HCLTech | Supercharging Progress™

Why is Windows Presentation Foundation (WPF) still relevant?

Table of contents

Abstract

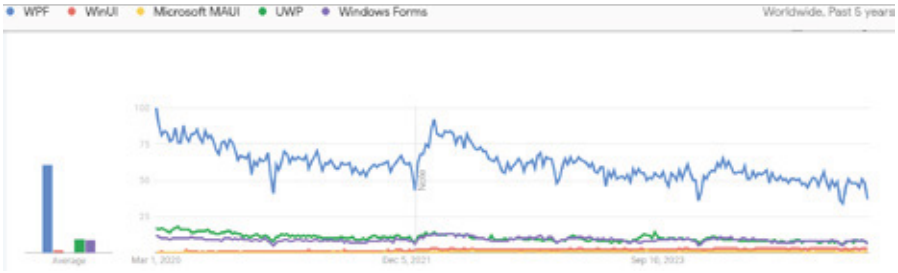
In 2006, Microsoft launched Windows Presentation Foundation (WPF) to provide developers with a tool for creating visually rich, interactive and data-driven desktop applications for Windows operating systems. Although WPF is an old technology, it is still a formidable technology that is widely used in various industries. This writing explores the state of the WPF framework and why it is still widely used and industries using the technology.

Introduction

Since its inception in 2006, WPF has been an integral tool for developing Windows applications. Many developers have been using the framework to create business-critical applications because of its ability to build rich, interactive and visually appealing desktop applications. Besides visual appeal, software designers and developers have been using the technology because of its unique features, such as separation of concerns, resolution independence and ability to facilitate rapid software development. Although WPF is an impressive and mature technology for developing Windows applications, there are claims that its use is declining and some quarters claim it is no longer a relevant technology in the modern world. This piece of writing demystifies the use of WPF in the contemporary world and explicates why it is still a technology to reckon with today. Also, the writing discusses some industries that heavily rely on technology.

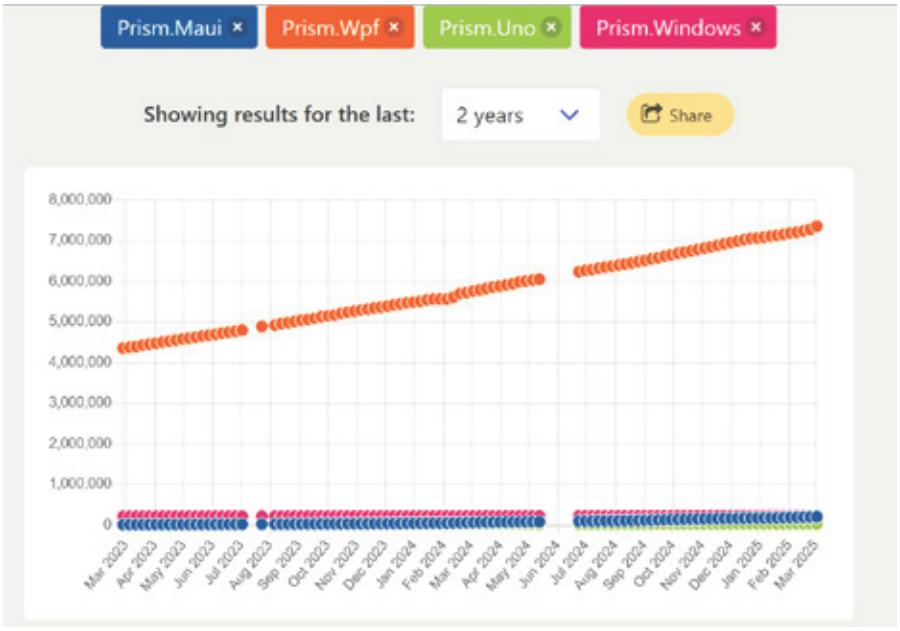
State of WPF

WPF continues to be one of the most popular frameworks for developing enterprise desktop applications. Today, about one million active applications worldwide use WPF technology. To gauge the actual popularity of WPF, let's crunch some numbers. On GitHub, WPF has over 7,200 stars, 1,200 forks, 142 contributors and over 6,200 commits spanning five years (GitHub, 2025). Using Google Search metrics, WPF is still one of the most popular frameworks for developing desktop applications. Google Search insights indicate that the number of WPF searches has dramatically decreased. However, WPF is the most searched compared to competing technologies such as Windows Forms, MAUI and WinUI (Google Trends, 2025).

