





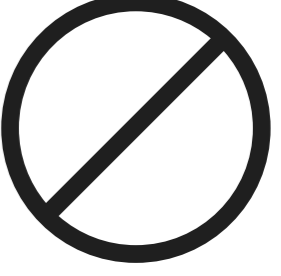


























Chains and Scripting Usage Guidance

Chains and Scripting can complement each other well and can be used together.

Key

-  Best option
-  OK option
-  Caution - limitations
-  Not an option

I have or need the following for my Workiva integration / automation project:	Chains and Scripting	Chains	Scripting
"Out-of-the-box" integrations (connectors) available for both Workiva and 3rd-party software			
Low-to-medium complexity application logic			
High complexity application logic (e.g. deeply nested loops, deeply nested conditionals, complex iterations w/ conditional, nested try-catch scenarios, etc.)			
Typically lower application development & maintenance costs			
Limited application development and maintenance budget			
High & strict runtime performance requirements			
Developer control over ability to tune runtime execution performance			
Less-steep learning curve for citizen developers			
Pool of developers trained on the application development method (e.g. Chains vs Python)		 Workiva Chains training required, but plenty of implementation resources & partners available.	 Top world-wide programming language. Massive developer pool.

Deciding When to Incorporate Scripting

