# TechEd: Developers 20075 November 2007

## Pre-Keynote Session

### S. Somasegar

#### Corporate Vice President, Microsoft Developer Division

##### Outline

I will take this opportunity to give you a summary of some of the things we are going to talk about later this afternoon in the keynote. If you look back to this month two years ago, November 2005, we launched Visual Studio 2005 and .NET Framework 2.0. I will reflect upon what kinds of things we have seen in the market with these products.

##### Successes

###### Overview

First of all, I want to say that we have been super-thrilled with the response, momentum and excitement we have seen for Visual Studio 2005 and .NET Framework 2.0. If you look around the world, we have more than a million professional developers building applications and experiences they want to develop to their customers in a highly effective and successful way using Visual Studio 2005 and .NET Framework 2.0. No matter how you think about it, the licences, how people use it or what kinds of applications they have built, we feel good about the kinds of successes people are seeing with these products in the marketplace.

###### Visual Studio Express

If you look back to before 2005 and what we were doing with Visual Studio then, with the Visual Studio 2005 release, there was one big thing we did with that product line: we said we were going to broaden the reach of Visual Studio on two ends of the spectrum. At one end, we said we were going to make a version of Visual Studio that we called the Visual Studio Express product, which was targeted at beginner developers – people who wanted to get started with programming, who thought the barriers to entry with Visual Studio were too high and wanted an easy way to get started, to learn and become proficient with programming and development. There was a series of products, which we called Visual Studio Express. The good news is, two years later, we have seen about 17 million downloads of Visual Studio Express, which is a huge number in terms of the developer tools and options. The good news is we have delivered on our mission of being able to deliver on a set of products that was reachable and usable by a broader set of people around the world, who are not necessarily professional developers today, but may want to get there someday or may want to build a set of tools to build applications that could be for fun, to automate some tasks they wanted or to build some experiences they could share with their friends and family. That is one of the reasons we have been thrilled with the response we have seen in terms of our options for the Visual Studio Express product.

###### Team System

At the other end of the spectrum, we also wanted to use Visual Studio 2005 as our first foray into the application lifecycle management tools space. Previously, we had always been focused on saying Visual Studio is a great toolset for you, as the individual developer, to be able to deliver high levels of productivity. With Visual Studio 2005 we said we wanted to take that focus on productivity for the individual developer and expand it for a broader set of people. More and more, in today’s world, it is important to have developers, but we are also seeing a situation in which there are more people other than just developers. They could be project managers, business analysts, database professionals, testers, architects or designers. All of these people have to come together to build next-generation applications, so we are providing a set of tools that work together in an end-to-end integrated way, which enable seamless workflow and data exchange between these different roles. We thought that was a good way to expand the focus from an individual developer to the development team at large to enable high levels of productivity for the team. That was our first entry in producing Team System and Team Foundation Server. Two years since putting out these products, about quarter of our developer base, or 25% of professional developers from around the world, are using the Visual Studio Team System tools and seeing the benefit of team collaboration and the team environment, where they can work and be more effective working hand in hand with other sets of people.

###### A Self-sufficient Community

Vision

Finally, as much as I can get excited about the product and technologies we have built, and feel good about that, a big part of our customer success is a result of having a vibrant and self-sufficient community. Here is my dream: if I am a developer in the world taking a bet on a Microsoft set of technologies, which could be tools or a platform, and I have a question, a query, need some information to help me write my application better or am running into a brick wall and need some help, I should be able to post a query and know that, within a reasonable amount of time – within a few hours, a day or maybe within two days – somebody out there in the world is going to help me get the information I am looking for, help me solve my problem or help me make progress. I know that that community exists for this set of technologies, platform and tools. We are very committed at Microsoft to working with the community not only to make sure that there is a great community but, more importantly, that that community is self-sufficient.

Response rate

If you go back a year or 18 months, and look at our forum, which is a place to come to in order to ask questions for them hopefully to be answered, we were running at a 30‑40% response rate, which means there is a 30‑40% chance you will get the information you are looking for or, correspondingly, there is a 60‑70% that you will not get that information. That is not a good place to be. We invested on the Microsoft side and partnered with the community to ask how we can work together to enable us to have a much better response rate. The result is that, as of today, we have over 80% of all the questions coming in through the forum being responded to, mostly by the community and some by Microsoft. We want to jump in and help, so have a scale issue. We wanted to work with the community and have seen some great progress and great strides in terms of making the community be as self-sufficient as possible. We will continue our part in making that happen.

