Interview Transcript:

Steven Spiller. Senior Software Engineer Xbox Microsoft.

The Fighting Mongooses had the opportunity to interview Steven Spiller, a senior software engineer for Microsoft, specifically for the Xbox platform. Even though the upcoming release of the new Xbox was quickly approaching, and Steven being extremely busy, we greatly appreciate the time Steve took to meet with us and we are extremely grateful for the knowledge and passion that he shared with us.

1. Please tell us about your IT work. What exactly do you do?

My name is Steven Spiller. You can just call me Steve. I work as a senior software engineer for Xbox/ Microsoft. I work in a little team is called, well not so little anymore, but it's called ATG, the Advanced Technology Group. Our role in the organization is to help game developers make the best use of the platform of Xbox. Specifically, I work in networking and multiplayer systems. If titles are having trouble or they need advice, or they are behind this severely behind and they can't get their stuff to work on our platform. Our team is the one that helps them out.
It's not as interesting on the networking multiplayer side, but the on the graphics side. Those are the guys that really help titles tune on our platform so they'll be able to take traces of the games running, evaluate them, and offer back tips that can help them run faster so they'll get games to run at least in the previous generation all the way at 1080p instead of only 900p. Things like that. So really our goal is to make all the titles that run on Xbox the best they can be on Xbox. I think that was our motto at one time.

2. Please tell us about the industry you work in.

Being at Microsoft, I've done a lot of things in a lot of different areas, but most recently we can just stay with video games industry, so the Xbox stuff is what I've been working on since 2013. So the last seven years. Microsoft provides platforms and services for game developers. It is a fairly large industry. You've got billions of dollars and millions of millions of gamers across the across the world and that's what we're trying to do. We're trying to make things better and easier for gamers across the entire world. Like a panacea of the world. Good way of saying that, but we want to make everybody's life easier.
Is there any?

3. What other kinds of work do you have to do?

In my role, I'll do things like answer simple questions on a developer forum that is specific for Xbox Partners. If they have issues, that's one of the first places they'll go, so I'll do a little time answering those, and those can be from really simple questions like *"Is this supposed to work like this too?" "Hey, we're getting a crash in this very strange situation." "Can you help us figure that out?"* I will investigate issues that have been reported by external developers. I work directly with the platform team, the people that actually wrote all the code. I work with them to get things figured out and resolve bugs, find gaps in our solutions. I've also developed samples and demonstrate best practices for using the platform and our services. For example, if you want to know how to do multiplayer matchmaking through Xbox Live, we have specific examples and lots of documentation that I put together to help with that. Then of course if people still have issues and they can come straight to us. Because we do work straight not just from forums. We will also engage via email, a lot more on teams lately, over the phone and video conferencing. We also go on site with developers to help them get issues resolved or to help them get their game working on Xbox. I've done that several times. Which is it's kind of fun. It's very busy though. We also deliver content for (what used to be) in-person conferences. We had to cancel the last few and turn them into digital only. I don't know what the future holds as far as those go. We have several annual conferences. Directed around Xbox developers in gaming and we provide educational training and. We've done discussion sessions where people actually come out, listen to us and talk, which is kind of fun.

4. Who are all the different people you interact with in your work? Please tell us about them.

Yes, I do a lot of different things as you could tell right? And those have different people I work with, so when we're developing new samples, we want them to have a consistent look and feel. We actually have designers, graphic designers that help us design the UI flows and figure out what the scenarios that we want to show. Program managers, we actually call them the *butt on the line*. These are the people that own features and are responsible for making sure that their status is tracked correctly and that they're (the team) doing the right thing. All that work other software developers obviously, and then we have another special type of person in the Xbox Organisation that we call *developer account managers* (DAM). These are the people that actually own the relationship with the different publishers. We have one person that owns Electronic Arts. When Electronic Arts has any sort of problems, they need to do something, they want to ask about future plans or anything related to this? They talk to their DAM and then their DAM will be the one that often contacts us with specific issues or other problems they might be having, or if they just want advice on things, or *runs the gamut*. Those roles are internal people. Then of course there is externals, which I think falls more under your clients.

