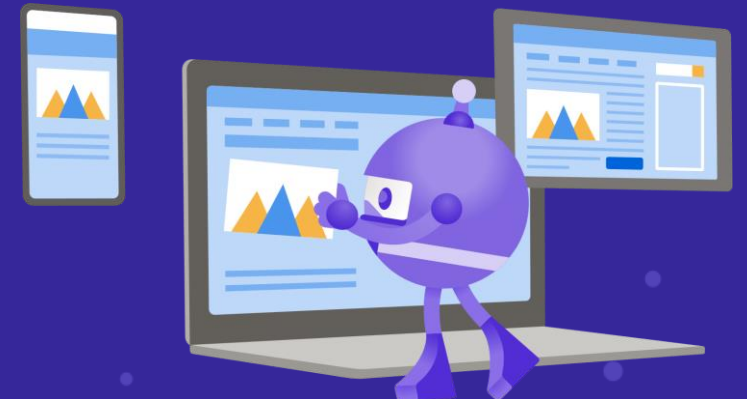


# Migration Paths and Tooling from Xamarin to .NET MAUI

Sweekriti Satpathy



# Who's this gal?



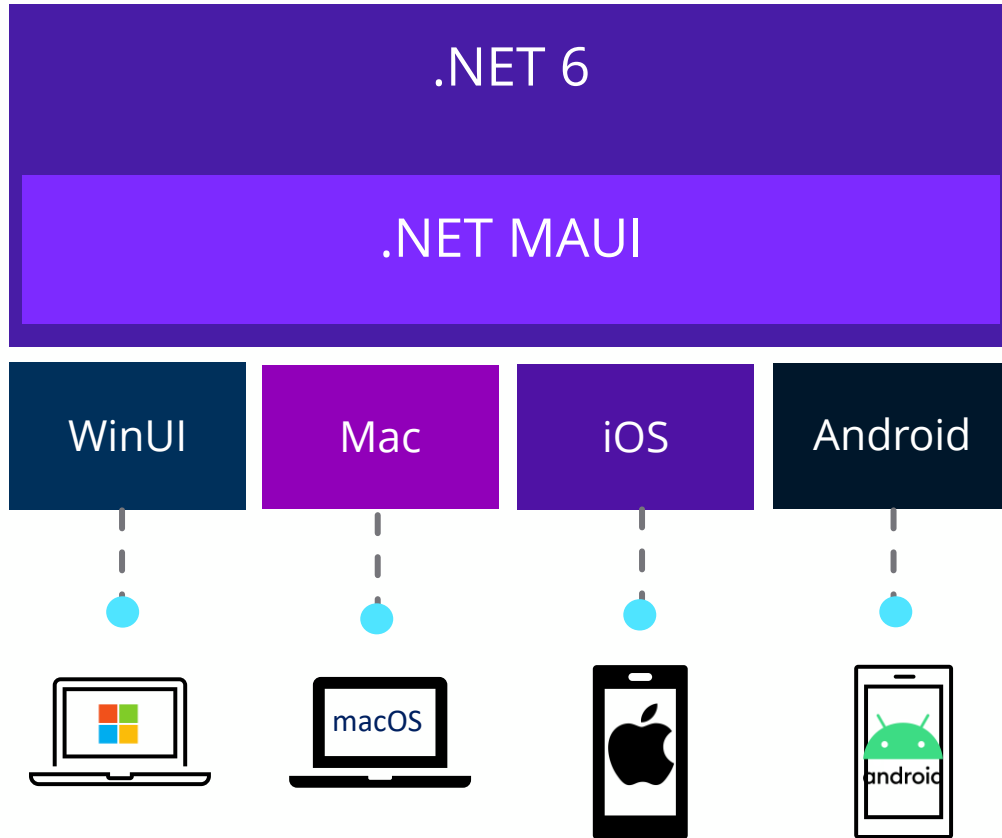
Sweekriti "Sweeky" Satpathy  
Senior Software Engineer– Customer Advisory Team, Microsoft

---

[swsat@microsoft.com](mailto:swsat@microsoft.com)

[@Sweekritis](https://twitter.com/Sweekritis)

# .NET Multi-platform App UI



Cross-platform, native UI

Single project system, single codebase

Deploy to multiple devices, mobile & desktop

Available

Visual Studio 2022

Visual Studio for Mac (Preview)

[github.com/dotnet/maui](https://github.com/dotnet/maui)

