



From: Richard McAniff
Sent: Friday, November 15, 1996 12:51 PM
To: Steven Sinofsky; Jon Reingold; Jon DeVaan; Craig Unger; Ralf Harteneck; Peter Pathe; Dawn Trudeau; Brian MacDonald
(Exchange); Richard Fade; Chris Peters
Cc: Richard McAniff
Subject: RE: Office 9 focus areas

The process that Stevesi is proposing below is the right start. I'd like to add to this process by putting down into words some of the ideas that I have been thinking about recently. A lot of what I am about to write is not new, but represents a collection of different peoples thoughts, and is the result of conversations (and email exchanges) I've had with everyone on the To:line, as well as conversations with people such as Bobmu, Adamb, and Darrylr and others.

From a purely process perspective, I believe we need to have an overriding framework which we (and our teams) can use to help make decisions/tradeoffs about Office. This involves four things:

- Identifying and agreeing on a set of threats that we think are most important
- Agreeing on a Vision for the Office Product
- Agreeing on a set of constraints (assumptions) that will help us when making decisions
- Identifying key scenarios that Office9 will be evaluated against

I've taken the liberty to put forth a strawman proposal with regard to each of these areas. This is not meant to be exhaustive, but is meant to stimulate discussion so that we can make forward progress.

Threats. I think we all would agree that we face multiple threats. Because of this, it is pretty easy to get confused about what should be done, or which priority is most important... i.e., should we build that next feature into product x, or should we put more efforts into a common feature in Office, etc., etc. We need to all agree on the set of threats that are most critical and then plan accordingly.

- Office itself is the competition. At one end of the spectrum, we are concerned about upgrades, and the very real possibility that our customers will not upgrade to our newer versions of Office. Customers are very concerned about the real cost of upgrading and the problems they experience when moving to new versions of Office. The costs of upgrading (installation, downtime, etc) are weighed against the benefits of higher worker productivity. When the gains are not substantial, people stop upgrading. In this case, the last two versions of Office is the competition.
- Office is a dinosaur. The Internet represents a fundamental change in at least our runtime strategy, and Office may not be well suited to this new environment. Office may be too big, too proprietary, and too tied to legacy formats to integrate smoothly and perform well in a new Web environment. In this view, Office is the mainframe in a Web world. The enemy is a new class of Java apps or applets that will make Office irrelevant in a few years.
- Netscape's Communication/Collaboration strategy. Netscape represents a whole new way of thinking about how people will work and collaborate in the future. The argument here is that Office was built on a set of assumptions that begin to lose validity in terms of what people will want to do in an internet environment. Rich document types that include voice, video, images become increasingly important- our app centricity can hurt us here. An increased demand for searching, navigating, organizing and viewing data become key new requirements. Information flow becomes critical. The threat is that Office documents become less relevant in the space of new documents that are being created. To the extent that we don't optimize for this new environment, our current user base will find Office less relevant over time.
- We fail to attract a new set of users. Stevesi has pointed out that a key threat to Office may be our inability to attract a broadening set of users within corporations. These new users are those people that are not using our products today. These users do not "author" information, rather they consume, review, consolidate and make decisions with the data that is presented to them. Office's feature set is not targeted at the online information consumer, but rather the information creator.
- We fail to attract developers to build solutions using Office. We haven't given this one enough attention in our '98 discussions. It is very important for us to make sure that we have a class of ISVs, SPs, and MIS individuals that target Office as a platform. The current set of Web tools are difficult to use and many of them are proprietary. We will be successful to the extent that we can offer both Office services and Industry standard (or at least Microsoft standard) Web services to these developers.

Vision and Objectives. We need to communicate an overall Vision for the Office product and corresponding objectives. These objectives should be pretty high level and easy to understand. Again, these objectives should translate the threats into actionable areas that we can base product decisions on. Given the above threats, I'll list a set of objectives I think we should consider. I'll leave the Vision statement to someone else.