5. Please tell us about your interactions with other IT professionals.

There's five people in my immediate team or my peers, and we all basically work on the same thing. There's a bunch of teams like mine that take on the different areas of Xbox. So, you add audio, graphics, ours is networking, systems and CPU and things like that. Of course, you have the whole platform side of things where the people actually write the OS. They write the special Xbox layers on top of things and we interact with them quite often.

6. What about your interactions with clients or investors?

Most of the work I do is direct with clients. Either through a forum, through email, through teams and up to on-site visits. The DAMs will set up what we call an engagement. If there's a real problem and we need to set aside time. If It's going to take more than a day or so to look at this. They'll set up what's called an engagement, and then, depending on the level of that and then other factors like, where are they located physically? Are there travel restrictions? Things like that. If there's a developer that's just down the street (there are some that are like that), we will just go there because it's easier. Other times, we can do things over the phone and over email. That actually works out fairly well. It's nice because it has a paper trail so we can go back and reference this stuff again. So anytime we go, and we solve problems and we figure something out, we have this big knowledge base (It's really a OneNote). We store all this information and so we can go back and search for it again.

7. What aspects of your work do you spend the most time on? Please tell us about these.

That is almost like a sine wave, right? We will go from where we are doing nothing but working with customers and right now is one of those times. We have the new launch of the console coming up in a month. We have a ton of titles that are all trying to certify, they're trying to get their stuff updated to the new platform. There's a lot of people all working in with new things all at the same time, so we're quite busy answering questions and helping people figure out how things are supposed to work these days.

Then on the other side of that, you'll have these big lulls where people have become familiar with the platforms and the systems. They don't have as many issues anymore and there's just not a lot of work going on. It is somewhat cyclical as to when software or games are released. During those times we get to work on what we think of is more forward-looking items. So, we will have incubation projects for new tools, new ideas, new services, as well as being able to do work where we're building up new samples. New middleware projects etc. One thing that we don't offer a lot of is middleware. By that I mean things like the Unity and Unreal engines. A lot of developers go to those because it it's *pre-canned* and has a lot of stuff. So, we work with Epic and Unity to have Xbox platform support built in for everybody and sometimes we will do things like that. In fact, just recently one of my colleagues and I finished adding a new platform piece that's called play fat party on Xbox. It is a pre-canned voice and networking solution and now that's something that you can choose to use. If you use Unreal, you can say *"I want to use that piece."* You check the box and basically it just automatically starts using it because it's built into their engine. We do some of that work too and that's a lot of fun, right? That's when you're spending more time coding than troubleshooting. Other times it's more troubleshooting than coding . It's actually one of the things I like. It comes up later. I think a lot of us would like to spend more time coding than dealing with some of the clients.

Adam: "I think a lot of us would like to spend more time coding."

Yeah, it's really hard. I've worked in a lot of different types of groups around Microsoft and no matter what, there is never a job where you're just spending all your time writing code. There is just too many other things that have to be done as part of the process of shipping software that you can't get away from.

8. Which aspects of your work do you find most challenging?

What I find personally often most challenging is trying to understand each different developer's actual needs. A lot of times they'll come and say *"I'm trying to do this and it's not working right. Can you tell me what the problem is?"* and well, yes, I could tell you specifically why that isn't working, but this seems like it's a symptom of a larger issue, right? You may be doing something at a more meta level. That's incorrect as far as the pattern that you should follow. We really need to spend the time, or understand what the goal is, what are they trying to achieve and what limitations they have so that we can then find the right way to solve that problem for them or to help them solve the problem. Then of course sometimes things work out really well but being a global product is challenging just with the translations. Some things like when I work with a Japanese developer. We have an intermediary that translates back and forth for us. So we type a response, they'll go in and translate it. The other person will then type their response. That will get translated back and it's kind of interesting, but you can lose a lot easily. It does make it take longer, and sometimes it just doesn't come out in translation exactly. You do lose what it is. I had one, I thought I saw that they were asking for, but then they came back and asked, *"It came back a slightly different way?"* I guess they meant this other thing and so that can be a challenge.