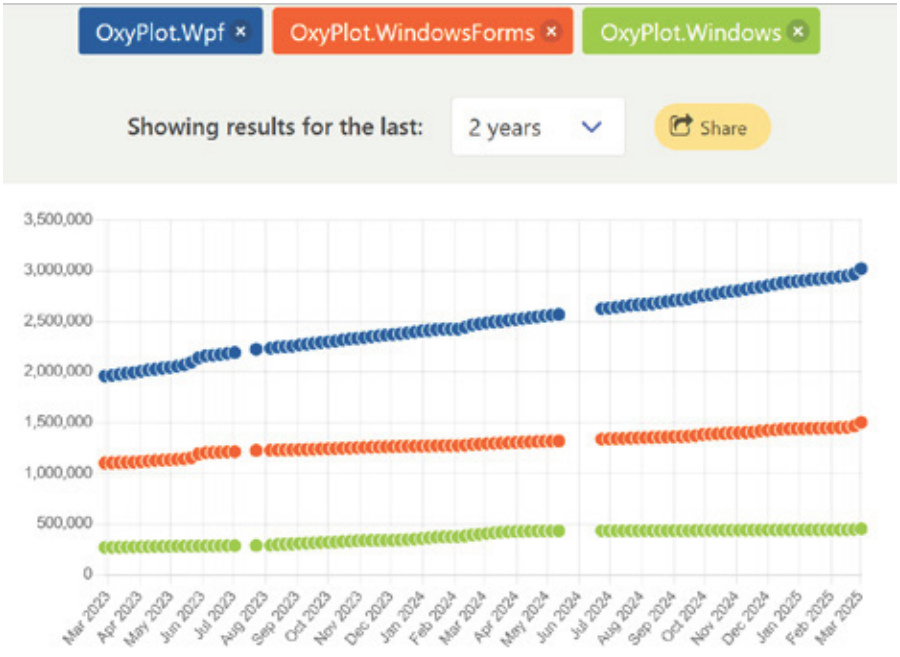


WPF is the most searched in its niche. Source: Google

The popularity of WPF can also be measured by assessing the number of downloads for its packages. Some of the most used WPF packages include WPF Toolkit, Prism, Oxyplot and Sciechat. WPF Toolkit, which is a collection of open-source controls such as DataGrid, ColorPicker and charts, has 12 million downloads and continues to uptrend. Prism, a dependency injection framework with WPF and other frameworks libraries, has close to 7.5 million downloads. Compared to other frameworks, as demonstrated in the graph below, Prism's WPF libraries are the most downloaded and up-trending (Nuget Trends, 2025). Similar to Prism, Oxyplot has libraries for multiple frameworks. WPF libraries have about 3.1 million downloads and are the most downloaded, as illustrated in the graph below. ScieChart, which contains high-performance chart controls, has over 7,00,000 downloads, which continues to trend upwards.



Prism's WPF libraries are the most downloaded and up-trending. Source: Nuget Trends



Oxyplot's WPF libraries are the most downloaded. Source: Nuget Trends

Developers are not just using the WPF framework. Since it was partially open-sourced, the user community has been working on projects meant to improve the capabilities and features of the WPF framework. For example, Avalonia XPF, which is a community-driven project, aims to make WPF a cross-platform technology. Avalonia XPF will make WPF applications deployable on both Windows and MacOS operating systems. The Avalonia team, which is a group of developers behind XPF, also aims to make WPF compatible with mobile operating systems (Android and iOS) and web browsers. In an interview published by DotNet (2023), Olia Gavrysh, .NET Product Manager, says, "We think of WPF as a very mature project... We're going to support this product as long as we have users for that product..." From this response, it is evident that Microsoft considers WPF a mature project and is committed to continuing to support it. With strong support from the user community and Microsoft, WPF's existence is guaranteed and its features will keep improving.