- Make Upgrading to the new Office9 environment a "No Brainer". This objective is about what we will tell our sales force next year when we are ready to have them sell the next version of Office. It is about the "value proposition" of why people should upgrade. It is not about the Web per se, but about the benefits that users derive from the Office product. When we talk about Office features, we should be thinking about how each feature fits into one of these "buckets".
- TCO - This includes everything from frictionless upgrades, integration with SMS, easy access to our Office Web site, super easy installation, central administration, scalability, etc., etc.
- Office is the best place to "Navigate and Find Information". There is a lot of work that we need to do in order to help Office workers navigate and find information in a world where information is growing at exponential rates. This is independent of the Web, but becomes even more critical in the context of the Web.

MS-FCA 1236010
CONFIDENTIAL

- Office is the best place to Organize your data. This is the flip side of navigating and finding data. We need to help the Office worker easily organize his/her data once it has been found. A logical place to think about doing this is in Outlook which already has the concept of folders, multiple namespaces, viewing and filtering.
- Office is the best place to Communicate and Collaboration. Much has been said and written on this topic, but we need to make sure that each application fully enables collaboration and communication. Users will want to know: who authored this; who else has read it; who can help me with it; who can I respond to about it or discuss it with, etc.
- Office is the best place to author and analyze your data. We need make is super easy to create documents that are rich in content. We also need to make it easy for users to interact and analyze information once they have it on their desktop.

- * Build for the Web. Office needs to look like an integrated product built for the Web. This of course means very different things to different people, so we need to be crisp about what we mean here. In building for the Web, it is interesting to ask ourselves "What's Changed"? In short, we now live in a very connected world driven by internet protocols (TCP/IP, HTTP), common naming conventions (URLs) and a common data exchange format (HTML). The implication is that we can now reach a much larger set of users than in the past. This represents a tremendous opportunity to us. However, this new class of user is more concerned about collaborating and consuming information than about authoring it. They will want much richer documents (images, voice, video, etc) than we have provided in the past. They will need better tools for searching, navigating, organizing, and viewing data. These new "Information Consumers" live on information flow.

There are different ways we can approach the opportunity that the Web provides us. As noted above, I think that Stevesi's focus groups will go a long way in terms of making sure that we have a much more consistent story, but I think that there is a higher level set of issues that we need to address. For example, how far do we go in terms of Componentization, what is our shell story, how does each group take advantage of Normandy, or Exchange or Denali. Here are some ideas Craig Unger and I have been talking about over the last couple of weeks in preparing the Access product plan.

- * Componentization. I'll argue new major product features should be wrapped as components. There are notable exceptions to this rule, but the point here is that we should be thinking about componentizing ourselves as much as we can. For example, Access/Excel should build a reporting component "aka Live Reports" that can be used by Excel and Access, but also is suitable for the Web. If we build a new List manager (Table by data) this should also be built as a component. In the short run it may be easier for products to think about building functionality into their own respective shells, but we should attempt to build new functionality that is "shell neutral".
- * Web UI. We should explore new and innovative ways in which present our UI in the Web browser itself (or within the Office/Outlook shell). If we are successful here, Office will keep the user "in context", since our users will spend an increasing amount of time in the browser. The WEB UI can also serve to simplify our products.
- * Office Client shell. Extend Outlook to be the Office Shell beyond its current functionality to include a rich set of services available to all Office apps. This includes email, and collaboration, finding and filtering data, organizing information (Outlook already does a credible job here). There are other services that this shell can provide such as annotation, UI customization, and incorporating "smart agents" to manage your information. Stevesi rightfully points out that Outlook may already have too many things going on in it to take on additional roles. The key point here is that each app should not have to worry about such areas- imagine if Escher were integrated only once in Outlook instead of by each individual app. We have a tremendous opportunity to really increase the synergy between all the apps, and at the same time provide a set of services and functionality that no app presently is doing. We need to see ourselves living in a world where we add value on top of these shell services.
- * Office Server. We need to build a document server a la BrianMac's memo...i.e., some key features are: automatic replication, better searching and access to documents, content indexing, versioning, check-in / check-out, etc. In some ways, it is less important who builds this, but Office needs to generate requirements so that we can leverage the technology. Right now, we do not have critical mass in Office thinking about this.
- * Leverage each other's technology. We should look at emerging technologies across the company to see what we can leverage. One ripe area to explore is Office/BackOffice integration. Access and Excel both need to leverage SQL server. Access and Outlook need to leverage Exchange. But there are other possibilities too- Office should look at leveraging Denali, Normandy and Merchant Server.
- * Office is the best place to build your "Office solutions". The third objective that I'll propose is all about how we can leverage the developer in building Office solutions. Another way to view this objective is to think about Office as a platform. Note that I am not saying that we need to turn the core Office product into a development environment. In fact, one could argue that we should take VBE out of the core Office product and put it into an Office developer box. We need to leverage the work that is going on in the Tools area, as well as making our individual products play together much more effectively so that people can build solutions using our products.
- * Provide smooth migration into developer focused tools such as VB and Istudio
- * Integrate Technology such as Terra Cotta into Office
- * Think about really investing in a "Developer Office SKU"... we sort of have this with the ODE, but the functionality can be extended.

