

La stratégie de Microsoft concernant Visual FoxPro

Aperçu historique des 15 années d'évolution de Visual FoxPro et de sa stratégie chez Microsoft

Ken Levy

Microsoft ne publiera jamais le code source de VFP en open source car il n'y a aucune raison économique pour le faire et de nombreuses raisons pour ne pas le faire. L'évolution de VFP dans la prochaine décennie se fera avec des produits nouveaux ou améliorés qui fonctionnent avec VFP, des projets communautaires comme VFPX, et bien sûr par la communauté VFP elle-même.

La stratégie initiale de Microsoft pour FoxPro

L'achat de Fox Software pour 173 million de dollars en 1992 était très stratégique pour Microsoft, cela a été le plus gros rachat d'entreprise que Microsoft ait fait jusqu'alors. dBase de Aston-Tate était encore populaire, Borland avait Paradox, et il y avait PowerBuilder qui était le roi des applications client/server. Microsoft attendait trois choses dans le rachat de Fox Software : l'équipe de développement de Fox, la technologie de Fox, et la part de marché de la clientèle FoxPro/FoxBase. Microsoft venait juste de commencer à travailler sur Access, c'était cibler des utilisateurs de plus de puissance, même s'il y avait des chevauchements. Visual Basic en était à ses débuts.

Il y avait une estimation de 500 000 développeurs FoxPro à son apogée autour de l'année 1995, et des millions d'ordinateurs utilisant des applications FoxPro (avec DOS ou Windows). Il a fallu près de 4 ans, et pas avant VFP 5.0, pour que Microsoft concentre d'avantage sa stratégie autour de VB et moins pour VFP. Naturellement, la base clientèle VFP et le nombre des ventes sont passés de croissants à décroissants ; comme le dit le proverbe dans le monde des affaires : *il faut grandir ou mourir*.

Les premières années après l'acquisition de Fox Software, Microsoft a déployé un effort énorme et beaucoup de ressources dans la création de VFP 3.0. Il y avait environ 50 personnes dans l'équipe de développement de Fox avec un budget marketing important. Les années suivantes, à la fois Access et VB ont gagné des parts de marché et également concurrencé le marché de VFP (et sa communication), et au moment où VFP 5.0 est sorti, nombre de haut gestionnaires Microsoft voulaient mettre fin à VFP à ce moment-là.

En fait, ils l'ont fait pour une courte période. J'étais là, dans une réunion de 40 personnes, où l'annonce officielle a été faite à l'équipe de développement de Fox que VFP était mort. C'était en tout début de 1996, et cette réunion a été relayée par le rapport du Gartner Group sur l'annonce de l'enterrement de VFP, cela a eu un impact majeur sur les ventes ultérieures de VFP. Mais les membres de l'équipe de développement et sa communauté faisait tant de bruit, également avec des personnages comme Eric Rudder (architecte de VFP 3.0, qui devient l'assistant technique de Bill Gates), qu'ils ont convaincu les gestionnaires des outils de développement qu'il était possible de maintenir VFP et le faire évoluer tout en diminuant ses ressources. En fait, la raison principale pour laquelle

VFP a duré une dizaine années de plus, avec 4 versions successives, tenait d'avantage aux ventes de Windows qu'à celles de VFP. Car il y a beaucoup de machines sous Windows qui utilisent des applications VFP. Quand Steve Balmer lance à corps et à cris « développeurs, développeurs, développeurs », il pense plus aux ventes de Windows qu'aux ventes d'outils de développement.

L'histoire de Visual FoxPro chez Microsoft

Pour chaque nouvelle version de VFP comme 6.0 et 7.0, il y avait moins de ressources, moins de membres dans l'équipe, et moins de budget marketing. Lorsque VFP a été inclus dans la boîte Visual Studio, c'était juste un paquetage et non une intégration. Quand la communauté VFP a vue que VFP était dans le Visual Studio et a envisagé la possibilité de fonctionner dans le cadre du Framework .NET, ils ont aussi compris que si VFP prenait cette voie, cela risquait sérieusement de briser la compatibilité ascendante des programmes FoxPro et de remettre en cause son interface de développement.

