Jamie Rogers – Swiss .Net Software Recruiter
Jamie.rogers@darwinrecruitment.com / +41 41 506 2919

I think to most important stuff would be more and more the universal apps, because these are the apps at the end, which can executed on any client devices (smart phones, tablets, desktop...), and yes HTML5 to implement powerful webbrowser apps of course.

- "XAML and MVVM know how" (to implement Universal Apps, WPF)

- "HTML5 and javascript" (to implement webbrowser apps)

- any kind of webserver technologies such like MS Internet Information Server (to implement web server side applications)

I think its worth considering one of the biggest tech trends right now (which is where I am heading) is virtual reality. One of the market leaders for content development (game) engines is Unity and its scripting is based in C#. So users of that will have access to the .NET API and the .NET classes that go along with it. I don't think this will affect the C#/.NET community across the board so much, but its perhaps one direction to dig into further.

You will possibly see a surge in the desire for ASP.Net Core 1.0, as a retaliatory technology against node.js. (Web based) Owin / katana based skills for web dev's in migration projects, and continued migration to the azure platform for non-finance based devs. Entity framework and other such db tools will continue as-is for the more traditional practices. IoT and Xamarin tools may continue to present in the "add in" space...

External integration with client side tools like AngularJS may begin to fade - the fad All with the disclaimer that I'm based in the Suisse finance sector - not a web-development company near the Limehouse link etc.

The real challenge for .NET devs next year is certainly on the frontend side, either web and mobile: which technology stack is the most reliable and a good choice for the next 3 years? it's hard to say, and I guess nobody knows for real. so, the most important thing for a dev is to find an appealing stack, and start looking at it asap: it's important not to fall behind, and be ready when it will be the time to make important choices. the good side of being unemployed is that I have lots of time to study new things. I'm currently looking at ASP.NET Core in combination with node.js and Angular 2, because today is not possibile to think about a Microsoft-only frontend stack. mobile side I'm looking at Xamarin: it's very promising, and it really reduces development effort to deliver apps for Android and iOS (and hopefully Microsoft in the near future). backend and infrastructure side, I'd certainly look at Service Fabric: it's a killer service, it's easy to use and gives a Microsoft-based installation a boost toward microservices, containers and distributed architectures.

Jamie Rogers – Swiss .Net Software Recruiter
Jamie.rogers@darwinrecruitment.com / +41 41 506 2919

Since Microsoft is heavily investing in new things, these could be of interest: - .net core instead of now getting old full asp.net - angular 2 with typescript (typescript is from microsoft) - Azure Cloud - IoT - Virtual Reality with Microsoft Holo Lense i think, these are future techs comming from MS :) Especially the below one could be amazing :D

Hi Jamie, Interesting exercise! The trends from the .NET world would be .NET Core and Angular with more and more micro-services.

I've been programming for 13 years in C# and i haven't seen anything equal than the next year trending movements. First of all .Net core will open the mind of open-source minded developers, that are a lot in startups for example. Also Visual Studio Code and containers will help to that. But for me, .Net core should be relegated to back-end because is a nonsense using MVC instead of angular 1 or 2. So angular 2 i think should be the next trending in front-end. Azure is and will be trending but I don't know about next Google steps in the cloud thing, so maybe it can be changed. About mobile apps, Microsoft as you know lost the war a lot of years ago, but with Xamarin and if it is pushed in the right direction (corporations) maybe it can accomplish its finality, using .Net in all the devices and open that closed door to .net developers. Future projects should be IOT projects with massive data using redis as a hub and ms sql server to persist, apart from the typical back-ends with angular or MVC. I love first type of projects but they aren't for the regular .Net developer used to only web, but I am not the typical .Net developer because I have been programming a lot of WinServices, WinForms and back-end, that's the reason and maybe I am not so objective about all things I have said 😂

I would be definitely interested to code some ASP.Net Core in 2017. I also think Xamarin is gaining ground and interest on the mobile since it was acquired by Microsoft. I don't know what they will do with UWP, but right now it look appealing only if you are targeting Windows Phone 10, which is dying... I hope that Microsoft will work on merging the .net Core framework and Mono in 2017 so it can truly compete with Java...