Secondly, with the complexity of all the systems and how they have to work together. You wouldn't believe how many things are involved. From when you when you sit on your Xbox sending an invite to one of your friends to play a game and for them to receive it and join and be able to play. There are many steps and different systems that all the things flow through. Sometimes it is really hard to find a root cause. You'll say*, "Oh, I have this problem*." It's showing up. Because of this, the user won't sign in correctly, but it's not actually a sign-in problem, but it actually turns out that that user is part of this title group that has these special properties set that only works in this other situation. It can take 5 to 10 people sometimes on an email thread just to figure out what's supposed to happen, what is happening and how we can fix it. And to be fair a lot of times fixing it means telling the title developer that they were doing something wrong, so it's not always our fault, but I usually take that angle first that I take the assumption that it's something that our stuff is doing wrong and then let people discover that it's not necessarily the case.

Adam: I think the end users will just blame you guys anyway, wouldn't they?

Yeah sometimes it's really hard to read Twitter comments because the lack of actual understanding. It can be frustrating when you read some of the things they say.

Finally, can you share an example of the work you do that best captures the essence of the IT industry?

When I thought about it, I came up with the definition that it's basically utilizing technology to accomplish tasks or to get work done. If I go by that, I would say that when we were developing our samples. Samples do range from like one code file that just does something really simple to a full game. We have a full game that is an example/ sample for a bunch of different things. It's is a lot of fun to work on. By doing that and having those samples out there, when we developed them, we developed so that they would pass our own certifications and that they follow all our own best practices so that it's a working example of the right way to do things. The nice thing about doing that is that a lot of developers will just take that code directly and they'll make it fit in their engine so we know when they're taking code that we've developed and tested and we know will pass certification and they're just using it. It's kind of win-win. I think that's actually one of those synergistic things where you gain more out of the end results, with the effort that was put into it initially. One of my favourite go-to is (when a client says) *"I'm trying to use chat and my chat doesn't work in this situation well."* I *respond, "Can you load the sample and see if the sample has the same problem?"* If the samples have the same problem, then we're probably looking at something had changed or broken the platform and we can investigate that way if it works in the sample then we can narrow it down to something in the way you're calling the code. So yes it does help a lot.

Daniel: Do you work from home at all and if so, do you prefer it to working in the office? Or if not, would you like to work from home at least some days?

Yes, so current situations aside, I had been working about 50-50 between home and office. I found that to be a really good balance for me because I do enjoy the comfort and convenience of being at home. I can just get up and I have everything around me. It's simple, but there's times when I'm working with people directly, or we're trying to troubleshoot issues, or I need more equipment than I have. At work, I have a stack of Xbox Dev kits that can utilized in many different ways. To do things at home. I only have two (Dev kits) and I can't do as many things and it's more difficult and more time consuming to do certain things. I do like to work from home, and when the office opens backup. I'll stick to our 50-50 style.

Adam: What aspect of your work do you find most enjoyable and why?

The things I enjoy the most is when I provide a solution or an answer to somebody and they come back with *"Oh thank you this works. I've been working this forever and I could beat my head over and you solve my problem. Thank you so much!"* When I help people solve problems, that's just that's my favourite thing. That's what I enjoy and end up getting the most enjoyment out of. There's personal satisfaction with doing your own projects, and these things like that, but when somebody physically tells you *"You solved my problem and helped me!"* I really like that. I also really enjoy the fact that my job is so eclectic at times. I'm not in the same rut of doing the same things. Each task, each problem is a new challenge. So it's it keeps it interesting.

Maddie: What advice would you give to someone who is just beginning to study or work in the IT industry?