###### Progress

If I look across the board at options for the products, what customers are doing with them and the kinds of successes they are seeing, and the fact that we have a good community around these products, I feel good about the progress we have made over the last couple of years.

##### Today’s Announcements

###### Release of Visual Studio 2008 and .NET Framework 3.5

We have been working on Visual Studio 2008 and .NET Framework 3.5 for a couple of years now, ever since we shipped Visual Studio 2005 and .NET Framework 2.0. The good news is, as we speak here, the product teams back in Redmond and the rest of the world – wherever we have people working on Visual Studio and .NET Framework – are putting the finishing touches to these products. We are on track to finishing them and shipping these products later this month. We are committed to making these products available to our Microsoft Developer Network (MSDN) subscribers initially, and then to the retail customers before the end of this month. We are very excited about this. We think these are some fantastic products, primarily based on the feedback we have received from our beta customers and people who have played around with our customer preview programmes (CPPs). Some of our earlier customers have taken a bet on these products and built applications that have already been deployed in their production environment. We feel good about what we are delivering and are on track to getting it done later this month. That is one big thing we are going to talk about later today.

###### Partner Ecosystem

Evolution

If you go back in history and think about how Microsoft has evolved over the years, we have always been a platform company in our hearts. We knew right from day one that our success and our customer success was going to be gated to a large extent by the strong partner ecosystem we have. If you look at Visual Studio today, we have over 200 partners who are delivering a little over 2,000 products in the marketplace today, which are add-ins, plug-ins and things that run on top of Visual Studio to provide a complementary set of functionality, which make Visual Studio a great tools platform and set of offerings for our customers. We have been getting requests from our partners for us to do more, so they can be even more effective. There are two specific things we are doing today, which make it easier for people to partner with us and build on top of Visual Studio.

Licence de-restriction

We used to have a licence restriction, whereby we told our partners that, if they wanted to use the Visual Studio ID and tools, and build applications, they should target those applications to run on the Microsoft platform. If they were not interested, they could not use our tools. We had that restriction. As of now, we are going to remove that restriction and tell our partners, ‘We understand you have needs, at least in some scenarios, to build applications that target not just the Microsoft platform but additional platforms as well. We will enable you to do that with Visual Studio, because this toolset is widely used and is very popular, and you have institutional knowledge and expertise of this tool. Rather than making you jump through unnatural hoops, use the same toolset to build applications to target any platform of your choice.’ That is one huge step forward we are taking.

Source code access

The second thing we are doing is giving people access to the Visual Studio source code. If you are a partner who has signed a Visual Studio Industry Partner (VSIP) agreement with us and are a premier VSIP partner, we let you have access to Visual Studio, so you can take a look at how the ID is built and what goes behind the scenes, so you can do a better job of designing your plug‑in to run on Visual Studio. More importantly, it makes it easier for you to debug your code and know that it works very well on top of Visual Studio. Those are two specific things we are doing to target our partner ecosystem to ensure our partners are even more successful with our tools and platform.

More technologies and content available

Finally, we are also making available a variety of technologies and some prescriptive content, mostly in a preview or beta mode, to enable developers to do interesting things. I will tell you about a couple of those things here.

###### Software plus Services

Outlook add-in

The world is moving towards software-plus-services. Everybody is thinking about how to build an application that takes advantage of local computing power and extend it to have a board-based offering to make the best of both worlds. We felt that it was important for us to be able to provide a set of blueprints that has a framework, some source code, some guidance and some tools that work within the Visual Studio context and enables you, as a developer, to be able to quickly write a software-plus-services application. Over a period of time, we want to deliver a number of these blueprints targeted at different scenarios and application types. Today, we are making available the first of these software-plus-services blueprints, which is a way to enable you to build an Outlook 2007 add-in easily, which lets you both expose data as well as interact with third-party services.

Example

For example, I have a lot of friends and family and want to remember their birthdays, but cannot because I am busy or absentminded; who knows what the reasons are. Would it not be nice if they can enter my contact information along with their important dates, and put it in Outlook to let Outlook automatically either remind me that someone's birthday is coming up tomorrow or, better still, connect to a Hallmark web service or some other web service to choose an appropriate birthday, wedding or anniversary card – whatever it is – and be able to automatically mail it to that person on the appropriate day? That would be a fantastic value-add for me personally. If we can do that easily using that software-plus-services notion added to Outlook that would add value to me today. Those are the kinds of things I want developers to be able to do using this software-plus-services blueprint: to build add-ins for personal needs or business needs, whatever they are, on top of Outlook. That is something we are delivering today.