In my opinion there will be no revolution in the ".NET world" next year. I expect the release of Visual Studio 2017 and C# 7.0 to be the most significant events. It seems to me that C# 7.0 is a (very welcomed) evolution of the language but it doesn't bring any killer features as the previous version of the language used to. I'm of the opinion that there will be more happening in the web-development space. I think (and hope) that many organizations will adopt TypeScript or Babel.js. There'll be a steady decline of AngularJS in favor of other frameworks such as

Aurelia or React / Redux. Many developers (myself included) and organizations felt left behind by Google when they released Angular 2 without much support with migrating legacy (angular v1) applications to it. Last but not least, many developers are simply tired with the volatility of the web-space. You can see that at meeting group or by reading blogs. There's a new JavaScript framework every day. Organizations (and developers themselves) struggle to choose the right ones and are not certain if someone is going to maintain and support it in the coming years (see Angular 1 / 2 case). Someone wrote a blog post the other day that a large javascript application with angular js 2, node.js and babel has more files and unnecessary complexity than Linux kernel. I think it will stop and hopefully companies and developers will start to settle down on some industrial standards, however they'd be defined.

Well the next big things are: .NET Core incl. EF, Xamarin and - of course - the blockchain. Domain Specific Languages (DSL) are less known, but their use will be more accepted. If you want to cantct me by phone: +41 79 650 68 46.

It increasingly looks like .net is going to a backend role and standard web technologies are used for UI creation. A big promise for .net is also Xamarin; but I've yet to see significant uptake. So I'd say the main trend is towards the standard .net stack (C#/WCF/Workflow foundation and in some cases WPF) + web technologies (HTML/CSS/javascript/jQuery etc etc). Cloud tech also plays a role, so Azure knowledge would definately be beneficial. Additionally, Win10 is starting being used for .net IoT products. And I'm keeping Xamarin in my "things to watch closely" list.

Hi Jamie. Here some keywords for me as a .net fullstack developer / architect that are interesting now and in the future: azure or cloud in common, nosql, event sourcing and cqrs, typescript, angular2 or react on frontend, .net core / web api on backend, websockets/signalr, nodejs as c# alternative, high scaling environments. Hope that helps a little.

It's a bit complicated to give a try about prediction for technology evolution, I can tell what I'm interested in .net core it's quite a big revolution. Typescript and Angular 2 are almost a standard. I'm spending a lot of time studying Atom (https://atom.io/) for desktop applications.

As for your market research. I personally think .NET will be around for a long time as so many companies out there locked themselves in with that tech (us included), but companies will transition out of "lock down" tech such as .NET as it's fairly closed source (even if Microsoft opened it up recently) and it's quite platform dependent including being owned by Microsoft, meaning you buy your support from them as well and it costs a lot. Lately it's all about open/free tech and "cost savings", and doing things on the web as far as it can be done, before considering "thick clients" (full

---

scale applications that run on the OS). There will always be a need for heavy applications that aren't web-based of course, and I think that's where .NET is still a big player, along with Java/JavaFX. We delivered our JavaFX trading application recently where I wrote most of it, and it's been a success. That said, even JavaFX is in the "danger zone" as it's just too slow on getting changes/fixes implemented and people are starting to look more toward Web there as well as they just want to rapidly build something that any client can run without having to resort to package installations, weird dependencies, security changes, you name it... it's a headache for any security person in any company and always comes with a big price tag. I think in the web world there's a few different "teams" of people who prefer Angular2 vs ReactJS vs [hot trend of the day]. They're all fine, but all of them have pros/cons of course. Web tech is moving so fast now, it's scary, and it's also very bad in a way for it as it creates a need to be "very much on top of the trend" and be able to adapt to quite destructive changes with each version of the web frameworks that come out. Just with Angular2 I feel like I've re-written the current application I'm working on 3 times over the process of new versions of it coming out, just in a few months. In summary, I think any .NET / Java developer that wants to be relevant and wants to work on new products, need to embrace modern web as a friend or they will miss out. It's a surprisingly steep learning curve as Javascript has gone from "a fun scripting language" to "it can do everything, including run servers", and it's sometimes almost too flexible in that regard. But I think once people embrace it, they will quickly realize that heavy frameworks such as .NET become more of a niche, rather than a "lets do everything in .NET". So I think that's where the trend is moving, to go from heavyweight to lightweight and from fat client to web client.