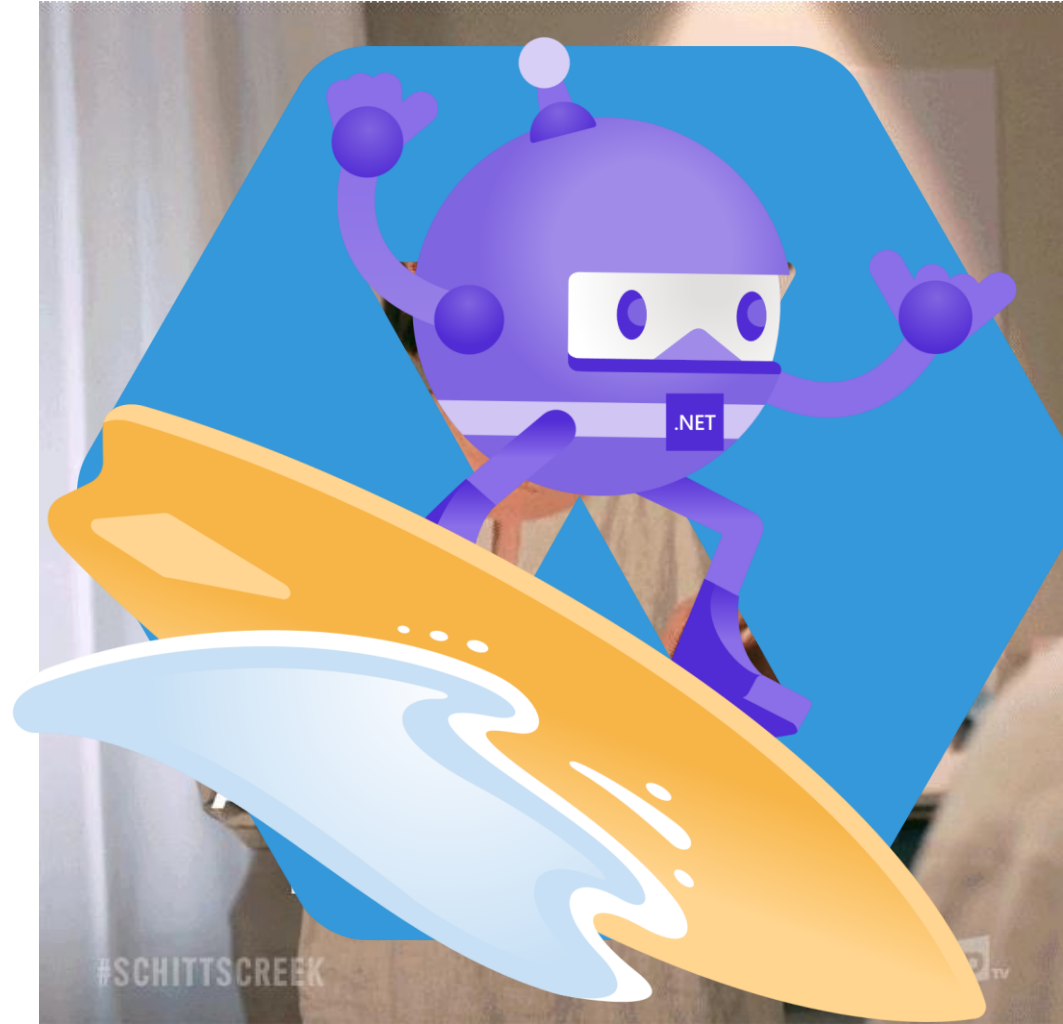
# What is Migration?



# What is Migration?



# What is Migration?







Why Upgrade?

## Is Xamarin dead?

No! !!!! !!!!!!!!!!!!! .NET MAUI and .NET 6 are an evolution!

## Do I need to rewrite my apps?

Also no!!!!



## What happens to Xamarin.Android? Xamarin.iOS?

They're just .NET for Android and .NET for iOS now!

## When do I start trying to migrate?

Get up to date now, but there is no rush. You have till **May 1, 2024**

## Ability to use dotnet everywhere!!

Simplify build process, ability to run dotnet CLI everywhere.