Assumptions. It is important for us to make several assumptions if we are going to give clear direction to our respective teams. The assumptions are of course tied up in many different areas. First and foremost we need to decide what we think our threats and key areas of innovation are... this will lead to lots of progress. Next there are a bunch of second level sets of assumptions, and as each team starts to move forward, we'll continue to add to this list. I'll list a few areas (primarily for illustration purposes) that need to address. There are many others.

- The role of Trident (I'll argue that we should put a stake in the ground and use Trident)
- Role of Java

MS-PCA 1236011
CONFIDENTIAL

- Cross platform assumptions
- Cross browser assumptions
- Dates and deliverables
- Dependencies on other groups

Scenarios. In order to evaluate whether our product meets the product objectives listed above, it is important to evaluate ourselves in the light of a set of scenarios. We should be able to ask ourselves "Does this help us to solve such and such problem"? Obviously there are literally thousands of different scenarios that we could think about building around. We need to make sure that we have covered the critical ones that Office think are most important... i.e., those that counter the threats listed above. Office as a group needs to address this question, but just as importantly, each product group should define their scenarios in light of what we think the threats/assumptions and objectives are.

- richard

-----Original Message-----

From: Steven Sinofsky
 Sent: Monday, November 11, 1996 11:50 PM
 To: Jon Reingold; Eric Michelman; Jon DeVaan; Andrew Kwatinetz; Alex Loeb; Craig Unger; Larry Engel; Ralf Harteneck; Richard McAniff; Marc Olson; Peter Pathe; Kathleen Schoenfelder; Dawn Trudeau; Brian MacDonald (Exchange); Richard Fade; Chris Peters
 Subject: Office 9 focus areas
 Sensitivity: Confidential

A note up front, these notes are more than we actually talked about so it is possible that there is too much here that folks don't agree with. A lot of conversations have taken place on this subject among program managers and PUMs so this attempts to roll those up into one piece of mail. Any errors or major leaps of faith/logic/guts remain mine.

Comments Welcome!

This mail contains the straw-person feature/focus areas for Office9 we talked about today. These are meant to be focus efforts that last the entire length of the product cycle (i.e. these aren't program manager task forces) and the hope is that we can find a way (from the PM perspective) that all of our features easily fit into these buckets (not in a convoluted way, but in a straight forward way). This will allow us to more easily make tradeoffs and to make sure we have a consistent product that meets some specific needs.

The best way to meet needs is to define scenarios based on research that each of these areas will use as design constraints. For many of these the work is already underway. Some will be harder since they are based on "unarticulated" needs (such as how does office play on internets when everything thinks internet=notepad). The program manager owner of this area working with product planning where appropriate will be responsible for these scenarios. A clear vision for this focus area will include some specific scenarios.

A main goal of these focus areas is to make it easier to decide cross-product issues. The people working on this area are responsible for the implementation of their work in all the applications (Word/Excel/PowerPoint/Access, not Outlook right now). There will be development and testing with a similar focus.