I don't think there is an easy answer to your question. Obviously, in a generic way, I could tell you that the future is clearly driving us towards BigData and Machine Learning, being .NET or not. Also, at least in my field, everything looks like Office 365 connected and, therefore, Javascript and REST are the key topics to look at. That being said, personally I would never give any value to the technologies to touch connected to a job position. Of course trends are always important and you will look with different eyes at a position demanding Delphi or demanding React.js but... what about if the project is extremely interesting and they just have a huge code base in Delphi? If technology was a key factor, I would have left SharePoint a while ago since it was always far away the latest trends in development. Opposite to that, I made the best of SharePoint to enjoy every position I have help over the last years.

Well i will be brief on this subject . I'm not a .net developer but a SharePoint developer so I only use .net in the context of building sharepoint components. And is this particular case, and as you probably already know, Microsoft is trying to move all the custom dev on the platform from server side to be executed on the client side (browser) which means in the near future all developers (specially sharepoint ones) need to have knowledge with javascript frameworks like react, angular, knockout etc... and bye bye .net. But it still depends on the context of the application the developer is trying to build. He can still work with server side code (.net) but maybe in this case a little bit of knowledge on Azure would be necessary to host the applications on the cloud.

Jamie Rogers – Swiss .Net Software Recruiter
Jamie.rogers@darwinrecruitment.com / +41 41 506 2919

I think the future trend would be in terms of technology around IoT and cloud in general. You know probably better than me what kind of profiles the companies are expecting today but there are many of them which are nowadays orienting there development stack to those 2, including the back-end services such as micro-services architecture, service fabric application, notification service hub and all the other services the cloud is offering. Those are of course combined with mobile development technologies: like xamarin or other mobile front-end framework (react, meteor,...). For me, the .Net technology won't be anymore more focused on pure development but instead on services (higher level). Those are as well what I start to do and want to do in the coming years.

In my opinion, in general, the future will be the addressed by two main areas: automation (office, Object, Industry 4.0 etc ...) and Big Data (Storage, Services, Analysis, etc. ...). In this scenario, as a .NET developer and software engineer, I cannot concentrate only on a single platform (eg, Windows or Mac), but we need to learn and develop applications (Web, desktop, mobile) that can run on different platforms, an example: if I need monitoring a production machine (CNC) or a production line by sensors, I can use C # with .NET or Mono Framework Core to develop a simple IoT application that run on Raspberry (low cost hardware), send read data from Web Server (eg. Asp.Net Web API) and display information on different devices (Android, Windows) so management can use them to quick decision about the business. I believe the future of .NET will be based on multi-platform tools and framework as Xamarin (Android, iOS, Windows Phone), VS code (Linux, Mac, Windows), .NET core (and the full version of the .NET framework). From a software architecture perspective, I think that in the future, SOA (Service Oriented Architeture + RESTful applications), Event Sourcing Pattern and CQRS Pattern (using Service Bus) will be more used. In the end, agile methodologies and tools that support them (such as Visual Studio Online).

Thanks for reaching out. Here are my thoughts about the future of .NET : As for now, I'd say it's not as popular as JAVA, but the trend might change, and here is why : 1) The platform is rock-solid. The first .NET framework has been released back in 2001, and it has continually improved over the past 15 years. 2) Seamless integration with Microsoft Azure, which is a competitive SaaS provider offering everything needed to host modern cloud-oriented applications. 3) The core Framework is being open-sourced, so it will make possible to run those apps outside of a Microsoft environment, which is a pretty cool thing. Basically, it might open the doors of a new era, and .NET could get more and more interest in the next few years. Hope those information's help.

Jamie Rogers – Swiss .Net Software Recruiter
Jamie.rogers@darwinrecruitment.com / +41 41 506 2919

This is a very interesting subject! I don't have many time at the moment, so I'll just answer you with a few words :

- TypeScript. I think Microsoft will somehow need to better integrate it in the dotnet family, cause ASP.NET won't survive the javascript frameworks like Angular.

- Xamarin. I think this will really take a bigger place in the mobile Developement in the upcoming years.

- Unity. VR is a big challenge, its interest is really growing, and Unity is a really great platform.