# 8 Enhancements “Under the hood”

- **Decoupled platform controls from cross-platform controls**  
Maintainability, easier to extend/modify, faster, and allows for alternate app models (MVU)
- **Multi-window**  
Support desktop & tablet scenarios
- **App Builder**  
To standardize app bootstrapping with common .NET pattern
- **Single Project**  
To simplify asset management, NuGet management, and leverage multi-targeting
- **Unified with .NET**  
Take advantage of common innovations in project system, CLI experiences, hot reload, and C#
- **Rebuilt Layouts**  
To improve maintainability, performance, and correct many quirks present in Xamarin.Forms
- **Layered renderer pattern over new handlers**  
To ease migration of Xamarin.Forms custom renderers to .NET MAUI
- **Refactored Shell implementations**  
To provide consistency whether you use Shell or base types like FlyoutPage; Maintainability



PlantLady.csproj M X

C: &gt; code &gt; PlantLady &gt; PlantLady &gt; PlantLady &gt; PlantLady.csproj

```
1 <Project Sdk="Microsoft.NET.Sdk">
2
3   <PropertyGroup>
4     <TargetFrameworks>net6.0-ios;net6.0-android</TargetFrameworks>
5     <UseMaui>True</UseMaui>
6     <OutputType>Library</OutputType>
7     <ImplicitUsings>enable</ImplicitUsings>
8     <!-- Required for C# Hot Reload -->
9     <UseInterpreter Condition="'$(Configuration)' == 'Debug'">True</UseInterpreter>
10    <SupportedOSPlatformVersion Condition="'$(TargetFramework)' == 'net6.0-ios'">15.4</
11    <SupportedOSPlatformVersion Condition="'$(TargetFramework)' == 'net6.0-android'">31.0</
12    <SupportedOSPlatformVersion>
13  </PropertyGroup>
14
15  <ItemGroup>
16    <MauiFont Include="Resources\*" />
17  </ItemGroup>
18</Project>
```



PlantLady.csproj X

C: &gt; code &gt; plantladyMAIN &gt; PlantLady &gt; PlantLady &gt; PlantLady &gt; PlantLady.csproj

```
1 <Project Sdk="Microsoft.NET.Sdk">
2
3   <PropertyGroup>
4     <TargetFramework>netstandard2.0</TargetFramework>
5     <ProduceReferenceAssembly>true</ProduceReferenceAssembly>
6   </PropertyGroup>
7
8   <PropertyGroup Condition="'$(Configuration)|$(Platform)' == 'Debug|AnyCPU'">
9     <DebugType>portable</DebugType>
10    <DebugSymbols>true</DebugSymbols>
11  </PropertyGroup>
12
13  <ItemGroup>
14    <None Remove="Resources\AmaticSC-Regular.ttf" />
15    <None Remove="Resources\materialdesignicons-webfont.ttf" />
16    <None Remove="Resources\Michella-Garden.otf" />
17  </ItemGroup>
18
19  <ItemGroup>
20    <EmbeddedResource Include="Resources\AmaticSC-Regular.ttf" />
21    <EmbeddedResource Include="Resources\materialdesignicons-webfont.ttf" />
22    <EmbeddedResource Include="Resources\Michella-Garden.otf" />
23  </ItemGroup>
24
25  <ItemGroup>
26    <PackageReference Include="Xamarin.Forms" Version="5.0.0.2083" />
27    <PackageReference Include="Xamarin.Essentials" Version="1.7.0" />
28  </ItemGroup>
29
30  <ItemGroup>
31    <EmbeddedResource Update="Views\PlantsPage.xaml">
32      <Generator>MSBuild:UpdateDesignTimeXaml</Generator>
33    </EmbeddedResource>
34  </ItemGroup>
35
36</Project>
```

