

Updating Legacy SharePoint Customizations

ASSESSING LEGACY SHAREPOINT CUSTOMIZATIONS



Danny Jessee

SENIOR SHAREPOINT DEVELOPER

@dannyjessee



Introduction

Over 200,000
organizations and
190 million people

Minor user
interface updates

Custom web parts
for data entry
and display

Tooling and
guidance
has evolved

Support for legacy
customizations
varies

Farm solutions
cannot be deployed
in the cloud



Course Scenario



**Currently using SharePoint
2013 on-premises**

Moving to SharePoint Online

Variety of legacy customizations

Migrate to SharePoint Framework



SharePoint Framework



New supported model for developing customizations



Used with SharePoint 2016 FP2 and later, SharePoint Online



Pluralsight course: “Getting Started with SharePoint Framework”



Pluralsight course: “Tooling up for SharePoint Framework”



Legacy Customization Approaches

SharePoint 2007

Farm solutions

Deploy assemblies to the server's Global Assembly Cache

SharePoint 2013

App (Add-in) model

Client-side solutions
Content and Script Editor Web Parts

SharePoint 2010

Sandboxed solutions

Process isolation
Simpler deployments

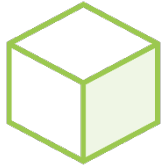
2013-2016+

Client-side approaches

App (Add-in) parts
App script parts



Identifying Legacy Customizations



Farm and sandboxed solutions



PowerShell scripts



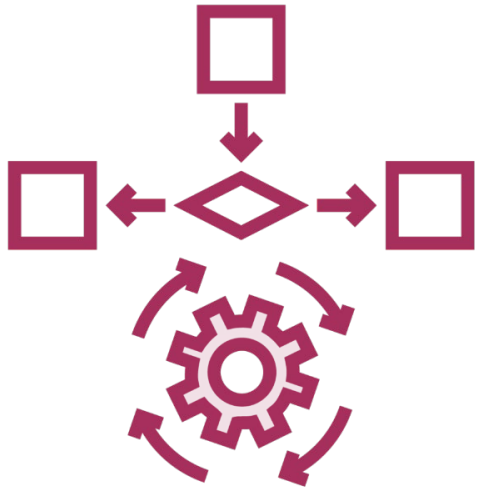
Navigate directly through SharePoint user interfaces



Content and Script Editor web parts on individual pages



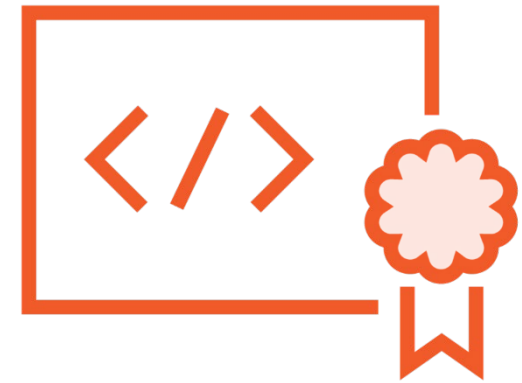
Why SharePoint Framework?



Existing customizations
may offer very
little governance



No-script sites protect
against ad hoc scripts but
disable all custom scripts



SharePoint Framework
allows custom scripts to
run on no-script sites

TypeScript

Adds static typing, classes,
and interfaces to JavaScript

Do not necessarily need to
rewrite all existing JavaScript

Ensure existing JavaScript can
be packaged in a module and
loaded asynchronously

Microsoft recommends
TypeScript for SharePoint
Framework development



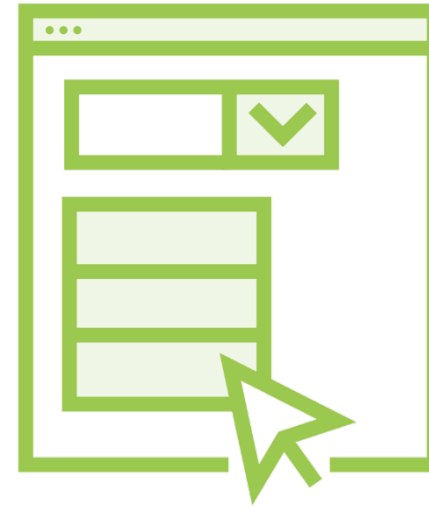
SharePoint Framework Migration Options

Client-side web parts



Traditional web parts
Some Content and
Script Editor web parts

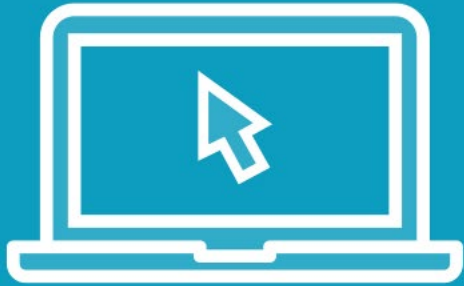
Extensions



DOM manipulation (header/footer)
Custom ECB menu items
JSLink scripts for list views



Demo



On-premises SharePoint 2013 site

How to take inventory of customizations

Script Editor web parts

User Custom Actions for ECB menu items

Farm and sandboxed solutions



Review



Assessing an existing environment

Legacy customization approaches

Benefits of SharePoint Framework

Client-side web parts and Extensions

Taking inventory of customizations
on an existing site



Up Next:

Updating Script Editor Web Part Customizations

