

From: Jeff J. Johnson
Sent: Thursday, September 14, 2000 10:24 PM
To: Steven Sinofsky; Antoine Leblond; Heikki Kanerva; Grant George; Dave Buchthal; Mike Kelly (Office); Beverly Sobelman
Cc: Jeff J. Johnson
Subject: NGO thoughts

I've been thinking a fair bit about NGO and how it relates to development over the past few months. Based on my musings, offsites, and hallway talks, I've written up the attached memo. Some of the ideas will be obvious, some controversial, others may be stupid or turn out to be dead wrong. My goal isn't to get things set in stone right now, but to get people thinking and discussing these issues.

(Disclaimer On) The most important thing right now is shipping Office 10 on 3/2/1 and any activities related to NGO planning are secondary (Disclaimer Off).

I'd appreciate it if you could give the memo a read and let me know what you think.

Thanks,

--jjj



NGO from a
Development Perspec...

Plaintiff's Exhibit

6840_A

Comes v. Microsoft

**MS/CR 0017695
CONFIDENTIAL**

NGO from a Development Perspective

NGO Meta-goal

I hear just about as many different goals for NGO as people I talk to. One of the first tasks we have is to get everybody on the same page. The chain of reasoning I have built for NGO is this: The (blunt and crass) goal of any business is to make as much money as fast as possible for as long as possible. Office has been more successful at doing this than just about any other franchise in history. However, we now believe that a number of factors preclude us from continuing to sell Office as we have in the past. Further, we have decided to take a substantial risk by betting on the subscription model. Thus, the number one goal of NGO is to build a subscription product that will attract as many users as possible and induce them to become long term, paying customers.

Note that the primary goal says nothing about being a service or building eServices. Doubtless, many of the things we do for NGO and subsequent releases will in fact be eServices or other web related offerings. But that isn't because our mission is to provide eServices, it will be because they offer the best value proposition for building something that will entice a large number of customers to enter into a subscription relationship with us. If people believe we will never do another large-scale, traditional feature (like Escher, PivotTables, or Command Bars) they are dead wrong. We will do whatever features are most likely to attract the long term subscription customer.

NGO Product Cycle Duration

It's been stated many times that we want to deliver NGO in a 12 month development cycle. It is clear that without incredible focus, the right organization, unwavering discipline, and a bit of luck that this will be difficult at best and impossible at worst. We need to honestly assess the reasons behind the 12 month goal and make sure they outweigh the negatives. Clearly sooner is better, all other things being equal. Here is a list of challenges that will make a short cycle difficult.

- We haven't demonstrated the ability to ship in under 18 months for the past three cycles. Over that time the complexity of Office has increased at a non-linear rate.
- Shortening a cycle by 1/3 doesn't shrink all parts of the cycle proportionately. Specifically, the stabilization and testing tail doesn't shrink as fast. We had 14 weeks of pure coding in O10, a third less leaves us with about 9 weeks. The actual amount of coding will probably be only about 7 – 8 weeks once the tail is factored in. Not sure how much compelling value we can create in this amount of time.
- We will have to greatly focus where we write code to reduce the testing impact and explicitly avoid certain types of changes altogether. If NGO were just about delivering 3 prototype services this would be rather straightforward to do. However, I see NGO as the project where we make one last trek into the bowels of the code to modularize it and optimize it for delivery via subscription and make it web-service aware. This will touch a bunch of code.
- We know how to do the 18 month release (or we at least think we do). There will be virtually no margin for error in a 12 month cycle and the layout of the schedule will involve more than the usual amount of educated guessing.

- Spec readiness and PM vision has declined over the past two releases. With a short 7-8 weeks of coding, we absolutely will have to have every spec completed and inspected before we begin. We won't have much time to change our vision mid-stream and reengineer features.

Desirability of NGO

Customers will not buy NGO because of the plumbing and infrastructure work we do. I see three reasons customers will find value in NGO.

1. Must have eServices or features not available anywhere else. These don't necessarily have to take advantage of any NGO work we do (though if none of them do, then why change development models?), they just need to exude tons of appeal.
2. Total customer care package. We provide a no brainer system that is self maintaining, always up to date, rarely crashes, connects the user to support easily when necessary and makes their productivity software bullet proof.
3. Provide a critical basket of web-centric services. These include an Exchange server in the sky, a web drive, roaming settings, accessibility for data across multiple devices and form factors, team collaboration, etc.