# Let's Migrate our Apps!



## DO

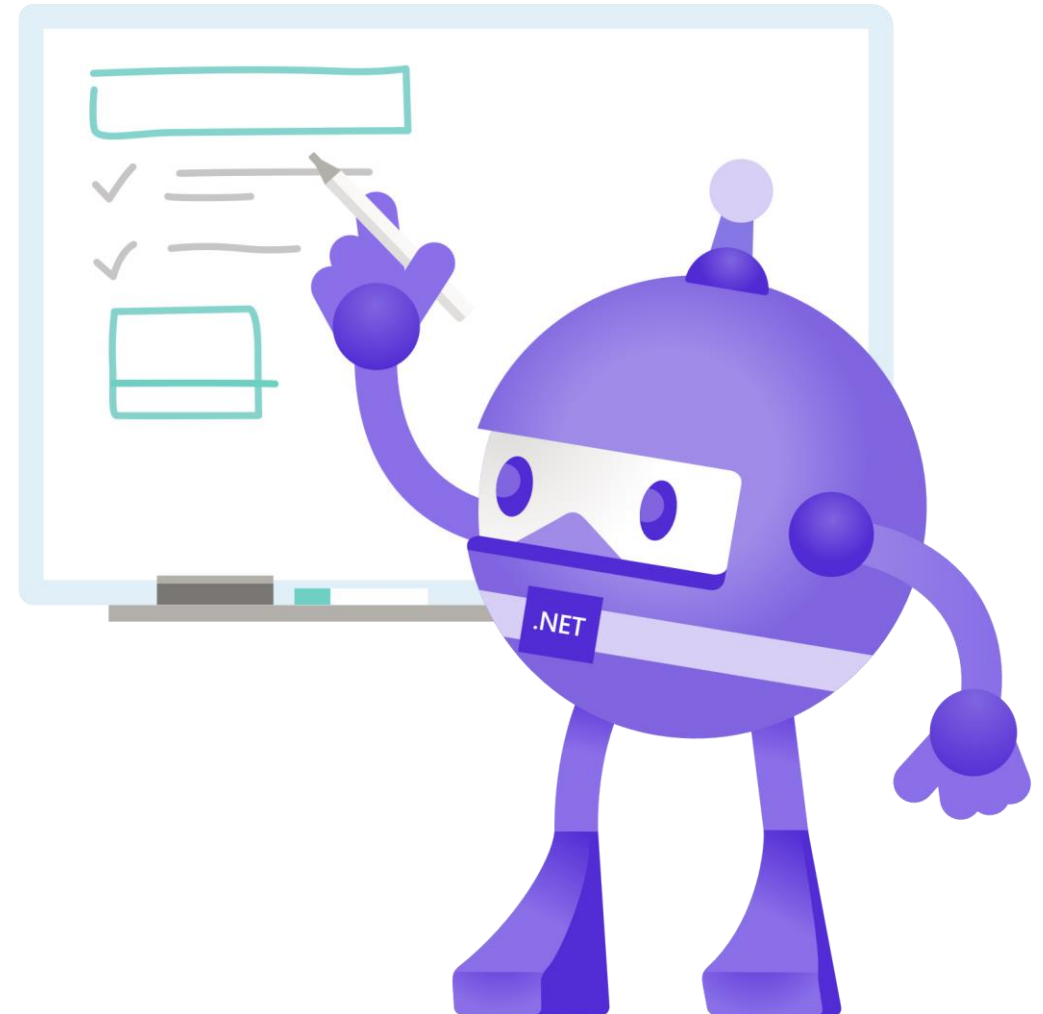
- Remove unnecessary custom renderers
- Use AppCompatActivity/AndroidX
- Use Fast Renderers
- Acquire Coconut water/fruity drink
- Upgrade to the latest reasonable Xamarin.Forms for you
- **Inventory your NuGet packages**

## DO NOT

- Use obsoleted APIs
- Panic!!!!
- Use DataPages or Themes NuGets
- Neglect your app for the next 6 months
- Forget to breathe.

