TeamViewer – Innovation Case

Intro

The Team behind the new Product from TeamViewer called Blizz decided to publish their application in the Windows Store as an additional distribution channel. The cross-platform Client didn't target the Windows Store as a platform and was ported with the help of the Desktop Bridge Centennial.

Core Team

- Constantin Comendant, Software Engineer, TeamViewer
- Rainer Ruoff, DevOps Engineer, TeamViewer
- Andreas Pohl, Technical Evangelist, Microsoft

Customer profile

TeamViewer focuses on cloud-based technologies to enable online support and collaborate in real time across the globe. Remote support, remote access, and online collaboration are not mere buzzwords. They represent helping people, better integrating technology into our daily lives, and creating new ideas.

People have collectively used the technology from TeamViewer in billions of instances where distance and time would have otherwise prevented them from accomplishing their goals. TeamViewer:

- has been installed on over 1 billion devices (each device generates a unique ID)
- creates 750,000 new IDs every day.
- has over 20 million devices online at any given time

These astonishing numbers have led 90% of Fortune 500 companies to rely on TeamViewer as their choice to bring colleagues together across all platforms and all devices.

Problem statement

TeamViewer started developing their new Product for team collaboration called Blizz as a cross platform application. They already had their Product TeamViewer available as an UWP Application published in the Windows Store and wanted to publish their new product there as well. Because the development of Blizz already started as a cross platform application targeting Win32 we wanted to avoid implementing a new target and decided to convert the existing application with the help of the Desktop Bridge Centennial.

Solution, steps, and delivery

TeamViewer planned to convert Blizz in a 2-week sprint but only needed a day to set up Centennial and configure their continuous Integration Pipeline to adopt UWP as a new Platform for Blizz.

During the conversion process the team run into some limitations that needed to be addressed to make Centennial work with Blizz.

Auto Update

Blizz has the feature to Auto update the application once there is a new version available. In order to do this the application, need elevated write access which is not possible if you want to convert

a application using Centennial. In this case, it wasn't a problem to deactivate the Auto Update in the application because the Windows store is able to update the UWP application of their users. This way the functionality of the Auto update was still in place but the application wasn't responsible for it anymore.

Plugins

Another challenge where the Plugin capability of Blizz. With this you can add any plugins to Blizz to enhance the capabilities of the application. The Plugin functionality is implemented in a way that the application downloads the plugins and starts using them. Because Blizz doesn't have Elevated write access as an UWP that has been generated with Centennial the team needed to deactivate this feature in the first iteration. A look in the Centennial Documentation indicates where the limitations of Centennial are today and give a hint how to solve these kind of problems in your app. In this case this would mean that the team needed to change the behavior of the application which they decided against because they wanted to get in the Windows Store as soon as possible.

