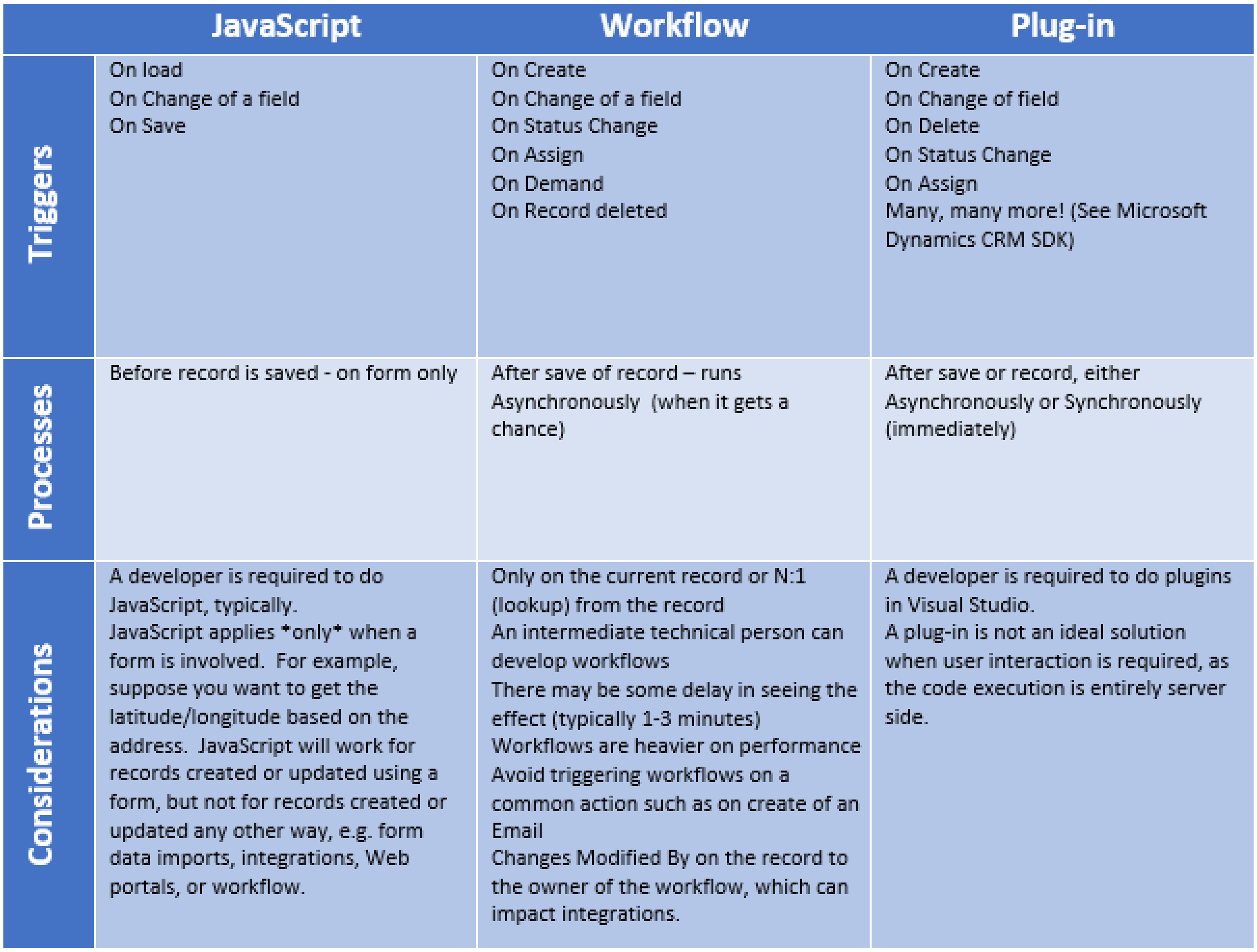
JavaScript vs. Workflow vs. Plug-In

Sometimes it is a challenge to know which methods to use when you need to extend CRM to meet more complex business needs. Often, you may be able to solve a problem in multiple ways. Perhaps JavaScript and oData will allow you to accomplish the same goal as would a plug-in. But, understanding the broader pros and cons of each method of extending CRM is important as well as knowing what methods are best for your programming resources.

The following grid provides some comparisons between JavaScript, workflows, and plug-ins. In addition, below is a list of important skills needed for various extending options.



	JavaScript	Workflow	Plug-in
Triggers	On load On Change of a field On Save	On Create On Change of a field On Status Change On Assign On Demand On Record deleted	On Create On Change of field On Delete On Status Change On Assign Many, many more! (See Microsoft Dynamics CRM SDK)
Processes	Before record is saved - on form only	After save of record – runs Asynchronously (when it gets a chance)	After save or record, either Asynchronously or Synchronously (immediately)
Considerations	A developer is required to do JavaScript, typically. JavaScript applies *only* when a form is involved. For example, suppose you want to get the latitude/longitude based on the address. JavaScript will work for records created or updated using a form, but not for records created or updated any other way, e.g. form data imports, integrations, Web portals, or workflow.	Only on the current record or N:1 (lookup) from the record An intermediate technical person can develop workflows There may be some delay in seeing the effect (typically 1-3 minutes) Workflows are heavier on performance Avoid triggering workflows on a common action such as on create of an Email Changes Modified By on the record to the owner of the workflow, which can impact integrations.	A developer is required to do plugins in Visual Studio. A plug-in is not an ideal solution when user interaction is required, as the code execution is entirely server side.