Microsoft n'a jamais eu comme but de travailler à la fois sur VFP et sur un VFP pour .NET. Produire un nouveau VFP pour .NET aurait enlevé des ressources utilisées pour faire évoluer VB.NET et C#, et rendu plus difficiles les ventes de Visual Studio et du framework .NET, et cela pour un résultat ne garantissant pas de faire fonctionner de vieilles applications VFP telles quelles.

Ensuite, après VFP 7.0, il a été décidé de garder VFP comme un produit autonome en dehors de la solution Visual Studio, car il ne faisait pas partie de la plateforme .NET. Cela avait un sens, et a permis à VFP d'avoir son propre scénario de nouvelles versions. Pour toutes les nouvelles versions lancées ensuite, il était prévu que ce soit la dernière. Personne dans l'équipe de développement de Fox, pas une seule personne ne s'attendait à ce qu'une version après la VFP 8.0 puisse être produite.

Les ventes ont continuées à diminuer chaque année, tout comme le budget marketing. La seule façon d'accroître les ventes de VFP aurait été d'être en compétition avec Visual Studio et de prendre sur les budgets et ressources de Visual Studio. En réalité, le plus gros concurrent de VFP, c'est Visual Studio (et non Delphi ou tout autre produit non Microsoft). La plupart des gens du marketing et les gestionnaires chez Microsoft aurait préférés que les développeurs Fox utilisent Delphi.NET de Borland plutôt que Microsoft VFP, car ils auraient été utilisateurs de la plateforme .NET plutôt que du vieil héritage de COM. COM est devenu un ennemi, dont il fallait éloigner les gens, tout comme le HTML/JavaScript est également un ennemi aujourd'hui pour la plateforme Microsoft.

Pour VFP 9.0, il n'y avait que 8 personnes dans l'équipe de développement, alors même que, VFP 9.0 était une meilleure version que VFP 7.0 et VFP 8.0 d'après la communauté unanime. Peu de temps après la sortie de VFP9, Microsoft a décidé de créer et d'offrir un complément Xbase en vue de maintenir les ventes de mise à jour et d'améliorer l'image de VFP. Cela a été le plan Sedna, un ensemble d'applications, d'utilitaires et d'exemples simples en téléchargements qui sont orientés vers l'interfaçage avec d'autres produits Microsoft (.NET, SQL Server, Office, Windows, etc.). L'autre raison d'être de Sedna a été de retarder l'annonce de la fin de VFP, pour sauvegarder les ventes de VFP 9.0 autant que possible mais aussi de protéger la communauté FOX et ses développeurs sur le marché du travail.

Le code source de Visual FoxPro ne sera pas versé dans le domaine public

La demande à Microsoft pour qu'il donne le code source de Visual FoxPro au domaine public est courante (et logique). Voilà en gros pourquoi Microsoft ne communiquera jamais le code source de Visual FoxPro. Il y a dans VFP de la technologie, comme Rushmore optimisant les indexations, qui est utilisée dans d'autre produit Microsoft comme SQL Server et Access. Ce n'est pas le même codage C/C++, mais les techniques et les algorithmes proviennent originellement de VFP.

Microsoft considère cette propriété intellectuelle comme un actif Microsoft et ne veut pas le publier.

Mais les deux plus importantes raisons n'ont rien avec les raisons ci-dessus. Elles ont rapport avec le business. Microsoft n'est plus axée sur les ventes de VFP, elle est centrée sur les ventes de Visual Studio et l'adoption de sa plateforme globale (ensemble de produits et services). Verser le code source de VFP en domaine public se traduirait par moins de développeurs VFP utilisant l'actuelle (moderne) plateforme de produits Microsoft, mais pourrait également aussi permettre à une personne ou à une entreprise de créer un produit compétitif contre Microsoft lui-même. Microsoft ne voudrait pas avoir son code utilisé pour un produit concurrent ou un nouveau produit qui interfère dans avec ses ventes de Visual Studio ou dans l'adoption de la plateforme .NET.

La stratégie finale de Visual FoxPro chez Microsoft

Au cours des 5 dernières années de Visual Foxpro chez Microsoft, alors que j'étais chef de produit (marketing / communauté) pour VFP, la stratégie était de commercialiser VFP à la communauté existante (principalement via les mises à jours), et de faire tout ce qui était possible pour maintenir une forte communauté Fox, et de faire en sorte que les développeurs VFP adoptent d'autres produits Microsoft (.NET and SQL Server).

