

Outlook E-mail

From: [REDACTED]
Sent: 4/3/2012 1:27:33 AM
To: [REDACTED] Mike Schroepfer
Cc: [REDACTED] Chris Cox; Mark Zuckerberg
Subject: Re:

I agree with what's been said here (especially copying is faster than innovating). Recently, I've been thinking about why we haven't moved faster on Roger and Snap, and as a part of the Apps team, I'm increasingly concerned as I watch startups siphon our graph and create awesome new experiences faster than we can.

I have two additional observations on what slows us down. To be clear, I'm not saying we should change our ways, but some things to be aware of.

1. When it comes to UI, we don't believe "done is better than perfect." We should be aware of the price of iteration.

The WhatsApp UI sucks, but they took the hit on UI to focus on reliable sending and fast growth. Those are arguably the features that win over users, not UI. We had an ugly initial UI for Roger ("Read" on each message), but we iterated on it for the last 3 weeks and spent ~6 eng man weeks on animation alone. [REDACTED] wrote an animation library for Android, which doesn't exist natively on Android. In that time, [REDACTED] and [REDACTED] could have hacked out a faster titan-bypass backend ([REDACTED] built the MQTT server in roughly the same time as it took him to build the animation library).

I'm glad we iterated on the Roger UI, but it had costs. The solution here is to hire more good frontend engineers so we can prototype faster. Even then, we should be good about recognizing when UI iteration is worth it; when building the product, it's easy for the team to get caught up on making it great, and the reality is that "this just adds another week" adds up to significant delays.

2. We try to fit everything into our system (design conventions, future proofing, friend graph, likes, etc...), and it slows us down sometimes.

The vast majority of the time, our site provides a strong platform for growth. We are building a business for the long-term, and it's important to build value for our whole ecosystem.

However, the downside is that we spend a lot of time making sure our designs fit conventions or settings are future proofed. Startups feel free to make small tweaks whenever, so they can afford to try things faster without fear of being tied to them. I've noticed this being something that has slowed us down on Roger and other projects.

Another consequence is that we have to deal with legacy settings or expectations, and even branding poses a concern (e.g. people don't think of FB messages as mobile messages) or the concept of friends (e.g. We immediately thought of messaging friends as the primary use case with Messenger, but mobile messages requires communicating with anyone).

Startups have the best of both worlds: they siphon our graph to build a new system with a new set of norms that better suits their use case. Instagram and Pinterest are great examples of this; they seed with friends, but their fundamental model is different (follow, board, pins, etc...), and it lets them create a different product experience, one that adapts better to needs.

On this front, I don't think we should change much. There is so much value in building a larger ecosystem. However, more and more, I wonder what it

would be like to invest heavily into a different app -- more heavily than Messenger -- where we actually have a different set of design aesthetics and different norms. I wonder if that would make a difference in iteration speed.

On 4/2/12 11:26 AM, [REDACTED] wrote:

>I also think that fast-following requires pretty tight / specific project
>goals so everyone is aligned on why we are fast following and what we are
>trying to achieve in doing it.

>
>
>
>
>
>
>

>On 4/2/12 10:21 AM, "Mike Schroepfer" wrote:

>>Fast-follow is something I'm a fan of.

>>
>>We have a mixed history of these things, however. The key challenge is
>>to make sure the engineers/pms on the project are excited/motivated by
>>what they are doing. We also need to make sure it is close enough into
>>our core that it makes sense to do. I'm thinking specifically of a few
>>projects: a) questions b) friend lists/subscribe c) nile d) stream search
>>e) location/check-ins. Some of these were direct responses to
>>competitive threats and their execution varied dramatically.

>>
>>In terms of the broader question on "how to move faster" - I think there
>>are only a couple of big levers to pull:

>>
>>a) Reduce design/product churn - fast following is a good way to do this
>>- but we can also do this by becoming more iterative in general
>>b) More product development resources (pm/eng/design)
>>c) Better core infrastructure (e.g. the GraphQL stuff and equivalent
>>should make it quicker to build products)
>>d) Even less coupling in our apps - e.g. the extreme is teams building
>>platform apps. Zynga, for example, shares some core code but otherwise
>>operates effectively as independent studios
>>e) Relaxing any "strategic/org taxes" - e.g. growth, revenue, si, legal,
>>etc. all subtly take their toll
>>f) Cancel a big project like Firefly

>>
>>Schrep

>>
>>On Apr 1, 2012, at 4:11 PM, [REDACTED] wrote:

>>> I personally am pretty in favor of this approach in life generally... I
>>> would love to be far more aggressive and nimble in copying competitors
>>>at
>>> the interface / last mile level -- and far more willing to launch early
>>> and then cut or double-down on an ongoing basis.

>>>
>>> The only caveat in my mind is being really deliberate about the layers
>>> where we are being nimble & fast vs. the lower-down frameworks that
>>>should

>>> move more slowly and deliberately. The way I think about things,
>>>consider

>>> FB a giant fly-wheel.... You want the external edge to move very quickly
>>> and iteratively -- and small dings when you get things wrong don't cost
>>> much, but the deeper you go in the stack / towards the infrastructure
>>>core

>>> (and really the human capital core even below that), you want things to
>>> move more slowly and predictably.