Note that we can do #1 and #3 without disturbing our core code that much. What we can't do is #2 – bake into Office an awareness of the subscription model and enable the very short dev cycles for NGO+1.

I believe that it is possible that the initial desirability of NGO will not be extremely high. If we find several killer services and execute well we can do well initially. However, I believe it would be a mistake to assume this will be the case. If we do not hit several home runs in our first set of services, it will be critical that we can release new services and features at least every three months. Any longer than that and I feel we will severely impede our ability to sell NGO as a product that continually improves.

NGO+1 cycles

While it will be critical to have an architecture that is flexible enough to allow new functionality to be delivered quarterly, it is clear that we will still invent features and services whose time to implement will range from several days to more than a year. We will need to establish a methodology that allows us to undertake these features in order of priority regardless of the time to market.

One approach is to plan quarterly releases. A month before the release, all feature teams whose code meets the quality bar goes into final debug and ship mode. Those who slip or are on longer schedules continue to code. This means that complex dependencies are limited to the feature that needs them and do not hold up the entire release.

There will also be features that require core code changes that will require a more traditional 18 month cycle. All such features should be on the same schedule. Since a

traditional cycle needs at least a 6 month tail, we would allow this team to integrate into the main tree right after a quarterly release occurred, cancel the subsequent quarterly release, and make the following release the major core code release. This would require a set of short cycle features to be shelved until the next quarter.

Cycle Length vs. Quality

No matter how you slice it, shorter cycles decrease quality (or at least increase the risk of bad bugs being shipped). If we expect to ship in 12 months (let alone ship new features every three months) we will have to have plumbing in place that lets us detect bad bugs quickly (extend DW) and push a fix (or at least a stop-gap measure) very quickly. We can afford to rely less on comprehensive testing and more on focused testing if we know we have a robust and resilient platform which can be rev'ed easily and quickly, without a ton of complexity when the need arises.

Quantity of quality eServices

Based on our experience with eServices in Office 10, I believe we may have some difficulty finding and delivering quality eServices in the quantity desired. When we started looking for eServices in Office 10, it seemed a foregone conclusion that we would find a ton of them, both shared and app specific. Instead, it turned out to be rather difficult to get more than two dozen, which quickly got whittled down to 10.

Of course, the scope of NGO is longer and larger, so there is a wider field to search, but we need to be prepared in case the list of killer services is smaller than we would like. Further, it's crucial that we identify these services and PM specs them very early in the planning process to prevent us from investing dev resources in the wrong places.

NGO Organization

It's pretty clear that there are four main things that we'll be investing in. The trick is figuring out the percentages and the proper org to support the vision.

- 1) Traditional feature work – finishing rough features, focused app work, etc. The more people I talk with, the more I see that every team has a list of stuff that is begging to be finished up in NGO. That said 2/3 of every list should be punted swiftly, but the last 1/3 deserves closer inspection. Try as we might, I don't think we can entirely avoid making an investment in this area.
- 2) NGO infrastructure plumbing – shared feature team work for core services, utility routines, making Office modular, and implementing the changes necessary for the subscription service delivery mechanism.
- 3) Client-side service feature work – wiring up the service in the apps and other non-server feature work.
- 4) Server-side eServices – behind the scenes plumbing, web pages, reporting, auditing, and other back end work.

My feel is that we will need to spend 15%-20% of our total resources doing traditional feature work, 50%-65% doing the infrastructure plumbing, with the remainder going to the exemplar eServices (client and server). At first glance, it may seem that my percentage for shared plumbing is high; however, when you begin to enumerate the

architectural changes required for the subscription model, the minimal set of work mounts rather quickly.

One way to organize would be to form one team responsible for #1, another for #2 and a third for #3 and #4. There is a great reluctance among a number of developers to move groups, so we need to incent people to learn new skills and move. If a single org owns both #1 and #2, there will be more temptation to shift resources towards #1, especially if there isn't much movement between dev teams.

The eServices work could possibly start later than the plumbing work (many of those devs could do some plumbing) since the time to market of a service is frequently much shorter than the plumbing work. Of course, we need to ensure that at least one service is developed soon enough to allow the plumbing to have something to integrate against.

NGO Statement of Success

It is important that we determine early on what our ultimate measure of success will be. How much time do we have to make a subscription based service a success? 18 months? 2 years? 3 years? Is it okay if the initial release of NGO isn't a top seller? Should we plan on the first version not being compelling and primarily being plumbing? What are the intermediate goals? These questions will ultimately shape the effort we mount.