WPF strengths over emerging frameworks

Various frameworks promise to offer better features than WPF. However, as demonstrated in the previous section, many developers still favor WPF over competing models. This is primarily because WPF was designed to provide developers with a powerful, flexible and modern way to create desktop applications for Windows and none of the emerging technologies has done it better than WPF. Some of the areas in which WPF outshines competing technologies are as follows:

1. UI and customization capabilities

WPF is specifically designed to enable developers to create rich and visually appealing user interfaces. Using XAML markup language, developers can conveniently create customized, visually appealing controls. Key UI features that distinguish WPF from competitors include:

- **Resolution independence:** WPF applications use DirectX components, allowing them to readjust resolution based on the screen size. The ability to autonomously resize resolution makes WPF applications deployable on a wide range of screens without worrying about loss of resolution quality.
- **Controls inside controls:** The framework allows users to add controls inside other controls. For example, developers can add a textbox inside a button. This feature was not present in older development environments.
- **Data binding:** In the modern world, where data is a crucial resource, data binding is vital for any application. WPF allows convenient data connection to controls.
- **2D and 3D animation:** WPF permits developers to create 2D and 3D graphical elements. 2D and 3D elements can also be designed to have animation features. The framework is crafted to ensure seamless implementation of animation in 2D and 3D graphics.

- **Controls styling:** WPF has an elaborate feature that allows UI designers to style controls to their liking. They can define the appearance of controls and other elements by stipulating attributes such as color and size and giving them custom shapes.

2. Separation of concerns

This feature separates software into two parts: user interface and behavior. The user interface is implemented using XAML markup language and behavior is implemented using Code-behind. The idea of separation of concerns offers the following benefits:

- **Decouples appearance-specific code from behavior-specific code,** reducing development and maintenance costs.
- **Permits separation of roles in development:** UI designers can implement application interfaces as developers implement app functionality simultaneously.
- **Separation of markup and code-behind** simplifies the globalization and localization of WPF applications.

3. Performance benefits

WPF employs DirectX for rendering. DirectX offers better performance than GDI-based applications. Benefits WPF from DirectX include:

- **Smooth rendering of complex UI:** DirectX makes WPF ideal for applications that require high-performance graphics, such as dashboards and visualization tools.
- **Efficient resource utilization:** WPF leverages GPU for rendering. This reduces the workload on the CPU.

4. Integration with modern .NET versions

WPF has been integrated into modern .NET versions such as .NET Core and .NET 8. Integration with modern versions equips WPF with the following:

- **Enhanced security:** Integration with modern versions improves the security of WPF. Modern versions are designed to address modern security risks.
- **Cross-platform capability:** WPF was strictly designed for Windows applications. However, integration with modern versions enables it to work in hybrid environments alongside frameworks such as MAUI and Blazor.

5. Strong ecosystem and community support

Being one of the most mature frameworks in its niche, WPF has a strong community that ensures WPF developers have access to ready support throughout their development journey. This support comes in the form of the following:

- **Rich libraries:** WPF has third-party libraries such as Telerik, DevExpress and Syncfusion, which are packed with advanced UI components.
- **Robust developer community:** WPF has a plethora of forums, blogs and courses for supporting developers and enhancing their skills.

6. Advanced stability compared to newer frameworks

Although newer frameworks are promising, they cannot compete with WPF in terms of:

- **Maturity and stability:** WPF is a mature technology that is hardly evolving. New frameworks are constantly evolving.
- **Feature completeness:** WPF offers a rich set of advanced features for desktop application development. WPF's features are fully developed, making it suitable for developing enterprise applications.
- **Large-scale app development:** WPF was released in 2006 and for the past 20 years, the technology has been fine-tuned and advanced, making it ideal for developing large Windows applications.