J'ai toujours considéré que mon rôle avait deux faces : l'une en tant que représentant de Microsoft comme salarié, et l'autre en tant que membre de la communauté FoxPro pour faire tout mon possible pour Fox à l'intérieur des murs de Microsoft pour le faire évoluer, le sauvegarder, le promouvoir et aider VFP et sa communauté autant que possible. J'ai passé presque la moitié de mon temps à « vendre » VFP au sein même de Microsoft, au siège à Redmond et dans ses bureaux externes, mais aussi à communiquer pour faire passer le message positif d'un VFP vivant.

Les gestionnaires chez Microsoft sont au dessus de l'équipe de développement du cœur de Fox, ils sont les décideurs pour tout ce qui est lié à la stratégie de VFP. Personne dans l'équipe de développement du cœur de Fox n'avait la capacité de décision concernant les budgets de marketing ou des ressources pour les équipes, ou pour ce qui suivait la livraison de chaque version. Il y avait quelques personnes clés dans l'équipe qui, travaillant ensemble, ont sans doute prolongée la vie de VFP de plusieurs années par au moins une version supplémentaire.

Je pense que la fidélité des clients a été un autre facteur dans la durée de vie prolongée de VFP au delà de la version 6.0.

Mais au moment où VFP 9.0 a été lancé, le montant des ventes de toutes les versions de VFP confondues pour une année étaient inférieur au chiffre d'affaire d'une seule journée de Visual Studio.

Le coût d'évolution de VFP relatif au chiffre généré (le retour sur investissement – le ROI) était de loin bien moindre à celui de mettre des ressources sur Visual Studio et les langages .NET. De plus, certains membres de l'équipe de développement étaient prêts à aller de l'avant ou à quitter Microsoft, et il était presque impossible de trouver des gens qualifiés pour les remplacer.

Cela aide à tout mettre en perspective si vous pensez à Visual Studio comme le produit concurrent de VFP, même s'il était détenu par la même société. Rappelez-vous quand Apple a travaillé sur les ordinateurs Mac et Lisa en même temps, un seul a survécu. Dans le cas de VFP, il a survécu une décennie entière après avoir été tué pour l'essentiel (il n'est de ce fait plus stratégique).

Dans la prochaine décennie

Le 15 janvier 2010, le support standard de VFP 9.0 prendra fin. Bien que le support payant étendu doive exister pendant 5 ans de plus, je ne m'attends pas à des correctifs logiciels supplémentaires (hotfixes) ou à ce que quoique ce soit d'autre soit fait pour VFP, sauf dans le cas improbable impactant le runtime de VFP pour Windows 7 empêchant aux clients ayant des applications VFP de migrer sur la dernière version de Windows.

On peut considérer que Microsoft a tué VFP avant que ça ne soit vraiment nécessaire, mais on peut avoir un point de vue historique et voir que VFP a vécu de très nombreuses années et eu plusieurs versions bien au-delà de ce qu'il était prévu.

Bien que Microsoft aurait pu faire d'avantage pour VFP, cela ne pouvait pas réellement se faire simultanément avec la promotion de Microsoft et les ressources nécessaires pour ses produits Access, VB et Visual Studio.

Seuls les développeurs qui ont utilisés FoxPro l'apprécient vraiment pour ce qu'il est et pour ce qu'il est encore.

Ken Levy est le président et fondateur de MashupX, LLC, spécialisé dans le conseil pour les communautés bâties autour de produits et de services, les techniques de consulting marketing ciblées, la création multimédia et les techniques logicielles. Ken est le co-animateur de CodeCast un podcast associés à Code Magazine. Avant de commencer MashupX, Ken a travaillé chez Microsoft comme chef de produit pour Visual FoxPro, puis responsable de l'équipe de la plateforme Windows Live, et enfin gestionnaire du programme communautaire VSX (Visual Studio Extensibility). Ken est un membre bien reconnu de la communauté FoxPro, il a créé GenScrnX pour FoxPro 2.x et de nombreux composants VFP y compris l'explorateur de classes. Vous trouverez Ken sur twitter @KenLevy, sur son blog à <http://mashupx.com/blog/> et vous pouvez aussi le à contacter par email klevy@mashupx.com.