It goes a long way for me is to find areas that you're really interested in and then saturate yourself with that. Really, the biggest thing you can do to help yourself is to do it hands on. If you think that you know configuring and running server farms is going to be fun, go out to Azure or someplace where you can get free virtual machine space for tinkering and do it. Do it over and over again. Try different things, break it because you will break it. Figure out what you did wrong. Basically. Now give yourself a sandbox to play in because that immersion is what, at least for me. Makes everything else come together more easily.

Channon: Do you find it hard keeping skills/learnings up to date?

That's a yes or no for me since we as Xbox, Microsoft will provide new services and new technologies all the time. It's just kind of part of my job to learn and know what is being developed and we're supporting them and helping other people figure out how we use them. I don't have a lot of time to invest in skills outside of my job right now. So, if I decided I wanted to learn something that's unrelated to my job, I would really have to spend most of my private time to do it. If it's something you really love then great, but it's harder to invest in something that you don't know will pay off when it's taking up that much of your time in my situation. It can be hard, but it if you put yourself in the right position, it can just come along with the territory.

Maddie: Do you find that's a result of the new release date coming up and Covid, or is that always the way it's been with your job over the past seven years?

More so lately, obviously. The way I do things, like when I find something that I'm interested in and I want to do it, I'm just going to do it right now. That's how I choose to spend my free time.

If that happens to be learning new technology, then that works out. A lot of times though, I tend to want to go back and do things that I would find more relaxing I guess, so doing more retro things, going back and playing on the old Apple the TRS 80, old game console programming. You know something simplistic. It doesn't have to go anywhere. It may or may not help you in the in what you're currently doing, but it does still help you. You keep using your skills fresh. I guess the more you do different things, the easier it is to switch between things.

Jorge: What do you think are the most important skills and attributes for a job like this?

My job specifically, we're looking at having strong troubleshooting skills and good knowledge of relevant technologies. Most games are written in C++. If you don't know C++ very well, you'll probably have a hard time understanding certain things because you're not familiar with what side effects are there or how memory is treated. Equally, if you're not familiar with rest. APIs over the web you could easily make a lot of mistakes. So basically, the dedication to be able to complete tasks. That's one of the hardest things I find between myself and a lot of my programmer friends is we all have a million side projects that never get done. If you can actually sit down and make yourself go through the process of doing everything to completeness, that just makes it makes it easier because you've practiced it and your life will be better anyway, because you won't have things coming back to haunt you as much.

Patience is another good one. Sometimes it's really hard. One of my colleagues was having this issue today where he just couldn't get a problem that somebody reported to reproduce even with their own code. And he's like *"I just don't know what to do anymore. I'm stuck!"* Like, Well, let's just keep going through it and you just kind of keep trucking and eventually things pop up or you figure something out. You can't put it aside or give up. Obviously can't do that too much when you're getting paid for it. Patience just helps overall.

Sam: What new skills do you think will be in demand in the industry in 5 years?

The big future investments these days are around artificial intelligence, machine learning and big data processing. In games, especially machine learning, not just for gameplay, but systems that are used to make games. A game is really about moving and manipulating large chunks of data over and over and over again. If you can have a computer that can use a machine learning type thing that can look and evaluate. It can say *"I seem to be loading these chunks from this specific place on this interval, in this mode. So, I'm going to self-optimize this and pre-fetch things that I think are going to come up."* By simply doing that you've all of a sudden increased your load times for your game and for other things. *A.I* would be more along the lines of if you are having smarter systems such as player interaction systems. When you're developing the game, you know there's a certain way to go about doing things and you try to optimize for what you think it's going to be. The most common scenarios, but you don't really know what those are going to be, and even if you do find the most common scenarios, what if it's only 60% of your users and the other 40% are having a less than perfect experience using something like ML The software itself could make the determination and dynamically pick the best optimizations for you, instead of having to statically choose ones that you hope are the right ones ahead of time.