

УДК 004.4

V. A. Antonyuk, M. G. Sydorova

Oles Honchar Dnipro National University

A CROSS-PLATFORM MOBILE DEVELOPMENT FOR ACCELERATING SOFTWARE DEVELOPMENT LIFECYCLE

Однією з найважливіших частин кожної програми є вибір мови та фреймворку, на якому вона буде розроблятися. Але досить складно визначити, яким має бути хороший фреймворк для розробки програмного забезпечення, адже на це питання немає єдиної правильної відповіді. В цілому, він повинен робити програмні продукти дешевшими у розробці та обслуговуванні, бути гнучким при виявленні помилок, розширюваним та придатним для довгострокового використання, адаптивним до вимог та змін; при додаванні функцій продуктивність не повинна знижуватися. Хороший фреймворк допомагає підтримувати якість програмного забезпечення протягом усього терміну його служби. Однак навіть найкращі проєкти зазвичай зазнають труднощів при виборі правильного фреймворку на початковому етапі. Завжди важливо думати про довгострокову перспективу, створюючи програмне забезпечення, щоб зробити його функціональним одразу і в той же час дозволити підтримку розвитку та будь-яких змін. Це може заощадити час розробників, обслуговуючи мінливі вимоги клієнтів. Вибір правильного фреймворку має вирішальне значення, тому що після створення продукту його вже досить складно і дорого змінити. У цій статті розглядається крос-платформний фреймворк .NET MAUI. Проаналізовано його переваги порівняно з іншими фреймворками. Ця інформація може допомогти при виборі фреймворку при розробці власного продукту. Також детально розглянуто його застосування на прикладі розробки авторського кросплатформного додатка “Draw & GO”.

Ключові слова: *.NET MAUI, Draw & GO, мобільний додаток, крос-платформна розробка.*

This paper presents and discusses the benefit of cross-platform development. The purpose is to demonstrate the advantages of the cross-platform framework .NET MAUI. The mobile application was designed according to a cross-platform approach to reach the broadest possible audience of customers, saving time and effort in development and maintenance. The code

is fully shared between mobile platforms (Android, iOS, macOS, Windows, and Web) allowing customers to use the application on any device.

Keywords: *cross-platform application development, mobile application, .NET MAUI, Draw & GO*

Introduction. In today's world, technologies are developing very fast. Every day, thousands of software updates are released in order not to lose the market and remain competitive.

Most projects have certain deadlines, budgets, and scope of work. This trio of time, money, and scope is called the project triangle because changes to one of these elements change both others. Quality, the fourth element of the design triangle, is at its center, and changes made to any side of the triangle almost always affect quality. Quality is not a side of the triangle – it is the result of what is done with time, money, and scope of work.

Every business wants to release a product in the shortest possible time (to remain competitive), with a lot of new functionality (to impress users with new features), with minimal budget expenses (since the main goal of any business is earnings, which should exceed expenses, to be profitable), and with high quality (the fewer defects – the more likely the user will continue to use this product). However, if you change the project plan, reducing the time, then the cost of the project will increase (additional workers need to be involved) or the scope of work will be reduced. If the project plan is changed to reduce its budget, the duration of the project may increase and the scope of work may decrease. Finally, if you increase the amount of work, the project will last longer and cost more.

One of the solutions is to create a cross-platform application. By allowing your application to run on different systems, you can see a noticeable increase in the number of users. The benefit from the implementation of platform-independent applications is immediately noticeable – development time is significantly reduced, since there is no need to adapt the code multiple times for various operating systems, and no need to know the nuances of each of the platforms. Along with this, the quality of the software product increases significantly, as it will be tested on several platforms at the same time, and errors will be corrected in the same source code of the program.

Developing the cross-platform application “Draw & GO”, helps to automate all daily work with the user's devices, recording and playing macros with the help of a graphic key, when choosing approaches and technologies to accelerate its development lifecycle and increase its competitiveness in the market the considered problem was carefully analyzed and taken into account.

Statement of the problem. This paper is devoted to demonstrating the benefits of cross-platform development, introducing a cross-platform mobile development approach, presenting the existing cross-platform frameworks, discusses the usage and advantages of cross-platform framework .NET MAUI in developing own application “Draw & GO”.

Main material. The question of a choice corresponding framework may occur before developing a cross-platform application. The following frameworks can be distinguished among the most popular:

- .NET MAUI is the .NET Multi-platform App UI, a framework for building native device applications spanning mobile, tablet, and desktop [1];
- NativeScript is a JavaScript-native mobile framework [2];
- React Native is a framework for building native applications using React [3];
- Flutter makes it easy and fast to build beautiful apps for mobile and beyond [4];
- Ionic is a powerful cross-platform UI toolkit for building native-quality iOS, Android, and Progressive Web Apps with HTML, CSS, and JavaScript [5].

In this work, preference was given to .NET MAUI (.NET Multi-platform App UI) – a framework for creating multi-platform applications for Android, iOS, macOS, Windows, and Tizen – due to the following reasons:

1. A single project for different platforms. .NET MAUI allows developers to manage all platforms in one project. This means that there won't be multiple projects for each target platform. In .NET MAUI shared resource files are within the same project. A single application manifest that defines the application's name, ID, and version. A single, cross-platform entry point to the application that helps develop projects faster and is much easier to maintain. This makes “Draw & GO” easier to release updates and maintain the same version for different platforms [6].

2. A single place to manage all resources. .NET MAUI manages fonts, images, application icons, screensaver, raw resources, and styles from a single place – the “Resources” folder. .NET MAUI optimizes them according to the unique requirements of each platform, which significantly saves time and resources on support. “Draw & GO” uses SVG images, which are automatically converted to the required images for each platform.