Industries using WPF

WPF is a mature technology used in virtually all industries. However, due to its unique features and capabilities, the technology is extensively used in specific industries. Leading industries that use WPF are:

Manufacturing industry: This includes industries such as chemical producers, textile companies and plastic producers. The manufacturing industry is one of the leading sectors actively running WPF applications.

Banking and financial services: WPF is commonly used for banking applications because of the nature of the sector. The banking and financial industries are highly regulated and needs top-notch security and institutions are distributed across the globe. WPF applications are very secure and most banks worldwide use Windows systems, designating WPF as an ideal technology for connecting global banking systems for finance institutions.

Health industry: WPF is commonly used in applications in health institutions. Most health records systems and patient management systems are developed using WPF.

Defense industry: WPF technology is used in the development of defense systems in Western countries.

According to Progress Telerik (2013), some of the WPF attributes that make it desirable for use in these industries include:

- It is a mature system that was developed in 2006. This implies it is secure and can run on virtually all Windows systems, including Windows XP and Windows 7.
- It has a robust UI that leverages vector graphic support, making it ideal for developing applications compatible with different OS and devices.
- It has full access to the Windows file system, making it ideal for developing applications that handle various connected locations.
- WPF systems can handle huge transactions simultaneously.
- It has a robust data binding mechanism that supports the separation of concerns, enabling designers and developers to work independently.

Conclusion

Although WPF is a mature technology, it is still one of the most used frameworks for building desktop applications. Insights indicate that a significant number of developers use WPF and the number is growing. WPF is still relevant today because of its stability compared to emerging technologies, integration with newer .NET versions, strong community support, performance benefits, advanced UI features and suitability to develop large enterprise applications. These capabilities have rendered WPF suitable for industries that emphasize security and reliability. Are you thinking of developing a Windows desktop application? You should consider the WFP framework.

References

Microsoft Learn (2024), Desktop Guide (WPF .NET). Retrieved From: https://learn.microsoft.com/en-us/dotnet/desktop/wpf/overview/?view=netdesktop-9.0

Dotnet (2023), Desktop Community Standup - News from WPF and WinForms team. Retrieved From: https://www.youtube.com/live/jYq_lVxpcTQ?t=1210s

Nuget Trends (2025), Retrieved From: https://nugettrends.com/packages?months=24&ids=Prism.Wpf&ids=Prism.Maui&ids=Prism.Uno&ids=Prism.Windows

Nuget Trends (2025), Retrieved From: https://nugettrends.com/packages?months=24&ids=OxyPlot.Wpf&ids=OxyPlot.WindowsForms&ids=OxyPlot.Windows

Google Trends (2025), Retrieved From: https://trends.google.com/trends/explore?date=today%205-y&q=WPF,WinUI,Microsoft%20MAUI,UWP,Windows%20Forms&hl=en

GitHub (2025), dotnet/wpf. Retrieved From: https://github.com/dotnet/wpf/

Author information

Sachin Kumar

Sachin has over 28 years of experience in the IT industry working on Microsoft Technologies, Product Security and AI. He works for HCLTech ERS as Senior Solution Director. He has diverse experience in emerging technologies and has worked for a long time in the Fintech domain for a big Swiss bank in Europe. He holds a Bachelor of Technology degree in Electrical Engineering. He is an alumnus of IIM Kozhikode, where he completed his executive MBA specializing in IT management and Finance. He holds multiple certifications, such as Microsoft Azure AI and TOGAF.

Jitender Chandwani

Jitender has spent over 16 years in the IT sector. He owns skills in .Net technology for robust web and desktop application development. His journey has involved engaging with prominent industry leaders, where he has delivered highly scalable and impactful solutions. Furthermore, he contributed as an inventor to US Patent US12143285B2, a testament to his involvement in developing a critical tool for Controller Area Network and connectivity health troubleshooting.