

From: Bill Gates
Sent: Friday, February 18, 2000 5:01 PM
To: Eric Rudder
Subject: FW: Our future and the Red Herring Article: Is Microsoft Losing its Grip?

FYI...

Maybe you and I need to meet with Brian.

—Original Message—

From: Brian MacDonald
Sent: Friday, February 18, 2000 2:36 PM
To: Bob Muglia; Bill Gates; Steve Ballmer
Subject: Our future and the Red Herring Article: Is Microsoft Losing its Grip?

I know how to fix this. Will you guys let me? I don't know.

<http://www.redherring.com/insider/2000/0217/tech-microsoft021700.html?id=yahoo>

COM+ and WebForms isn't going to fix this. That's not to say they aren't important. Fixing this means that the 90 million Office customers are considered strategic and that a new business must consider how they integrate their web services into Office or be relegated to the consumer market. Office is so out of the game on this now. The world is very rapidly becoming a universe of URLs and Office doesn't know what to do with them. It also really doesn't matter much to users whether those URLs came from Microsoft technology. With COM+ maybe we can make it matter to developers but how much is that really going to buy you.

WHAT WE STILL HAVE A GOOD SHOT AT IS MAKING IT MATTER WHAT CONTEXT THE UNIVERSE OF URLS PLAY IN. That might be the most misunderstood part of NetDocs. The framework and platform for web services to tightly integrate with email, PIM, authoring, viewing is KEY KEY KEY. Its mostly now a client framework but within our overall knowledge worker service we can lead and add value with middleware servers that aggregate and normalize XML data & serve up interesting views to the client. (this is different than message bus - both are important). The point is that a rich framework client & middleware can make this URL universe a less random place and one where targeting Microsoft's high end service is important not irrelevant.

I'm not sure I've done a good job in getting buy-in that the platform has trifurcated into a stack. At the bottom level, Windows is still important but its importance is as the underpinnings of the platform layers on top. I.e. its importance as a platform is hidden and the next higher level platform is what needs pushing and evangelization. For users and developers, its importance is at the device driver level. The shell is a historical artifact, exposes a mess along with the good, and isn't of much go-forward value. People are going to fell allegiance to one shell and that will be the one that goes with their core service offering. Beyond being a default, the leverage of the OS isn't what it use to be. We'll still take it though - but like with MSN an inferior one won't get a lot of traction just because its the default.

The next tier is the browser / Trident level. This is a super important level and we've done a great job. We did a great job because Silverberg prioritized and focused on it and Adam brought a great concentration of talent onto the problem. We achieved our goal of ousting Netscape. The good news is that we own the platform. The bad news is that its so free & generic that beyond reinforcing the Windows layer underneath and preventing someone else from stealing that limelight, it doesn't go further. I.e. its was our defense but not our offense.

One other thing about this layer is that it showed that once a good abstracting value add layer existed, the development world flocked to it. A large part of the flocking was because the abstraction allowed for a bigger flock. Big influential things are done by companies with marginal talent that would have a tough go of things making GUI apps.

The last point about this layer is that while we rocked in creating it we have let this team go from being the best at MS to a B class. Letting the team implode is one of our biggest fuckups of the last year.

Then there is the shell / viewer / core service platform on top. The value add for third party developers is to integrate with the core schemas for Email, PIM, documents, to attach to the right contexts in the shell, to be able to assume & target an on-line doc type based on XML that can edit payloads & spit out HTML, that can persist state info in a better way than cookies (e.g. fully roamable), that can leverage viewing controls, that can piggyback on logon services, that can piggyback billing services (eventually) without a huge infrastructure investment, that exposes Exchange storage services, etc. Like with the Trident layer, this layer provides an even greater abstraction that can increase the size of the flock than can build on our platform. It also reinforces both layers below it.

There is a network effect with adding services to a rich feature set at this level of the platform. If we can get to critical mass, we can make the Office level as essential as the Windows level but even more relevant to the URL world. We also get to double dip the revenue stream.

All of that still adds up to running in place. Not quite because we can increase Office penetration but at least its the same order of magnitude. We can go beyond by better integration of core value-add services than anyone else.

Bob is a bit chagrined that I've done a poor job driving home the points about the platform aspect of NetDocs with the two of you. I agree. Its key to get buy in from you two in order to support the rest of what needs to be done to make a NetDocs/Exchange knowledge worker service world class in the first run. Its the KW service but the platform can migrate to the OS and become what Neptune never was and the skin can change and be what Mars should be from a platform perspective.

MS-PCA 1451995
HIGHLY CONFIDENTIAL