# Heeft Blazor een toekomst?

## Doel

Voordat we definitief kunnen zeggen of we Blazor gaan gebruiken voor het maken van onze apps, is het van groot belang om te weten wat de ‘houdbaarheid’ is. Natuurlijk is het in deze wereld moeilijk om een beeld te schetsen hoe lang iets meegaat, maar ik ga door middel van dit onderzoek toch een beeld proberen te schetsen.

## Strategie



Literature study (library) 1

Zoals aangegeven in mijn projectplan ga ik tijdens dit onderzoek gebruik maken van de “Literature study” onderzoeksmethode, welke valt onder het “Library” veld.

Initieel was de strategie het opzoeken van roadmaps gemaakt door de ontwikkelaars van Blazor. Hierbij wilde ik door middel van (officiële) posts in kaart brengen wat er nog aan zat te komen betreft toevoegingen en updates in de toekomst. Echter heb ik hier tijdens mijn onderzoek van af moeten wijken. Wel ben ik trouw gebleven aan de onderzoeksmethode.

## Uitwerking

Wanneer ik zoek naar een roadmap kom ik er al snel achter dat er geen officiële roadmap is voor releases van Blazor. Ook kan ik op de officiële platforms geen informatie vinden over eventuele releases in de toekomst of eventuele toekomstplannen van Microsoft zelf.

Wel zie ik dat het een hele populaire manier is voor het maken van front-end applicaties door C# developers. Veel van diezelfde developers stellen dezelfde vraag: “Heeft het zin om in te zetten op Blazor, heeft het een toekomst”. Ik moet dus op een andere manier gaan zoeken naar mijn eigen antwoord, aan de hand van informatie die mij die keuze kan doen maken.

Aangezien de release van Blazor pas in 2018 was, is het een relatief nieuwe methode voor het ontwikkelen van front-end applicaties. Behalve dit is het ook de enige manier voor het maken van front-end applicaties in C#. Dit bekent dat er geen splitsing is tussen verschillende kampen, waar er gekozen kan worden uit meerdere mogelijkheden voor het maken van C# front-end applicaties. Dit spreekt er in het voordeel van Blazor, omdat iedereen dit bovenstaande wilt bereiken, toegewezen wordt tot Blazor. Wat zorgt voor een community waarvan de ogen allemaal dezelfde kant op staan

Ook omdat dit zo nieuw is, wordt er nog veel over gesproken op fora. Er wordt ook door die gebruikers veel gespeculeerd over de toekomst van Blazor, maar niemand kan een duidelijk antwoord geven. Vaak wordt er gesproken dat Blazor, aangezien het gebruik maakt van WASM (WebAssembly), een grote kans heeft langer te overleven dan bijvoorbeeld oudere flops van Microsoft (bijvoorbeeld Silverlight). Omdat WASM een nieuwe standaard is voor web-development, waarbij het zeer goed ontvangen wordt, lijkt dit een bevestiging waar ik naar op zoek ben.

## Wat zeggen anderen?

Om tot een goed beeld te komen ga ik op zoek naar wat, naar mijn idee, goede bronnen er over schrijven om vervolgens tot een conclusie te komen.

**Matthew MacDonald, 10.000 volger op Medium.com en Microsoft MVP**

“…The technological underpinnings for Blazor are rock solid. It’s built on WebAssembly, a cross-platform standard that has significant vendor buy-in, universal browser support, and better performance than even today’s optimized JavaScript engines. WebAssembly also has some jaw-dropping proof-of-concept demonstrations, like 3D games and emulators running natively in the browser…”

“Blazor is a tough sell to current web developer, because it means leaving behind many of the libraries and technologies that have been up over a decade of modern JavaScript development. It’s not a seamless transition, and there’s no way to migrate a JavaScript application to a Blazor project. Switching to Blazor still means moving from the rough seas of JavaScript to a much smaller swimming pool. Most developers are going to check it out but won’t be ready to leave the ocean just yet.

For experienced C# developers, it’s a different story. If you have a team that knows .NET and they’re building an internal business application, Blazor lets them be more productive and build the front-end and back-end with one stack. If you were considering a Windows desktop application (built on something like WPF), Blazor is a more future-forward choice. But if you’re a business building modern, public web applications, there’s very little chance you’ll recommend a client start with Blazor today.

The bottom line? Blazor’s a niche — a powerful one — best suited to Microsoft developers who love .NET and C#. But don’t let the present blind you to the future. There’s a lot of promise in stacks that build on WebAssembly, and a lot of room for other languages to nibble away at the edges of JavaScript. And in the ever-changing world of web development, we’re overdue for another revolution.”

Bron: <https://medium.com/young-coder/is-blazor-the-future-or-just-another-walled-garden-441842cc249d>

**Emergents Software, award-winning software bedrijf met o.a. Philips als klant.**

Deciding whether to use Blazor WebAssembly instead of one of the many JavaScript frameworks readily available largely depends on your team’s experience and comfort level with learning a new language. If you or your team is more comfortable coding in C# vs JavaScript, Blazor is a solid option for you. Keep in mind, though, that Blazor’s developer community is relatively small and its knowledge base is still maturing. A team who already has a solid grasp of JavaScript and project constraints like timeline or budget may decide to avoid the hassle of learning a new framework and stick to one of the tried and true JavaScript Frameworks available.

Finally, developers who have the time and desire to learn new technology should feel confident giving Blazor a shot at this point. Our team at Emergent Software has numerous successful Blazor projects under our belts and will continue to leverage it on projects Blazor fits well with.

While there’s little doubt to us that with each new release Blazor will continue to grow in popularity (and for good reason, too), we don’t see Blazor dethroning JavaScript in overall adoption any time soon. Even though Blazor has many clear and distinct advantages over JavaScript, JavaScript and its multiple, well-used frameworks that have stood the test of time, are here to stay for years to come. That said, the future appears to be bright for Blazor and we look forward to utilizing it on projects with goals that align well to what Blazor brings to the table.

Bron: <https://www.emergentsoftware.net/blog/is-blazor-better-than-javascript/>

## Conclusie

Mijn conclusie is, dat het samen met vele andere frameworks, moeilijk te zeggen is hoe lang de levensduur van Blazor is. Wat, zoals aangegeven, een groot voordeel is, is dat Blazor compileert naar de een nieuwe standaard WASM. Waarbij de verwachting is dat WASM alleen maar groter wordt en meer frameworks zich hier op gaan richten. Zoals ook door sommige ‘experts’ genoemd, is het als ervaren C# developer erg tijdbesparend om het wel te doen. Waarbij je als bedrijf zijnde zelf de afweging moet maken, aan de hand van het gevoel dat je er bij krijgt. Toch neig ik te zeggen dat we voorlopig niet bang hoeven te zijn dat Blazor ophoud met het krijgen van support. Als je ziet hoe groot het aan het worden is en het feit dat er tot nu toe nog productief aan ontwikkel wordt, denk ik dat we er voorlopig nog niet van af zijn.

## Bronnen

<https://medium.com/young-coder/is-blazor-the-future-or-just-another-walled-garden-441842cc249d>

<https://learn.microsoft.com/en-us/aspnet/core/blazor/?view=aspnetcore-6.0>

<https://learn.microsoft.com/en-us/answers/questions/708050/microsoft39s-long-term-support-lts-plans-for-blazo.html>

<https://www.ifourtechnolab.com/blog/is-blazor-the-future-in-web-app-development>