3. Full support for Hot Reload. Hot Reload allows developers to make changes to the code and apply them to the running program. Recompile is not required, and whenever possible, the state of the program is preserved

unchanged. This significantly speeds up the development of the program. Since “Draw & GO” is quite a big project, rebuilding the application usually takes some time. With Hot Reload, changes happen on the fly [7].

4. Blazor support. .NET MAUI is also great for web developers who want to run web code in native client applications. Blazor is ideal when you need to create a website and mobile app that look identical. Developers can also use their web development skills to create cross-platform custom client applications for mobile, desktop, and the web. The web application “Draw & GO” is developed with Blazor. UI framework elements, such as Flow Diagrams used in the mobile application use the same component as the web application (fig. 1).

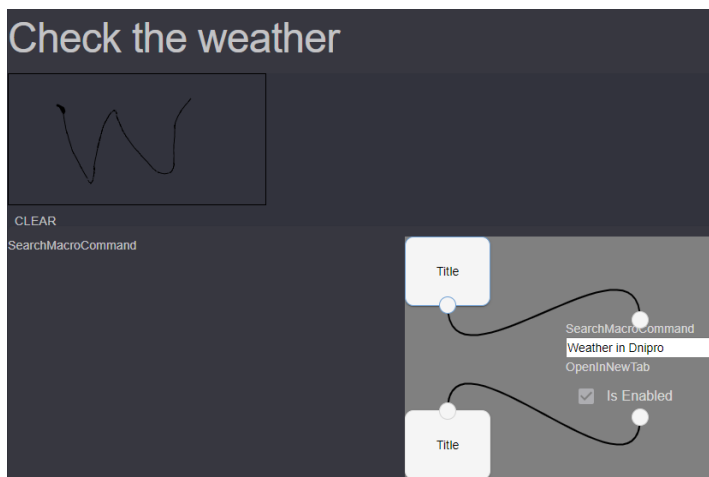


Figure 1 – Inject Blazor in .NET MAUI app

Association of libraries. .NET MAUI makes it easy to use device capabilities such as sensors, photos, contacts, and other services that the customer uses regularly, such as authentication, notifications, Bluetooth, geolocation, file system, and more. “Draw & GO” actively uses authentication, notifications, and integration with the web application, while at the level of abstractions, without the need for implementation for each platform [8].

5. .NET MAUI comes with more than 40 UI components out of the box. One major advantage of using native UI is the inherited accessibility support. It makes it easy to create highly accessible applications with

semantic services. “Draw & GO” was developed with the possibility of use by people with disabilities.

6. There’s a good improvement in performance and reduced app size after .NET 6 transition. The team reduced the Java <=> C# call cycles. This also affected the size of the app. It was possible to reduce the size by 15% (47MB) and speed up the launch by 30% (514ms).

7. .NET MAUI is architected for extensibility. Take, for example, the Entry control — a canonical example of a control that renders differently on one platform. Android draws an underline below the text field, and developers often want to remove that underline. With .NET MAUI, customizing every Entry in your entire project is just a few lines of code [9] (fig. 2).



Figure 2 – Custom control DrawingView

Rich components, accessibility first principle, speed optimization, platform-specific customization, and fast development made it possible to finally choose .NET MAUI when developing own product.

Conclusions. Cross-platform development significantly saves time and resources, in turn reducing the amount of work and improving the quality of the final product.

Using cross-platform development with .NET MAUI allows building native apps for Android, iOS, macOS, Windows, and Tizen from a single codebase, which, along with single-project features, allows to focus on a single app instead of juggling the unique needs of multiple platforms.

.NET MAUI allowed the development of a product “Draw & GO”. “Draw & GO” is a fully cross-platform product (Android, iOS, macOS,

Windows, Web) with the function of synchronizing macros between several devices. “Draw & GO” uses a plugin system, so it has an unlimited number of features. Also, “Draw & GO” accelerates the search and launch of macros due to a person's psychological tendency to associative memory, which is why each macro is associated with a graphic image. To start the macro, it is enough to draw the image. “Draw & GO” is still under development. Try the web version by reference [10].

Bibliographic references

1. .NET MAUI documentation. URL: <https://learn.microsoft.com/en-us/dotnet/maui/?view=net-maui-7.0/> (дата звернення: 16.11.2022)
2. NativeScript documentation.
URL: <https://docs.nativescript.org/> (дата звернення: 16.11.2022)
3. React Native documentation. URL: <https://reactnative.dev/docs/getting-started/> (дата звернення: 16.11.2022)
4. Flutter documentation.
URL: <https://docs.flutter.dev/> (дата звернення: 16.11.2022)
5. Ionic documentation.
URL: <https://ionicframework.com/docs/> (дата звернення: 16.11.2022)
6. **Goldman M.** .NET MAUI in Action. *Manning Publications*. 2022. 350 p.
7. **Stonis M.** Enterprise Application Patterns using .NET MAUI. 2022. 101 p. URL: <https://aka.ms/maui-ebook/> (дата звернення: 16.11.2022)
8. **Yassine A., Berrada M., Tahiri A., Chenouni D.** A cross-platform mobile application for learning programming basics. *International Journal of Interactive Mobile Technologies*. Vol 12(7). 2018. P.139-151. DOI: doi.org/10.3991/ijim.v12i7.9442
9. **Ortinau D.** Introducing .NET MAUI – One Codebase, Many Platforms. 2022. URL: <https://devblogs.microsoft.com/dotnet/introducing-dotnet-maui-one-codebase-many-platforms> (дата звернення: 12.10.2022)
10. Draw & GO URL: <https://drawgo.azurewebsites.net> (дата звернення: 16.11.2022)

Надійшла до редколегії 16.11.2022