Visual FoxPro Strategy at Microsoft

Historical overview of the 15 year evolution and strategy of Visual FoxPro at Microsoft

Ken Levy

Microsoft will never release VFP source code into open source because there is no business reason to do so and a list of reasons not to. The evolution of VFP goes on into the next decade with new and enhanced products that work with VFP, community projects such as VFPX, and of course the VFP community itself.

Microsoft's Early FoxPro Strategy

The purchase of Fox Software for \$173 million in 1992 was very strategic for Microsoft, and was the biggest corporate purchase Microsoft had ever made up until that time. Aston-Tate's dBase was still popular, Borland had Paradox, and there was PowerBuilder as the king of client/server tools. Microsoft needed three things from the Fox Software deal - the Fox developer team, the Fox technology, and the customer market share of FoxPro/FoxBase. Microsoft was just starting work on Access and it was more targeting power users, but there was still some overlap. Visual Basic was still in its early days.

There was an estimated 500,000 FoxPro developers at the peak around 1995, and millions of computers with FoxPro apps running (either DOS or Windows based). It took almost 4 years, not until after VFP 5.0, for Microsoft to focus more strategy around VB and less for VFP. Basically, the VFP customer base and sales went from increasing to decreasing, as the saying goes in business: *if you aren't growing, you're dying*.

In the initial years after the Fox Software merger, Microsoft put a huge effort and lots of resources into creating VFP 3.0. There were about 50 people on the Fox team with a big marketing budget. In the following years, both Access and VB grew in market share and also competed in ways with the VFP market (and messaging), and by the time VFP 5.0 was released, many upper managers wanted Microsoft to just end VFP there. In fact, they did for a short time. I was there, in a meeting with 40 people, and the formal announcement was made to the Fox team that VFP was dead. It was very early 1996, and that meeting lead to the Gartner Group releasing their report that VFP was dead, which had a major impact on future VFP sales. But the Fox team members along with the community made so much noise, combined with people like Eric Rudder (architect of VFP 3.0, who became Bill Gates' technical assistant), who convinced developer tools management to keep VFP evolving while decreasing the resources. In fact, the primary reason VFP lasted another decade with 4 more versions released was more about Windows sales than VFP sales. There are many Windows machines running VFP apps. When Steve Ballmer jumps around like monkey boy and yells "developers, developers, developers", he's thinking about selling Windows more than sales of developer tools.

History of Visual FoxPro at Microsoft

For each new version of VFP like 6.0 and 7.0, there were less resources, team members, and marketing budget. When VFP was included the Visual Studio box, it was just a bundle, no integration. When the VFP community saw VFP running inside Visual Studio and the possibility of running on the .NET framework, they also learned that if VFP went that path, it would seriously break FoxPro code backward compatibility and the VFP IDE would be gone eventually. Microsoft never had a goal to work on both VFP stand-alone and VFP for .NET. Having a new VFP for .NET would just take away resources from evolving VB.NET and C#, make it harder to sell Visual Studio and the .NET framework, and not really result in anything useful since it would not run old VFP code as-is.

Then after VFP 7.0, it was decided to keep VFP as a stand-alone product outside of the Visual Studio bundle, since it would not be part of the .NET platform. This made sense, and allowed VFP to ship on its own new version timeline. Each new version released, it was expected that was the last version. Nobody on the Fox team, not a single person, expected a version after VFP 8.0 released. Sales continued to decline annually, and so did the marketing budget. The only way to grow/increase sales of VFP would have been to compete with Visual Studio and take away budget and resources from Visual Studio. In reality, the biggest competitor to VFP was Visual Studio (not Delphi or any non-Microsoft product). Most marketing and management folks at Microsoft would have preferred Fox developers use Borland's Delphi.NET rather than Microsoft VFP, since they would be building on the .NET platform rather than the old legacy COM. COM became the enemy, to move people away from it, just like HTML/JavaScript is a current enemy to the Microsoft platform today as well.

For VFP 9.0, there were only about 8 people on the Fox team, and even so, VFP 9.0 was a better release than VFP 7.0 and VFP 8.0 according to the community. Soon after VFP 9.0 released, Microsoft decided to create an Xbase add-on to give away in order to maintain initial upgrade sales and an upbeat perception of VFP. The result was a plan for Sedna, a download of useful sample apps and utilities that focused on VFP interop with other Microsoft products (.NET, SQL Server, Office, Windows, etc.). The other reason for Sedna was to delay the announcement of the end of VFP in order to save sales of VFP 9.0 as well as to protect the Fox community and the job market for VFP developers.

