

From: Steven Sinofsky (Xenix)
Sent: Wednesday, December 20, 1995 4:11 PM
To: Bob Eshelman (Xenix); David Heiner (Xenix)
Cc: Greg Maffei (Xenix); Paul Martz (Xenix); Pete Higgins (Xenix); Richard Fade (Xenix); Bill Gates; Chris Peters; Nathan Myhrvold; Steve Ballmer
Subject: Vermeer Meeting Summary

ATTORNEY CLIENT PRIVILIDGE

Highlights of our meeting with Vermeer 12/19. Chrisp, SteveSi, DeniseW

We had a chance to answer questions at some length to all of the employees. In general, the questions were what we expected and showed that the concerns held are not unreasonable (relocation, being "swallowed", lack of creativity, being forced to do things a certain way, how important is this to our strategy, etc.)

Their biggest strategic question, other than how important Vermeer would be to Microsoft, had to do with the overlap to other Microsoft efforts, mostly for them Internet Assistant was competition. To them, HTML isn't the big advantage as much as having the end-to-end tool. They were concerned that we might be putting in more end-to-end stuff in Office, which would water down FrontPage. We explained this by saying that to us HTML is just another ASCII format (TXT=HTML) and therefore every group at Microsoft will support the format in some way, though they will not be end-to-end "web document" tools like FrontPage, but will supply those individual nodes in the web tree.

Part of talking about this we spoke about how we are aiming for the broadest set of users, much like Word chose not to aim for the high end DTP market and PowerPoint does not aim for teh Director market, but chooses to add enough of those high end features to make the documents very rich but still easy to author for people that "have other jobs". This surprisingly fit with their own internal understanding of where they want to aim Vermeer. They feel that tools like NaviPress and iBand (they didn't say by name but hinted at) are aimed for "WebMaster" people, which aren't FrontPage customers. Scripting is a feature for them, but only interesting if it at the level of VBA in Excel (simple code behind a single document). They don't see FrontPage as becoming a VB programmers tool.

Overall I think we left them with a positive impression of both us and Microsoft. Chris definitely came across as very enthusiastic and personally invested in this possibility. There was a definite connection between chris and the group I think. I also feel good enough about the code and processes at this point, though we really only had a very cursory look at any code (a couple of hundred lines, little in any depth or detail).

Overall, the biggest thing that became clear during the visit was that the group is entirely focused on the single problem of authoring a set of documents on the Web. Although they have feature teams focused on the editor, bots, server, and explorer, the real focus is on the average end-user trying to create a series of related/linked documents and handling the management of them. This is their "super document" notion--they are in fact authoring a single new document type with a single new tool, not just authoring a bunch of traditional documents and using some tool to manage them or add some behavior. Although editing single pages is an important part of this, they view it as just one step in the process. Each portion of FrontPage is aimed at solving the end-to-end problem, not at being the premier component--this is expressed across the board by the team as both their development/testing emphasis and their general philospfy for making trade-offs.

In terms of team dynamics, each of the people we spoke with 1:1 as a feature team lead spent a great deal of time discussing the dynamics of building the entire product. For example, Andy (the server extension lead) spoke a great deal about how important it was for him to work with the leads of the authoring tools. They also talked about the movement of people between each of the feature teams as something that was fluid and critical.

Our day was a meeting with key people, followed by an open Q&A with everyone, followed by 1:1's, then a wrap up with everyone for more Q&A. The end of the day was very good as people came back with more questions. During the day DeniseW met with people to discuss individual cases in some detail which was comforting.

Overall, they were somewhat more in the loop than we expected, so we could have been more prepared with specifics than we were. They seem to have been sold on this as "come out to redmond for 2 years and see what happens" rather than as a long term commitment to Microsoft. This might explain why people were so positive. We will have to sell on permanency.

We met 1:1 with the key folks in development. Frank Germano (Lotus and then Beyond before Vermeer) is the experienced scheduler and manager. He does drive the schedule whip and manages milestones, but has little technical input to the product. There is another key key developer in the bay area that Peter knows from Metaphor (Scott?). He is enthusiastic we are told, but I think it is necessary to get him since he wrote the editor kernel. We also talked for a while 1:1 with Ron (?) who designed the u/i for the Explorer. He came from the Matlab company (statistics/math editor) and seemed to be concerned about being a lead (having previously managed 12 people working for Randy's wife). He struck me as a program manager person or someone who would be a good development lead but give program managers a hard time.

Andy and Peter are the real development leads. Consensus among charles, randy, and john is that Andy should be the next dev mgr. Unfortunately both chris and I independently arrived at the conclusion that Peter was the stronger of the

MS-PCA 2599141

Plaintiff's Exhibit

9602

Comes V. Microsoft

HIGHLY
CONFIDENTIAL

MS-PCA 000002599141

two.