>>>
>>> Practically speaking, I think this just means that we all have to be
>>> dealing in the same product abstractions, following a consistent

>>>privacy
>>> model, etc. I absolutely think that we can accomplish this balance.
>>>
>>> Let's 'copy' (aka super-set) pinterest!
>>> [REDACTED]
>>>
>>>
>>>
>>>
>>>
>>> On 3/31/12 7:36 PM, [REDACTED] wrote:
>>>
>>>> Hey Mark - just some quick thoughts on this:
>>>>
>>>> Copying is *definitely* much faster. When I was 'copying' FB before
>>>> joining back in 2007 I managed to launch a social networking site in
>>>> Spanish with feature set parity to FB with a small team of 3-5 guys in
>>>> India. If you think of all the A/B tests and hundreds of iterations
>>>>and
>>>> optimizations that we run on a flow like NUX / Friend Browser / etc...
>>>>in
>>>> order to come up with that version of the flow that performs best, we
>>>>end
>>>> up spending a lot of engineering / design / analytics cycles that the
>>>> guys cloning don't need to spend. There is a lot of IP on the UI that
>>>> unfortunately is not defensible, and the guy copying can free ride on.
>>>>
>>>> The guys copying also don't need to spend time launching products that
>>>> they saw didn't work that well. They just copy whatever they see works
>>>>/
>>>> can make money and iterate a bit on it (which is much faster than
>>>>trial &
>>>> error). We spend a lot of time on products and iterations on products
>>>> that are not that used, or not anymore around- e.g. marketplace, Q&A,
>>>> facebook lite, calendar, etc...
>>>>
>>>> If you gave the top down order to go ahead, copy e.g. pinterest or the
>>>> gaming dynamics on foursquare, or feature x on product y as is (with a
>>>> couple of minor tweaks to make it fit better on FB), I am sure very
>>>>small
>>>> team of engineers a PM and a designer would get it done super quickly.
>>>> There would be costs on this approach in terms of how we are perceived
>>>>in
>>>> the industry (copying vs. innovating), our platform strategy (we would
>>>> scare developers), etc... but the approach is not necessarily a failed
>>>> business technique (Zynga built a multibillion \$ company doing
>>>>this...)
>>>>
>>>>
>>>> -----Original Message-----
>>>> From: [REDACTED] markzuckerberg
>>>> Sent: Friday, March 30, 2012 12:55 AM
>>>> To: Chris Cox; [REDACTED] Mike Schroepfer; [REDACTED]
>>>> Subject:
>>>>
>>>> I spent some time with the Renren founders today and Robin Li from
>>>>Baidu
>>>> earlier in the week and wanted to pass along a few things:
>>>>
>>>> In China there is this strong culture of cloning things quickly and
>>>> building lots of different products instead of just focusing on one
>>>>thing
>>>> at a time. This allows them to plant lots of seeds, and although it
>>>> yields lower quality products in the short term as they're cloning and
>>>> the markets are growing quickly, as markets mature there seems to be
>>>>less
>>>> of a gap between the clones and the originals.

>>>>
>>>> As an example, the Renren site now seems almost as clean and polished
>>>>as
>>>> our own, despite being more of a mess in the past. They also have more
>>>> features than us, including:
>>>>
>>>> They have built their own version of Pinterest. In addition to your
>>>>own
>>>> timeline, you can have board pages that people can follow. They are
>>>> tightly integrated into their NF. This is similar to what we've talked
>>>> about doing in the past, and they have a version of this that seems to
>>>> work well.
>>>>
>>>> They have built their own version of Tumblr. Same deal as with their
>>>> version of Pinterest. Both seem relatively well executed to me and are
>>>> two of their fastest growing properties.
>>>>
>>>> They have built their own standalone messenger app, where one key
>>>>feature
>>>> is using it like a walkie talkie -- basically like Voxer. Apparently
>>>> Tencent QQ has also released a Voxer-like app which is really blowing
>>>>up
>>>> in China. Robin Li confirmed that a lot of people are using it,
>>>>although
>>>> he wasn't sure if that was because it's relatively harder to type in
>>>> Chinese or if it's a universal need. Still, interesting that it's
>>>>taking
>>>> off here.
>>>>
>>>> Renren has also built their own games and they have 6 of the top 10
>>>> Chinese games in the iOS app store.
>>>>
>>>> They have also built out a full music product where they have licensed
>>>> all the music in China themselves.
>>>>
>>>> They have done some innovative things with commenting, like enabling
>>>> people to easily fork comment threads to turn them into separate
>>>>threads
>>>> if they want. They also have emoji integrated into basically every
>>>>text
>>>> input field.
>>>>
>>>> They also just implemented a version of divebar with a nice touch
>>>>where
>>>> whenever you hover over a person's name, you get a little home icon
>>>>which
>>>> takes you to their timeline. We should implement this as well,
>>>>although
>>>> we'll probably want to use a different icon than home. (Linking from
>>>> messaging to people's identities has been an issue for a while, so we
>>>> should also fix this in the Messenger app, in the title bars of each
>>>>chat
>>>> window, etc.)
>>>>
>>>> Overall, seeing all this and the pace that new mobile apps seem to be
>>>> coming out from other companies makes me think we're moving very
>>>>slowly.
>>>> If we were moving faster, then we might be able to build out more of
>>>>the
>>>> social use cases ourselves and prevent our competitors from getting
>>>> footholds.
>>>>
>>>> Maybe it's just a lot easier for these guys to move quickly since
>>>>they're
>>>> just copying other people, but a lot of the stuff that we're doing
>>>>around
>>>> messages, photos, etc doesn't have a huge number of original problems
>>>> either.

>>>>
>>>> I wonder what we could do to move a lot faster.
>>>>
>>>> Sent from 中国
>>>
>>
>