###### Microsoft Sync Framework

The second thing we are delivering today is a community technology preview, which we call Microsoft Sync Framework. If you look at how the world is evolving, we have a plethora of digital devices coming out. Pretty much every one of us here has half a dozen or more devices we use on a day-to-day basis. It could be our cell phone, PC, laptop or who knows what other gadget. Sometimes I use a particular device for a particular application. Let us say I am working on my home finance spreadsheet. I am on the road now and have my laptop so am working on that. Tomorrow, when I go back to my home, I am going to use the same application, working on the same data, on my home desktop. The day after tomorrow, if I am talking to my broker from my office, I will be using my office desktop, but I do not have the energy or wherewithal to know which application I am using on which data. Which is the latest version? That is a super-hard task for me today.

It would be nice if the application I am using knows who I am, what I am trying to do and the sets of device I am working on, and automatically centralises the data so that I do not need to worry about whether I have the latest version or if this is an older version. Being able to provide a set of frameworks, which enable you, as a developer, to sync-enable your application in a seamless, straightforward way, irrespective of the data type, data flow or kinds of protocols you use, is a huge step forward in terms of delivering on our data vision. That is precisely what we are doing with this Microsoft Sync Framework. This is a community technology, and an early look for developers, but our hope is that we can give developers something they can look at, play with and look at what it means to sync-enable their applications and give us feedback, so we can do the right things in terms of delivering this Sync Framework.

##### Beginner Developers

###### Overview

I was talking earlier about how we want to broaden the field of Visual Studio to a broader set of people. With Visual Studio Express, we wanted to go after people who are beginner developers. We want to take that one step further. Particularly in today’s online world, there are hundreds of millions of people who spend a lot of time online, and want to either customise or create experiences they want to share with their friends and family, but do not know programming or want to be programming. They just want a set of visual tools that make it easier for them to create those experiences.

###### Popfly

Take my 11‑year‑old daughter. She’s on Facebook. She doesn’t know programming and is not interested in programming, as much as I want her to get started on VB or C#. She says, ‘No, dad, that’s not what I want.’ She wants to customise her web page on Facebook. She wants to create some funny gadgets that she can share with her friends. Popfly is a fantastic tool that enables you to create mash-ups to create gadgets to create web pages with some simple applications or customised applications, using a visual set of tools, without having to write code. This is something we put out last year and, more recently last month, we shipped a beta version of Popfly that means anybody in the world can now use it as a beta product. Since we made it publicly available on 17 October, we have had over 100,000 customers playing around with Popfly, giving us feedback and creating some interesting experiences.

###### Popfly Explorer

Today we are announcing something called Popfly Explorer. Think about this is an add‑on to Visual Studio that lets you do two things. The first thing it does, if you use Popfly to create a Silverlight gadget, it lets you easily drag and drop the gadget and apply it to your web page. If you want to create a cool web page, that task becomes a lot easier using Popfly Explorer. The second thing Popfly Explorer does is let you publish your web page and share it with your friends and family. It is a great tool to provide a set of integrations between Visual Studio and Popfly, which enables you to do easy things, particularly in a context where you do not want to write a lot of code or be programming. This collection of things we are making available today is going to take the developer experience, whether it is a professional developer, hobbyist or enthusiast, and make them able to do more fun things, cool things and great things with their applications. These are the sorts of things we are announcing today.

##### Technology Roadmap

###### Rosario

As I mentioned before, we are putting the finishing touches to Visual Studio 2008 and .NET Framework 3.5, which we will deliver later this month. If I look ahead to the kinds of things we are working on over the next 12‑24 months, and think about the innovations underway, I am even more excited about what we can do for our developer community. As we start winding down on Visual Studio 2008, we have to start thinking about the next version of Visual Studio, which we have codenamed Rosario. We have already delivered a community technology preview of Rosario a couple of months ago. Over the next few months, you will see more technology previews of the theme coming over.

###### Expression Studio 2

We are also working on the next version of Expression Studio, which is a set of tools targeted at professional designers. Expression Studio 2 is a set of tools that will help you target the Silverlight runtime and be able to build Silverlight applications.