Currently I see the trend is going towards analysis for data and predicting trends and to tap in this area Microsoft bought R scripting language and called it Revo scale R (Microsoft R server) which is integrated with awl server 2016. This is where my current interests are. At the moment I am working on analysing big data using Microsoft R server and spark in Hadoop environment. On top of this you would need visualization tool like Microsoft power BI or other web based visualization tools to show these reports for the business users to analyze the data. I also feel going forward lot of reporting tools like SSRS would be also used heavily for generating business critical reports

---

I can only speak for a small subset of .NET technologies, as my current work revolves mostly around C#.NET Desktop Apps with WPF, and some data science I perform in Python. As well, I have not performed any "market" analysis, and can therefore only tell you what I will be focusing on in the next year. My biggest frustration when I work with other developers is typically the lack of knowledge (or correct application) of fundamental concepts in software architecture, which have been around for decades. Developers often reinvent the wheel, and the result is usually poor compared to mature frameworks. In the past years, I have seen more and more great frameworks and tools (Telerik, SciChart, many Nuget Packages, JetBrain tools, Postsharp, NCrunch, etc) that make it easier for developers to quickly deploy the right patterns. I am confident that these tools and frameworks will continue to prosper in 2017, making existing code easier to maintain. Lightweight IDEs will certainly been an import trend. Visual Studio 2015 was a great step, much more efficient then the previous versions, and I hope that VS 2017, with its modularity concept, will be even faster at executing my most common tasks. In any case, I am eager to explore JetBrains Rider IDE for .NET, that promises to be lightweight. Especially, it allows to quickly write Python plugins. I am very excited to start automating my workflows to my hearts desire using Rider's plugin system. The developments in C# 6 and 7 are very nice. The features are not fundamentally new, as I use most of them already for years in Python. I highly appreciate that to use the powerful/short syntax of Python in a strongly typed language such as C#. Recently, I have gotten more into functional design patterns (Monads, etc), and their close relation to proof automation systems such as Coq (Coq can export to Haskell). Functional programming languages are equivalent to mathematics (see Curry-Howard correspondence), which makes them very powerful as the code can be proven to be correct and thus exclude any bugs (any developers dream). Therefore, I eagerly started to learn F#, which I could immediately integrate with existing C# projects. Seeing the developments in C# 7, in particular pattern matching, I believe that the coming year will see more

steps towards a more functional C# language. Maybe in the long term C# and F# will merge? Who knows. Any case, F# will continue to prosper to write code that can at least partially be proven to be mathematically correct (using a GUI written in C#). This reduces costs massively for developments in finance, medical applications, engineering, and any sensitive application were the correctness is crucial. https://fsharp.tv/gazettes/f-the-most-highly-paid-tech-worldwide-in-2016/ Last but not least, as I am moving towards integrating some simple AI features into existing projects, I believe that the use of libraries such as Math.Net Numerics (for leveraging MKL), CUDA.NET, AForge or Alea will keep growing, as the performance of pure .NET is insufficient for such applications.

It looks like .Net will become more mulit-platform environment with development of .Net Core framework. There is a lot going on in ASP Core ^ Entity Framowrk Core frameworks. Also with VS Code a multi platform IDE is available and with contribution for open source community is getting more powerful each month. Another trend I hope :) will happen is functional programming getting more popular in .net community. On one hand some features from functional languages are included in new versions of C# (C# 7 will support tuples and pattern matching), but F# still does not have the some popularity as Scala in Java world. So maybe next year :)

In my opinion Azure and Automated UI testing is going to be the most interesting topic of incoming 2017, or at least I hope so as I have some strong knowledge in Automated UI testing :)

Xamarin

- VR - AI - Linux (.net core) - Web assembly – Security

Working in the medical devices domain, I would say change happens here slower than in other areas, but I can see in the entire industry a higher focus on security and open source technologies. I think there is also a trend to "free-up the DB", moving business logic from databases to higher levels, like C# applications. In this case, people with good skills in both C# and databases would be needed. It's relatively easy to find people who know one of the two, but really difficult to find people who know both. Lately I was contacted by a few recruiters looking for embedded software developers, even though my embedded experience was very old in my profile, so I suspect the need for such developers is quite high

---

I'm working in a very specific part of software as I'm currently dedicated to industrial field. We tend to be in a field where we are far being in term of technology, however we see a recent shift with a lot of focus on concepts like "industrial IOT", "Industry 4.0". Even some of our clients start talking about cloud hosting and big data with analysis capability or even predictive capabilities. From my stand point the next focuses I'll have will be like this:

- Quality : Visual studio team services online, continuous integration, Sonar Qube, Unit tests, continuous deployment to cloud hosted dev environment.