Peter is an ex-Metaphor person which explains in many ways the nice user-interface and well thought out user model. He is enthusiastic and very passionate about the product--very much in the Microsoft way. His questions centered around how we do things and where our priorities lie with the product. Andy runs the tools side of development (Editor and Explorer and the client-side of Bots)

Andy was a little more distant and skeptical. His concerns were all over the map, none of which were really central to the product. He was concerned about cross-platform issues, C++ v. C, and what source code control system we used. I still don't have a warm feeling about him. He comes to Vermeer via DEC and then Apollo.

Randy continues to be the star and I really feel as though he was the driving force behind the product, though the real driver still seems to elude me. He is a pleasure to talk with, understands the product and its role. Clearly the leader among folks.

In looking at the code things are pretty much what we expected. There are about 300,000 lines of code overall. There are really two major code bases: the tools and the server extensions (and server-side of BOTS).

The tools (explorer and editor) are MFC apps through and through. They used appwizard to create the apps (OLE enabled) and classwizard for the dialogs and message handling. They followed the general conventions pretty much as expected though decided not to follow naming conventions. The code looked pretty reasonable, though lacked the rigor of a larger Microsoft application (in terms of ASSERTS and very strict error checking). They used MFC 4 exceptions (native C++ ones that are fat) for some non-essential things, which is not too good. They feel that in the 180,000 lines of Tools code there are probably 100 or so try/catch blocks. Peter claims to have done extensive work to avoid GDI leaks though much less to avoid memory leaks. They don't run under the debug kernel religiously and when they do only some of the developers use it. Very few globals or even CWinApp level variables, which is good.

They were not consistent about isolating strings in resources and from the code we saw there is definitely a risk for DBCS work. Lots of *p++ sorts of things and a number of strings buried in the code. Peter talked about how important that is (metaphor is a UNICODE company) but even strings that we put in resources in MFC made their way out of MFC (for example the COMMDLG file type strings). I think given the volume of code it could be easily 2 months of an experienced person to make things DBCS enabled and working.

The most interesting thing I learned about the Tools is that the Editor does not do any WinSock or HTTP calls--rather it uses the code in the Explorer to do this and controls it via OLE automation. This means the editor does not follow links as a stand-alone product. This could be OK, but is something to note.

The most concerning thing still remains scalability. They clearly understand that the product will probably not scale to anything more than 1000's of pages, and even then in the low 1000's. This is something that will need to be tested and looked at very hard, since I feel our customers are likely to hit this very quickly. They know about this problem and indicated only some vague notion of breaking things up into sub-webs, which just sounds like a hack to avoid loading all the pages at once (Explorer does read in all of the pages). If there is an Achilles' heel this is it, but they understand the problem and there are a lot of solutions (from just implementing with a better ISAM to using a system service).

The server extensions really don't seem all that complex. Their main functionality is to provide a framework for the CGI apps and then implement that functionality. The code is decidedly different from MFC/C++ since it involves no GUI. There is a distinct UNIX bias in this code--in the style, choice of names, and idioms. Andy is a little computer science oriented in the way he talked about problems and solutions, though clearly he understands the whole product and the role the server extensions play. As an indicator his primary machine is a UNIX machine and he seems to be the driver for the tools environment (the source code is all golden on a UNIX server).

As far as process, they really seem to have their act together on this as good as any small company I have seen--realizing what a challenge they have with all the platforms and releases. They have a web-based bug database. They use CVS and branching (for all their platform work on the server extensions) for source control. They have a full time release person. The only downside is that they are not religious about daily builds all the time, having just restarted for the 1.1 release. The build process seems fairly complex, with only some of it attributed to their platform matrix. I noticed a number of custom tools going buy during the build, though they used a VC++ project file.

As far as 1.1 it looks like the table editing will be nice but not 100% (no nested tables, which is ok for the real world but could be a review ding) and it looks like they will be doing some link management.

Things I think we should do in follow-up, along with huge list denise has:

- make sure we introduce key people to bobmu, chrisjo, bens, adamb, jallard, robbieb
- get them some copies of *Debugging the Development Process* and *Writing Solid Code*
- be sure to find time for them to speak with CLWILL regarding the similarities to Fox acquisition
- get them the artwork for Volcano coffee from stevesi for a possible demo
- get them a few "Win95 Logo" shirts from stevesi
- pick mentors from DAD for everyone by functional area

MS-PCA 2599142

**HIGHLY
CONFIDENTIAL**

MS-PCA 000002599142

- decide on program managers v. product planning for some people
- see if we can get them a VC++ 4.1 beta
- make sure when we talk again that we are careful to listen to their individual concerns a little bit more
- next time we talk make an effort to play down the role of switching to Microsoft tools (something cwill can talk about)

MS-PCA 2599143

**HIGHLY
CONFIDENTIAL**