Integration into Jenkins

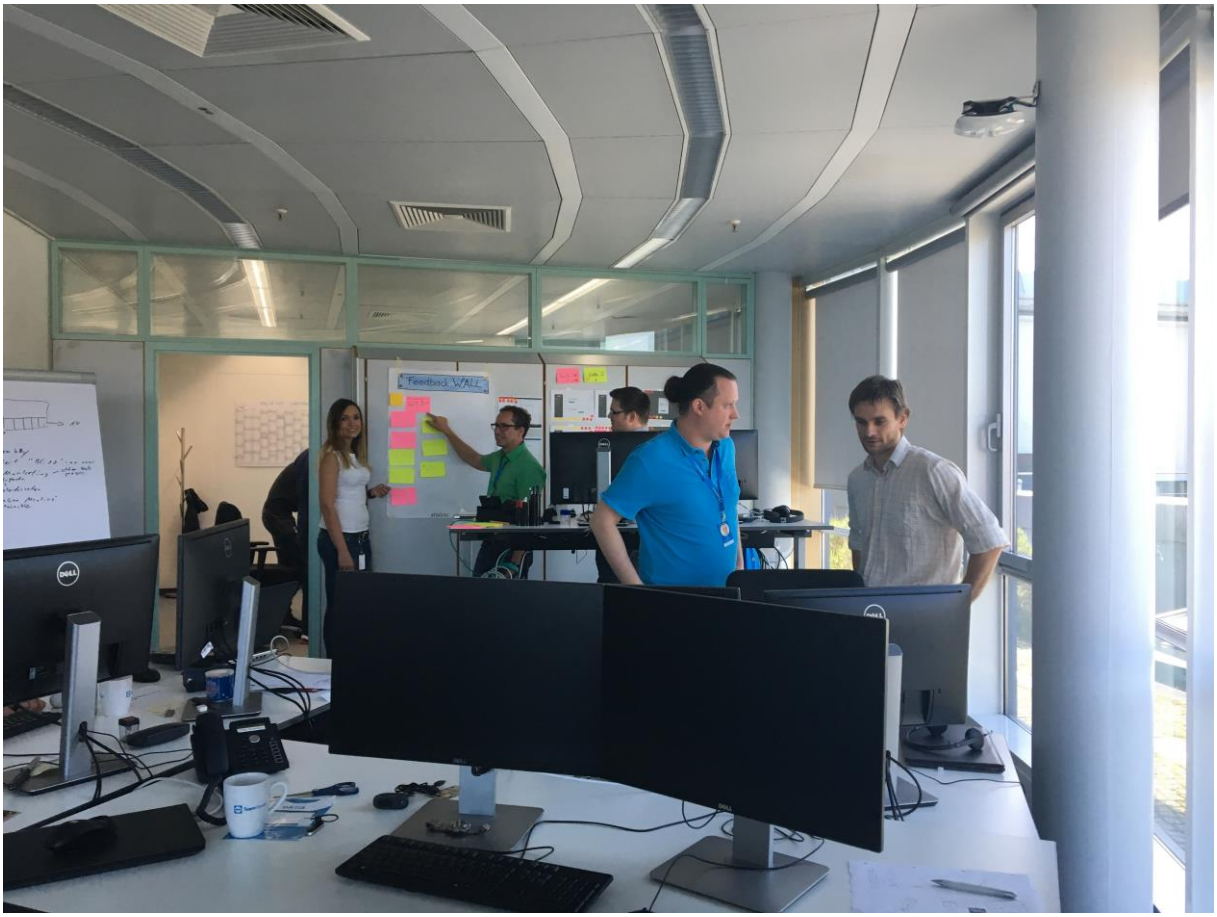
TeamViewer has modern and agile process which includes DevOps Practices like continuous Integration. Because of this it was very important to have the conversion with Centennial and the publishing in the Store automated. Integration Centennial into Jenkins was easy, the job needs to be run on a Windows 10 PC with the latest creators update for the most recent version of Centennial. Everything else is handled by a simple PowerShell script which looks like this

```
#
# convert Blizz setup to uwp application
#
$WorkSpace="$env:WORKSPACE"
$SetupName="Blizz_Setup.exe"
$SetupDir="$WorkSpace\setup"
$SetupPath = "$SetupDir\$SetupName"
$AppxOutDir = "$WorkSpace\appx"

$TargetDirRoot = "\\some\path\to\Release Candidate Centennial appx"

# retrieve version information from setup
$FileVersion   =(Get-Item $SetupPath).VersionInfo.FileVersion
$ProductVersion=(Get-Item $SetupPath).VersionInfo.ProductVersion

$TargetDir = "$TargetDirRoot\$ProductVersion"

$ArgList = "-Installer " + "`"$SetupPath`" "`
+ "-Destination " + "`"$AppxOutDir`" "`
+ "-Version " + "$FileVersion "`
+ "-PackageName TeamViewer.BlizzMeeting "`
+ "-PackageDisplayName `"Blizz`" "`
+ "-Publisher `"publisherKEY`" "`
+ "-PackagePublisherDisplayName `"TeamViewer`" "`
+ "-MakeAppx -Verbose"

# delete old package conversions
Remove-Item "$AppxOutDir" -Recurse -Force

# call the desktop app converter
$proc = Start-Process -Verbose -Wait -NoNewWindow -PassThru -FilePath
DesktopAppConverter.exe -Argumentlist "$ArgList"

# copy results to storage
Copy-Item -Recurse -Force -Path $AppxOutDir -Destination "$TargetDir"
Copy-Item -Recurse -Force -Path $SetupDir   -Destination "$TargetDir"

# delete Blizz setup
```

```
Remove-Item $SetupDir -Force -Recurse

exit $proc.ExitCode
```

Conclusion

Over all the process with Centennial is really easy and integrates great into already established processes. The most important part is to have a look at the Pre Check List for Centennial to discover possible problems as early as possible.



With Centennial TeamViewer was able to distribute Blizz through an additional channel, the Windows store, which is essential to a newly launched product.

For the future the team is looking into adopting the new possibilities like notifications that are available with UWP.

“The blizz online meeting app download from the trusted Windows 10 app store continues TeamViewer´s tradition of offering secure, reliable meeting technology – with easy access for all.” - Holger Jung, Product Owner, TeamViewer