# NuGet and Libraries and Packages, AAAAAA!

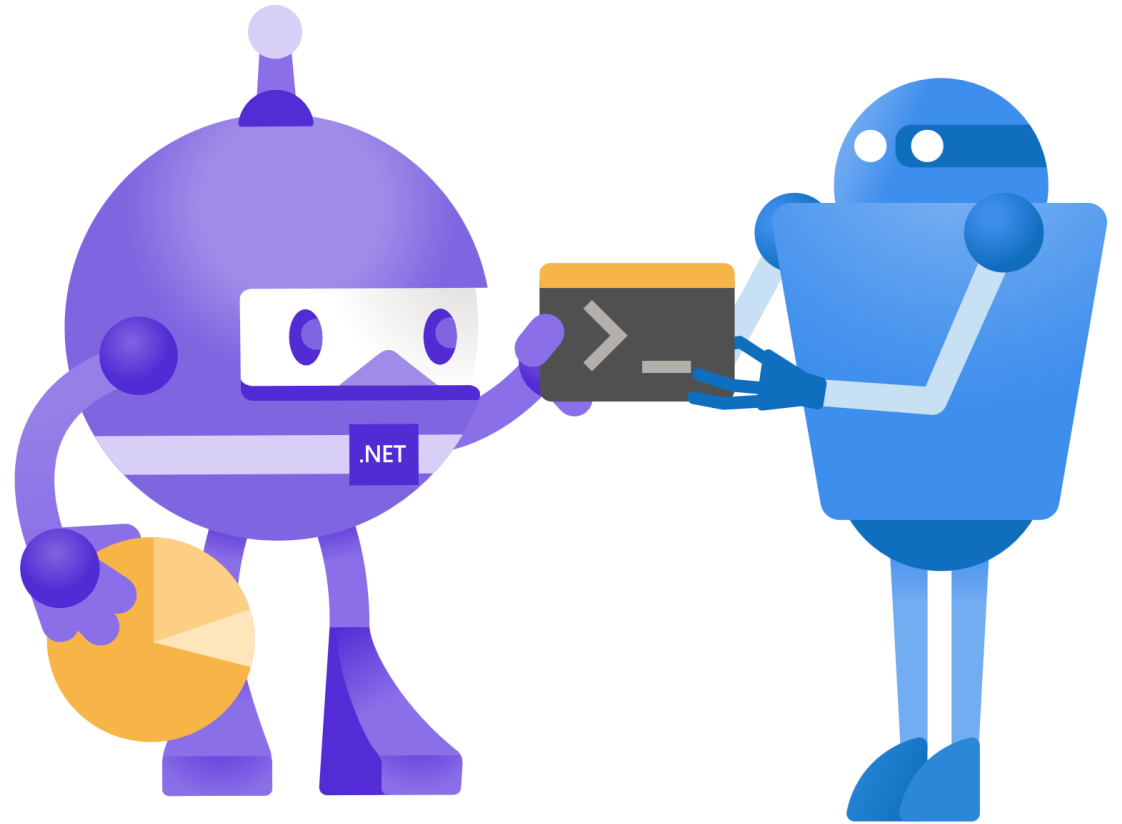
- **Non-UI specific packages that are built off of .NET Standard should just work**
  - Expection – Xamarin.iOS dependencies need to be recompiled
- **Libraries that have UI components will need to update namespaces, add MAUI compatibility packages, and test their layouts**
- **This is going to be your bigger blocker to updating to MAUI – start your inventory now!**





# NuGet Libraries Supported Today \*\*\*\*\*

- David Ortinau's dev.to series:  
<https://dev.to/davidortinau>
- Table of Nugets : [Xamarin -> Maui Libraries](#)
- <https://github.com/jsuarezruiz/awesome-dotnet-maui>



# I maintain a library! What do I do?!

- **Email Maddy at [maddy@microsoft.com](mailto:maddy@microsoft.com)**
- Thank Javier Suárez for being amazing  
<https://github.com/jsuarezruiz/mvpsummit2022-dotnet-maui>
- <https://devblogs.microsoft.com/xamarin/tips-for-porting-your-xamarin-library-to-dotnet-maui/>
- Samples :  
[.NET MAUI Community Toolkit](#)  
[In-App Billing Plugin](#)  
[SkiaSharp](#)  
[Zxing.Net.Mobile](#)

# I have a ton of custom renderers! What do I do?!

- Checkout this Wiki : <https://github.com/dotnet/maui/wiki/Porting-Custom-Renderers-to-Handlers>
- Gerald Versluis has a great video on it  
<https://www.youtube.com/watch?v=EKznvARe8o>
- Checkout Javier Suárez's amazing repositories:
  - <https://github.com/jsuarezruiz/awesome-dotnet-maui>
  - <https://github.com/jsuarezruiz/mvpsummit2022-dotnet-maui>
- Sam Basu + Javier : [https://www.youtube.com/watch?v=6PvL398I\\_y8](https://www.youtube.com/watch?v=6PvL398I_y8)

There is this random 8000000 year old NuGet we rely heavily on. What do we do?!?!