###### Silverlight

We shipped Silverlight 1.0 last month of September. We are working on the next version of Silverlight, which we will ship next year. In that version, we are going to bring the power of .NET programming to the front end so that, if you are used to the .NET programming model and a particular programming language, you can reuse your skills and expertise and bring that to bear to this Silverlight runtime to deliver rich internet applications.

##### Conclusions

Between all these technologies, our work on .NET Framework 4.0 and all the Silverlight products, I feel like we have a comprehensive set of offerings coming out in the next 12‑24 months, which are targeted at people who are building applications. No matter what your skill set is and no matter what platform you are targeting, we have a consistent programming model and set of tools to help you do your job and be successful. That is how I think about the roadmap for the next couple of years.

## Questions and Answers

Unknown Speaker

How is the blueprint different from a code sample?

S Somasegar

Think about it as fancy terminology that we use. That is the first thing I would say. It contains three things, as I mentioned before, and a code sample could contain all three, but not necessarily. The three things that we are talking about specifically that every blueprint that we want to put out is going to have are: some frameworks that help you build applications, and we also want to make the [inaudible] available for the framework; a set of tools that help you create those applications, and these tools will be plugins or addins; and a set of guidance that helps you understand what is the best way to build an application. You can call it a super-duper sample or you can fall into the marketing trap and say it is a blueprint, but it is really a set of guidance, some tools and some frameworks.

Simon Bisson, Freelance, UK

Looking at the Sync Framework, how do you compare it with existing sync frameworks from other vendors, such as Google Gears and Adobe AIR?

S Somasegar

That is a good question and I am going to redirect it to Francois.

Francois Ajenstat, Group Product Manager, Microsoft

Our approach to the Sync Framework was to enable two core scenarios. The first was the offline scenario. If you think of all of us being occasionally connected workers, we work offline, we connect to a server and we have to synchronise those scenarios. The second key scenario is around collaboration: multiple different applications that need to share data with one another in order to ensure that that data is synchronised and manage all those complex connections. When we compare Sync Framework to other offerings in the market, our approach here is to go broad to ensure that any data source that you have, any device, any application that you need synchronisation capabilities on we can support. It is very easy to use. We have a meta-data-driven approach that enables you to define the data that you need to synchronise and use that so that you can go in through any protocols. Hence whether you want to do peer-to-peer or other such protocols, we do not have that limitation. Therefore, the concept, and why we are calling it a framework, is so that you can go broad to any data source, any device, any protocol, anywhere.

Silviu Cojocaru, Net Report PC Magazine, Romania

Could you tell me if Visual Studio 2008 Express Edition will be launched this month and also say a few words about SQL Server 2008 Normal?

S Somasegar

There are two parts to the question: will we be launching Visual Studio 2008 Express later this month and what does the roadmap look like for SQL Server 2008? Visual Studio 2008 Express will launch at the same time as the rest of the Visual Studio family of products. Our plan is to be able to launch all members of the 2008 family at the same time, so you will be able to download and start using Visual Studio 2008 Express Edition in a couple of weeks from now, before the end of this month, whenever we make everything else available.

Turning to SQL Server 2008, over the development lifecycle of the product we have been putting out a series of what we call ‘community technology previews’ (CTP). I am sure Francois can tell us the exact date, but I think we are planning to have a CTP this month and I am sure we will have one at the beginning of the New Year and then we will continue with them. SQL has turned into a model now where every CTP is going to be better than the previous CTP, not just in terms of functionality, but in terms of quality, performance, scalability, and in use in a deployment environment. The SQL team is working with a close set of customers from around the world who are taking a bet on SQL Server 2008 and deploying that database in their environment before we ship. Our hope is that we will be done with SQL Server 2008 sometime next summer. I do not know whether we have said anything more beyond that.

Francois Ajenstat

That is exactly right. We are going to have the next CTP of SQL Server coming out this month with a following CTP before launch and then our target release is Q2 calendar year 2008.

Daniel Clarke, The Register

My question is about Visual Studio Team System. You mentioned you had one million licences in 2005, 25% of which were Team System. I wondered what the breakdown was and out of the 25% how many people are writing code, how many are architecting and how many are testing.

S Somasegar

That is a good question. You want to know what the breakdown is between people who are testing, people who are developing and people who are architecting. One of the realisations we are having as we go along here is that these roles are not as accurately defined as we would expect them or want them to be in the real world and there are many cases where today I am an architect because I am designing something, tomorrow I am writing some code, the third day I am testing some code. Thus, the roles vary, but I would say that the vast majority of the people who are using Visual Studio Team System are using the Visual Studio Team Developer Edition. That is the bulk of what our customers are using. We have a number of people who are using Visual Studio Team Architect Edition, who are spending most of their time architecting systems, there are people who think about both architecture and development, and there are people who are using the test tools. However, the vast majority, probably 70%, of our customers are using the Team Developer Edition.