- Productivity : Tweak of scrum methodology for small team with multiple projects, Resharper (visual studio plugin)

-Technology : Azure, shared database server strategies, MS SQL 2016 with R service (R is a data science oriented language supported by MS SQL 16), .NET Core (for linux hosting and embedded applications), .NET MVC with angular 2. I may have a look into containers also, but it ma not prove relevant for my industry.

---

the trend is going to the cloud, so cloud platforms like Amazon or AWS are the key. .NET core is also catching up. C# remains the leading language, VB has lost its popularity because of functional languages like F#. Javascript on the server side with Node and cognitive services/machine learning are also trending. Personally I am interested in back end development or everything you can scale to millions of users.

In our company we have stared a couple of month ago to develop cross platform apps with the Xamarin framework! I think developing Apps on all mobile platforms with Microsoft languages and tools have lots of benefits ! We also have developed UWP line of business Apps and it's really great ! Unfortunately Windows Apps for end consumers are not yet very popular ! But thanks to the Surface devices (maybe soon Surface phones), Microsoft dev tools and the Microsoft open source strategy it could change a day. Regarding Web applications there is currently lots of SPA frameworks in the market ! But as Angular2 it still not largely used I would go on with ASP.NET Core MVC framework. In summary, for me the futur Microsoft dev technology/tools trend could be: - UWP + Xamarin - ASP.NET Core MVC / Web API - Microsoft Azure - Application Insights / HockeyApp (APM) - Microsoft Team Services I hope my contribution will help for your research! By the way, even if I'm pretty happy with my current job, I'm always open for new opportunities. :) As I'm pretty new on LinkedIn, my profile is not yet filled in details ! But if a day you're looking for Microsoft Software Managers (ideally in the Bern region) I could be interested! :)

I am a big fan of functional and distributed programming. Therefore topics that are interesting to me are F#, reactive systems, agent-based programming, and containerization.

Well, maybe I will not answer exactly to your question but in my daily job, typically, I am bound to existing technologies and cannot afford to be completely up-to-date with the new, bleeding edge trends. However, for the year 2017, the Azure Service Fabric and the Windows IoT Core OS will be in my focus mainly, these will be the new .NET related environments that are quite new and I will be looking into. They are two completely different worlds. The reason why I am

interested in both is that I got a new job and will switch in March so I will be switching from cloud based software development to embedded or hardware close development.

All .net Projects will be developed with Microsoft azure platform because it s an hight Scalable, fast and cheap cost cloud platform. The future of mobile application is Xamarin platform : you could develop ios and Android application in .net languages , it Must be coupled with Microsoft azure cloud api. The future of the web is Asp.net mvc 5, asp.net web api and asp.net mvc core , coupled with angularjs 2 , coupled with Microsoft azure cloud api. The futur of entreprises infrastructures and organisation tools is sharepoint 2016 and office 365 . It s cutomizable with .net développement.

The main disadvantage of .NET was that in comparison to Java ,it was slow and also all tools needed for development were expensive and also not supported on all servers. With the new release of .NET Core last November all benchmarks show that is superfast like Java .Also Microsoft made .NET open source and added support for mac and linux. In addition everyone can download visual studio work or practice by himself for free. All these facts i think going to boost .NET and c# . In 2017 .NET core will become a bit mature and companies will start migrating big and core projects from java to .NET . Also as i mentioned before .NET Core runs and is supported by microsoft on linux servers , so migrating java projects or c++ will be easier cause of less costs of web servers(as they don't need to change anything) On the front-end Angular 2 i think will win vs React for bigger applications. So for me i would like to get involved more with Angular 2 and get experience building microservices and back-end logic with .NET Core Api and MVC. Finally i like a lot the financial sector so i would like to keep working on this one . Thanks a lot for your contact and sorry for delay i was on vacation.

Recently I've seen .NET roles that require knowledge of Azure and IoT so this seems to be a trend in the past while. MS have an IoT offering but I'm not sure how advanced it is: https://www.microsoft.com/en-us/cloud-platform/internet-of-things-azure-iot-suite Also knowledge of Xamarin and mobile development seems to be an asset although I rarely see roles for pure Xamarin development. Not much but I hope it helps :)