We will also be prioritizing these areas in terms of a product vision which will help to enable groups to make tradeoffs, either in designs within their group or across groups for resource allocation. It is fair to say that at this point TCO ranks #1 (so decisions that are a negative impact on upgrade are bad) and Web Client/Server are tied at #2, most likely siding with the client if we have to since the server will require additional infrastructure which we can't always count on. The other areas are relatively orthogonal after these.

The time frame for this release will be less than 18 months and probably scheduled for around 12-15 months. This depends on two things: the Mac and getting some specs and estimates very soon. The resource allocation to both Office9 and across these areas will be determined based on this first round of feature list generation and normal cuts. The assumption, in terms of process, will be that we will be much more fluid about moving resources across these areas as we get started.

There are some other key assumptions we will generate over time in the areas of: hardware, platform, system services (trident, denali, IIS), working with netscape/apache, etc.

Finally, although there have been lots of discussions about organizations, this mail is about focus and technology. RichardF has been thinking through the organizational issues associated with setting up a structure that will ensure that we can build a great product with the minimal amount of organizational friction.

Major Areas:

TCO: This group owns the issues associated with the deployment and management of Office9 as well as the upgrade and interoperability issues. The key things to think about are setupless applications, self-repairing applications, applications that can discover features, http://office for locating components and installing on the fly, management/logging of the use of office in a corporation (SMS-like/integration). It is important to solve the problems associated with free-seating/roaming

MS-PCA 1236012
 CONFIDENTIAL

users as well. Out setup technology fits in this group as well.

Web Client: This includes work that makes each application a great client on the new browser platform. This means we look into specifics about being an activex control, documents as mail notes, and the use of HTML forms as the programmability container for solutions. The definition of HTML for Office9 would also be in this group. An important part of being a web client is participation in collaboration and what we called workgroup in 97, so this group would own this as well [brianmac].

Web Server – Leveraging a web (nntp/internet) server as a key way for users to store, collaborate, locate, and distribute documents is the central focus of this group. This also includes tracking user preferences (as part of free seating), personalization, and general services for each user of Office that would best be served by residing on a server.

Graphics and PowerPoint – What Ralf said about preparing and delivering presentations. In addition, this group would own taking our graphics efforts and moving them forward.

Word Processing/IntelliSense/Assistance – This area would focus on making the best editing and authoring tool. The core wordmail work would be in the Client team (most likely) but many of the other features mentioned in the word9 plan would fall into this category. Because of Word's traditional focus on ease of use and because of the code issues this group would also own our ease of use message. Assistance would also be driven from here. [See late breaking proposal at the end.]

Components [and Content] – This group would own the component aspect of Office9. This includes charting, OLE servers, and a great deal of shared infrastructure which might include DAO, Jet, etc. (to the degree that we decide no to put these in other groups). This group would also focus on delivering the in-box content, coordinating with the Rossh/SamH group.

Data Access/Reporting/Access/Excel -- The primary focus on this team would be delivering on a strategy for providing the best tools for data on the web for end-users, including analysis and reporting, as well as web db application development for end-users. The plans outlined by Excel9 and Access9 are featured. By having these in one focus area we can drive a strategy and an implementation that complements each of these products. Basic use for spreadsheet input is also in here.

Programmability – The shared development model, object model, language implementation, use of Java, use of JavaScript/events, etc. fall into this area.

Outlook – being the Hub in a great way.

Notes:

Note 1: Andrewk suggested that the Word mission be moved into Web client, since nearly everything word wants to do is about being an HTML mail client and then breaking out IntelliSense/Assistance/UI into a single focus area. This sounds good to me.

Note 2: As we talked about a primary goal of this is to reduce the number of people that need to meet about the whole product. So there is still a desire to reduce the total number of these groups (currently at 9). One thought is to put programmability in with one of the web groups (client or server, depending on Note 1).

Note 3: Office Shell is something that severa have mentioned (brianmac, richardm, craigu). This is something that we should understand better and factor in appropriately.