scaling	No	Yes
Development and testing in a browser	No	Yes
Pay-per-use pricing	No	Yes
Integration with Logic Apps	No	Yes
Package managers	NuGet if you are using the WebJobs SDK	Nuget and NPM
Can be part of an App Service application	Yes	No
Provides close control of `JobHost`	Yes	No

how to choose a service


Start
Design-First or Code-First?
Design
Code
Who will develop the workflow?
Users
Developers
Microsoft Flow
Azure Logic App
Part of an existing App Service app?
No
Yes
Do you need to customize the JobHost?
No
Yes
Azure Function
App Service WebJob