EXHIBIT 56

UNREDACTED VERSION OF DOCUMENT SOUGHT TO BE LODGED UNDER SEAL

From: Mike Vernal </O=THEFACEBOOK/OU=FIRST ADMINISTRATIVE
GROUP/CN=RECIPIENTS/CN=MVERNAL>
Sent: Tuesday, January 15, 2013 9:34 AM
To: Douglas Purdy; Justin Osofsky
Cc: Sam Lessin; Sean Ryan; Dan Rose; David Swain; Jennifer Taylor
Subject: Re: Addressing app quality

Note, I'm 110% supportive of us being really proactive and holding a really high bar for app review for both App Center and Open Graph, and working to make sure everyone we allow into those two channels are really high quality and clearly adding value to the ecosystem.

My feedback below is about going back to folks that have already launched and creating thrash. E.g., if we make Canvas "Games only," we're going to either kill Causes or make an exception, they're going to escalate to Mark, and we're going to spend 30-60 minutes talking about this with Mark instead of talking about how to maximize data acquisition or maximize revenue acquisition.

-mike

From: Mike Vernal <vernal@fb.com>
Date: Tuesday, January 15, 2013 9:29 AM
To: Douglas Purdy <dmp@fb.com>, Justin Osofsky <josofsky@fb.com>
Cc: Sam Lessin <sl@fb.com>, Sean Ryan <seandryan@fb.com>, Dan Rose <drose@fb.com>, David Swain <dswain@fb.com>, Jennifer Taylor <jht@fb.com>
Subject: Re: Addressing app quality

I agree this is probably a better group conversation, even though it might be less efficient.

My high-level net is that we need to be spending the vast majority of our time -- cross-functionality -- on pushing core metrics (acquiring data via Open Graph or driving revenue from games or mobile ads).

On the one hand, I definitely want to clean-up a lot of the issues called out below. On the other hand, I think we have a (negative) history of spending a lot of time "cleaning stuff up for the future" under the theory that it saves us work down the line, but I think what it actually does is create a bunch of work in the present that distracts us from our core goals. I think Mark's comfort with us not implementing Platform 3.0 in H1 is because he'd much rather us focus on moving core metrics that create value than worrying about being defensive / in clean-up mode.

So we should discuss, but my overall nets are:

- The vast majority of our time (80+%) should be spent on getting data or money from developers

- To the extent that other developers are legitimately distracting our ability to have this focus, we should fix that (I think they create little distraction for the product/eng team, but it might be higher for partnerships and much higher for ops)

- On the flip side, changing rules, deprecating stuff, etc. has very real cost -- partnership cost, ecosystem cost, PR cost, eng cost, etc. That cost also tends to be very thrashy (things will get escalated to Mark, and we'll end up spending our time with Mark talking about low-leverage things in the past rather than high-leverage things for the future.

Where we have issues with our platform that are legitimately distracting Facebook employees from being able to focus on the future we should probably fix them, but our overall intellectual energy should be focused on figuring out how to optimize for the future rather than cleaning-up the past.

-mike

From: Douglas Purdy <dmp@fb.com>
Date: Tuesday, January 15, 2013 8:50 AM
To: Justin Osofsky <josofsky@fb.com>
Cc: Sam Lessin <sl@fb.com>, Mike Vernal <vernal@fb.com>, Sean Ryan <seandryan@fb.com>, Dan Rose <drose@fb.com>, David Swain <dswain@fb.com>, Jennifer Taylor <jht@fb.com>
Subject: Re: Addressing app quality

As much as I hate large meetings, I think we need to have an interactive dialogue with the key folks in the room.

We have a meeting tomorrow to discuss a bunch of ecosystem issues, perhaps we can do it there.

I personally want to get the duplicative policy in place now and use it as the mechanism to implement harder reviews.

BTW: I met with Singly yesterday. I don't think we are all calibrated on them as they are decidedly in the helping us rather than hurting us camp.

On Jan 15, 2013, at 8:25 AM, "Justin Osofsky" <josofsky@fb.com> wrote:

> Sam, Sean, and I had a good discussion about app quality yesterday. While we've begun to address the most significant competitive issues (e.g., Wechat), we're making less progress on (1) apps that reproduce core functionality (e.g., Mycalendar) and (2) apps that read lots of data with little reciprocal value exchange.
>
> I think the challenge with these latter categories is that no individual case is compelling enough to address. However, collectively, these apps present a real cost. They're misaligned with our emerging overall direction, send the wrong signal to the next generation of devs and investors, and are a drain on our xfn team as we deal with them as one-offs.
>
> One of my top goals for the Ops team in 1H '13 is to solve this issue. I don't want us to find ourselves in mid-2013 in the same place we are today, and am concerned that we will be unless we address this head on.
>
> In the next few days, I will set up a quick 1:1 with each of you to get your input and align on a clear recommendation of how to move forward. I'm more focused on a pragmatic approach that pushes us in a direction that we all know feels right (e.g., eliminate the category of Mycalendar apps and apps which leach FB data) rather than developing a perfect philosophical framework. There are a few options (which are not mutually exclusive):
>
> - Policy. Update our policy to (1) restrict apps from reproducing core functionality and (2) emphasize reciprocity. We would then enforce against the Singly and Mycalendar app categories. Alternatively, we could more rigorously enforce the threshold of when devs need to have a deal with us. However, if we go the policy route, I'd prefer to explicitly signal our intentions than enforce an ambiguous policy.
> - Ops reviews. Implement more stringent ops reviews and prevent apps from launching in categories where we're not comfortable (e.g., reproduce core functionality). Essentially, this would make our app review process look more like Apple.
> - Remove friend APIs. This is the most scalable solution, but likely requires a meaningful product give (e.g., better friend invite flows) to make it palatable to developers.
> - Distribution channels. Restrict the distribution channels which enable spammy apps to grow.

- Do nothing. I think that the status quo is worse than any of the above options. These issues will only get more painful as the next set of devs raises VC to build on our platform, and we'll continue to spend internal cycles.