No Open Source of Visual FoxPro

The request for Microsoft to make the Visual FoxPro code base open source is a common (and logical) one. Here is some insight to why Microsoft will never release Visual FoxPro source code into open source. There is technology in VFP, like Rushmore optimized indexing, that is used in other Microsoft products SQL Server and Access. It's not the same C/C++ codebase, but many techniques and algorithms originated from VFP. Microsoft considers this intellectual property, an asset Microsoft does not want to be released.

But the two more significant reasons have nothing to do with the reason above. They have to do with business. While Microsoft is not focused on sales of VFP, it is focused on sales of Visual Studio and adoption of the overall Microsoft platform (stack of products and services). Releasing VFP into open source would result in less VFP developers using the current (modern) Microsoft platform of products, but may also result in someone or some company creating a competitive product against Microsoft. Microsoft would not want to see the code used to enhance a competitive product nor would they want to see a new product created that interferes with Visual Studio sales or .NET platform adoption.

The Final Strategy of Visual FoxPro at Microsoft

In the final 5 years of Visual FoxPro at Microsoft, while I was the last Product Manager (marketing/ community) for VFP, the strategy was to market VFP to the existing community (mainly via upgrades), to do what was possible to keep the Fox community as strong, and to get VFP developers to adopt additional Microsoft products (.NET and SQL Server). In my role, I always viewed myself in 2 positions - one representing Microsoft as an employee, and the other as a FoxPro community member doing everything I could for Fox within the walls of Microsoft to evolve, save, promote, and help VFP and the community as much as possible. I spent nearly half of my time marketing VFP within Microsoft, at the Redmond headquarters and to the field offices, on messaging and keeping the VFP message positive/alive.

Management at Microsoft above the core Fox team were the decision makers for things related to VFP strategy. Nobody on the core Fox team had decision making ability around marketing budget or resources for the team, or what was done after each version shipped. There were a few key people on the team who, working together, probably extended the life of VFP an least one additional version and several years.

I think the customer base loyalty was yet another factor in how long VFP was extended beyond 6.0. But by the time VFP 9.0 was released, the amount of sales for all versions of VFP combined annually was less revenue than Microsoft sales of Visual Studio in only one day. The cost to evolve VFP relative to the amount of money it generated (ROI) was far less than putting more resources into Visual Studio and .NET languages. Plus, some Fox team members were ready to move on or leave Microsoft, and it was nearly impossible to find qualified people to replace them. It helps to put this all into perspective if you think of Visual Studio as a competitive product to VFP, even though it was owned by the same company. Remember when Apple worked on both the Mac and the Lisa computers at the same time, only one survived. In the case of VFP, it survived an entire decade after it was essentially killed (by it no longer being strategic).

Into the Next Decade

On January 15th, 2010, VFP 9.0 standard support ends. While paid extended support will exist for 5 more years, I don't expect any additional hotfixes or anything to be done for VFP, unless in the rare case it impacted VFP runtime on Windows 7 preventing customers with VFP based apps to upgrade to the latest version of Windows. Some suggest that Microsoft killed VFP before it should have, and another way to look at the behind the scenes history is to see that VFP lived many years and versions beyond what it was planned. While Microsoft could have done more for VFP, it just really couldn't happen with Microsoft promoting and giving resources to Access, VB, and then Visual Studio at the same time. Only developers who have used FoxPro really appreciate it for what it was and what it still is.

Ken Levy is the president and founder of MashupX, LLC, specializing in consulting for community building around products and services, guerilla marketing techniques, multimedia creation, and software technology. Ken is the co-host of CodeCast, a podcast show associated with CODE Magazine. Prior to starting MashupX, Ken worked at Microsoft as the product manager for Visual FoxPro, a product planner on the Windows Live Platform team, and the community program manager for VSX (Visual Studio Extensibility). Ken is a long time recognized member of the FoxPro community, created GenScrnx for FoxPro 2.x, and many VFP components including the Class Browser. You can find Ken on Twitter @KenLevy, Ken's blog at <http://mashupx.com/blog>, or contact Ken at klevy@mashupx.com.