---

**Best option: Replace with a modern version**

---

**2<sup>nd</sup> best option: Reach out to the maintainer**

---

**3<sup>rd</sup> best option: Fork it and DIY**

---

**4<sup>th</sup> option : it's okay to cry.**

# Pick your own migration adventure!



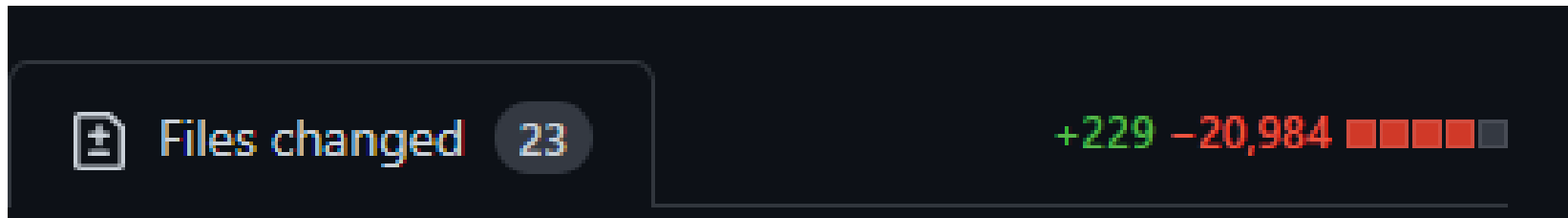
upgrade-assistant

Manual migration



# Manual migration

- [Get Started with Migration Doc](#)
- [PlantLady](#) Migration Branch



# .NET upgrade-assistant support for .NET MAUI

CLI tool that **assists** you upgrading your Xamarin.Forms apps to .NET MAUI



.csproj  
updates  
(moving to  
SDK style  
projects)



Source code  
updates (C#  
and XAML  
namespaces,  
APIs)



NuGet analysis  
and updater

**\*Currently under development!!!!**

# Upgrade Assistant support for .NET MAUI

- Types of upgrades to-be-supported:
  - Xamarin.Forms -> .NET MAUI (not single project!)
  - Xamarin native -> .NET native (basically a fancy try-convert!)
  - (Probably) Libraries
  - (Maybe) Native binding libraries

Also, it **will** work on the Mac! Support coming soon!

# It does (almost) all of this!

1. Backup project
2. Convert project file to SDK style
3. Update target framework monikers
4. Update NuGet packages to compatible ones
5. Add template files
6. Migrate app config files
7. Update C# source
8. Update XAML source
9. Migrate custom renderers
10. **Move to next project**
11. Test your layouts

## Before

Xamarin.Forms project  
(.NET Standard)

Xamarin.Android project  
(Mono.Android)

Xamarin.iOS project  
(Mono.iOS)

## After

.NET MAUI project (.NET 6  
with multitargeting)

.NET Android project  
(net6.0-android)

.NET iOS project  
(net6.0-ios)

# Expected Limitations

- **UWP (and other non-.NET 6 platforms) are not supported**
  - Support coming soon!
- **Probably will require some manual cleanup (we'll have blogs and docs, don't worry!)**
- **Will not get rid of “head” projects for you**
  - Won't move resources to MauiImage etc.
  - Won't move platform-specific code
  - Won't move things into startup for you
- **Will not build/run immediately after running tool**
  - Expect few deletions/additions to csproj
  - Expect to adjust resources
  - Expect a few build errors
- **Please engage with us on GitHub, Twitter, etc!**



# Expected Limitations

- **UWP (and other non-.NET 6 platforms) are not supported**
  - Support coming soon!
- **Probably will require some manual cleanup (we'll have blogs and docs, don't worry!)**
- **Will not get rid of “head” projects for you**
  - Won't move resources to MauiImage etc.
  - Won't move platform-specific code
  - Won't move things into startup for you
- **Will not build/run immediately after running tool**
  - Expect few deletions/additions to csproj
  - Expect to adjust resources
  - Expect a few build errors
- **Please engage with us on GitHub, Twitter, etc!**

# Let's migrate an App!



# Help us help you

- **What you can do today is go to [aka.ms/migratetomaui](https://aka.ms/migratetomaui)**
  - This includes the link to the manual migration steps
- **Please try it out! It doesn't have to be good – or work – or anything! Then create an issue to let us know how it went.**
- **We will use this feedback to**
  - Make Upgrade Assistant more robust
  - Help unblock you as we can
  - Identify popular NuGets not on our radar
  - Create better/more thorough docs, blogs, guides, etc

# Thanks for Listening!!

[maddy@microsoft.com](mailto:maddy@microsoft.com) & [swsat@microsoft.com](mailto:swsat@microsoft.com)



[https://www.theurlist.com/devreach\\_2022\\_migratemaui](https://www.theurlist.com/devreach_2022_migratemaui)