Daniel Clarke

Do you have any information about how the data exchange between the different roles is improving productivity in development teams or is it too early for that yet?

S Somasegar

I am going to share an internal story because I am among friends here. If you look at what we have been doing with Visual Studio Team System and team formulations, our point is that just as the world is getting excited about it and using it, we are equally excited about using it internally at Microsoft for our own development needs. Take Visual Studio 2008, for example, we have been running internally in our team formulations Visual Studio Team System for the last year and a half or so. Since January 2007 or so, we have been running our development on the Visual Studio 2008 version of Visual Studio Team System and team formulations forward, so literally we are using the product to build itself and to be able to deliver so that by the time we deliver it we know that it is meeting the needs of large development teams. If you look at the team that is working on Visual Studio 2008 and [inaudible] 3.5, that is a team of about 1,000, 2,000 or 3,000 people, so we know that the system is able to withstand that level of load and having hundreds of people working in parallel.

One of the reasons I feel pretty good about where we are in terms of how we build and deliver products is if you go back three, four or five years ago, we had aspirations for when we wanted to deliver Visual Studio 2005. If you remember, unfortunately or fortunately, depending upon how you look at it, it took a little bit longer than we originally thought it was going to take us for us to deliver Visual Studio 2005. One of the things we learned from that is that we need to have better and deeper insight into what is happening in the development project at any given point in time to make sure that the tools work together in an integrated way and that the data flow happens in a seamless way, so that if I am a member of the project I can understand what the status of the project is at any point in time. That was a huge benefit that we derived internally as we went through building Visual Studio 2008. When we started working on this project we had an aspiration to shift this product with a full set of features and the right level of quality before the end of 2007 and now we are down to the wire in terms of 2007, but we are on a path to deliver on that. Part of the reason why I think we have become a little better in terms of being able to deliver products successfully on a predictable schedule is because we have better insight into what is happening at the right time. There have been many instances during the development of Visual Studio 2008 when we have felt in our heads that this particular piece of the product is in trouble or that bit has too many bugs or this bit could use a few more developers or whatever, and because we knew in a timely manner what the hotspots were where we were doing well we were able to reallocate resources and be able to adjust how we configured the project so that we can land the project in a timely manner with the right set of features and the right level of quality. Thus, getting that insight so that data flow happens in a seamless way, so that I can know what the bug backlog is for a developer or how well we are doing against a feature and all those things, was super-valuable for me and the rest of the managers in the division to be able to do course corrections and steer the project along in the right direction.

Mary Branscombe, Freelance

I was wondering if you could say a little about how the Outlook blueprint fits in with the existing variety of tools that Office developers have. [inaudible] Does it build on anything specific?

S Somasegar

We have something called ‘Visual Studio Tools for Office’ that you can use to create Outlook add-ins and the like. First of all though, some good news that I forgot to mention earlier. Previously, if you were building an application on top of Office we would have told you to get a new product called the Visual Studio Tools for Office. It is a different product from VS Pro and VS everything else, kind of thing. One of the pieces of feedback that we received from our customers was ‘There are more and more developers who are interested in customising or building things on top of Office and do not make it painful for us to go and get a different product. Why do you not just make access to that set of tools easy for us?’ We took that feedback and said we are going to take all of the Office tools functionality and we wanted to integrate it into the Visual Studio Professional products, so that each and every developer out there who has access to a VS Pro product automatically has access to Visual Studio Tools for Office. Now you can do whatever the blueprint is going to offer you using Visual Studio Tools for Office. The problem if you just use Visual Studio Tools for Office is you will have to figure out the nitty-gritty details of what code you need to write, where you need to write it, how you plug in to servers to enable your application. What we are doing now is providing a set of frameworks that abstract from all those complexities and giving you some guidance on how to do that, so that we make it a great deal easier with the blueprint to be able to do that. You can do it without the blueprint; it just makes life for developers a whole lot easier. That is how I think about it.

Participant

As there are no further questions, thank you for attending.

S Somasegar

Thank you all.

**This Full Transcript Report was produced by Ubiqus Reporting
🕽 +44 (0) 20 7269 0370**