From: Steven Sinofsky <[redacted]>
To: Jeffrey Epstein <jeevacation@gmail.com>
Subject: Fw: [New post] "One Strategy" — organization structure and transition (book/blog post reprint)
Date: Sun, 14 Jul 2013 14:14:35 +0000
Importance: Normal

I did this blog post--a little more subtle...

Sent from Windows Mail

From: Learning by Shipping
Sent: Saturday, July 13, 2013 6:59 PM
To: [redacted]

Respond to this post by replying above this line

New post on **Learning by Shipping**

"One Strategy" — organization structure and transition (book/blog post reprint)

by Steven Sinofsky

A recent post by Ben Thompson touched on some of the issues in going from a product focused organization to a discipline focused organization. And while I don't agree with the broad (and binary) view of orgs, there are several interesting things in general in the post which are worth discussing. Many of these issues are not about a big (or specific) company at all and I've seen the same things in startups with 5 people or 100 people. Organizations of all sizes have "seams" or "boundaries" -- while one is product, others include disciplines (engineering, marketing, sales), code architecture or subsystems, geographies, target customers, external partners, revenue sources, budgets, ship dates, product cycle lengths, and more all represent places that multiple groups of people might arrive at different optimizations, choices, or decisions while representing ways in which groups of people can be organized.

This topic was on discussed at length in the book, *One Strategy: Organization, Planning, and Decision Making* (hurry "only 17 copies left"), authored by myself and Marco Iansiti we tackled some of the challenges faced in transitioning the Windows team to a new way of working and organization structure during the 2006-2007 timeframe. The "reprinted" post below is representative of the approach to organization and the transition the team undertook. I think it contributes to the overall dialog of the complexities and challenges one faces in organizing product development teams.

The book features a collection of blog posts from a blog I authored internally along with a broader view provided by Marco. The book focuses a great deal on our journey as a team to (according to the jacket copy):

> *Aligning a complex organization around one strategy requires all members of a team to participate—learning, sharing, communicating, and contributing to the team's success. One Strategy provides a unique combination of real-world experience managing a large-scale organization with academic research in strategy and innovation to describe what it takes to align an organization around one strategy, manage its execution, and reach for "strategic integrity."*

Keep in mind that context plays a critical part in looking at how all these variables come together. Organizing a team is doing your best with imperfect information to optimize a seemingly infinite set of variables. Product development and business are both social sciences. In fact in reading this post, you can see the context change as we introduce the book and posts noting just some of the challenges such as moving to bring a regular release cycle to Windows and cut the release time by half, to scale software as a service, or to bring Internet Explorer to current web standards. Context matters as we say.

This post below is from page 25 of our book and talks about the transition the organization was making from a team composed of a large number of "product units" to a "functional" or "discipline" focused organization.

In this jargon, a "product unit" is a team consisting of the engineering disciplines and often related marketing (or at least dedicated marketing) with a general manager. A function or discipline means job functions such as development, testing, program (product) management, and more.

3/20/2007 12 months and about 120,000 words later...

Friday marks the first 12 months of our new Windows and Windows Live organization (and also the 100th post to this blog) so it seems like a good idea to take a step back and look at the progress we have made together and think about the work left to be done.

Before getting started, I wanted to thank everyone on the team and all of those that have been supportive of the team and the work we have been doing for the past year. Together we have embarked on a great deal of change and put in place many new operational methods, and none of that would have been possible without the open-minds and cooperative efforts of those involved. Thank you!

I also want to use this as a chance to reinforce the “open inbox” that I (and all the leaders in WWL) maintain. Some of the best input and best conversations I’ve had have been with people who just send mail or stop me in the halls. It is one thing to see averages and deltas from the mean on an employee survey, but the real improvements in our team come from written comments or conversations. This substance is what really has helped to make a big difference in how our team is evolving.

We have had a year that at times has seemed rather focused on process and operations, which is in a sense by definition, but also at times a bit less than ideal. Last spring we were still months from finishing Vista and our first wave of Live services would start to slip months beyond their originally planned dates. Yet we came through that and through the midst of significant changes in organization and structure we released Vista, a plethora of Live services, a major release of Live Search, and Internet Explorer 7. To a person

everyone contributed to delivering an amazing amount of software. At the same time there was a consistently expressed desire to do things better, and so that has been the motivation for the focus on the "how" we get things done. As I said in our announcement in March we will aspire to have the best engineering organization in the company—one that is focused, with modesty, on discipline expertise, agility, creativity and above all is a learning organization. Across every dimension we have been working hard to instill a sense of clear accountability with the resources and framework to be successful. I also committed to help us achieve these goals with transparency of process and this blog has been part of this work. And of course we want to move forward with a renewed emphasis on planning our work, and working those plans.

With all of these changes it is important to point out that it takes time to see the results. Some would have hoped for faster results, and others would have hoped for less change. We are a work in progress for sure. Nothing says that more than the fact that we are just getting started with big releases of software—a new wave of Live services, a new release of Internet Explorer, a new release of Search, and of course a new release of Windows (and SP1). In the hallways, folks ask me every day "how's it going?" and I generally think it is more than rhetorical—I think people really want to know how things are progressing. Things are progressing well. I am very optimistic. Things are upbeat. I do indeed have the benefit of a broader perspective and some "altitude" which helps. I know that some of the challenges many individuals face on a daily basis—the challenges that seem to get in the way of getting work done—are still there and did not disappear overnight. I know that some new challenges have appeared. But I don't have to look far to hear feedback about things moving forward and heading in the right direction.

It is worth looking at three of the main areas of feedback that correspond to three of the biggest changes we have made as an organization and provide a "state of the organization" view of **organization efficiency**, **planning**, and **decision making**. It is hard to separate these changes because they are interrelated. We can make changes in our organization structure because we are more focused on executing on a plan. We can create a plan that reflects the whole team because we put in place more empowered decision makers. But I also recognize that for each of these areas we have much work to do, few concrete results, and for some the jury is still out. That's why I think it is good to take a step back and make sure to communicate and acknowledge that the management team does understand where we are today and also to share the sense of optimism that we each have.

> *If you want the short version of this post…we have created an organization that is flatter, has fewer managers, and is much more focused on core engineering values. With this organization in place we have embarked on a product planning process that is about bringing the best ideas to our products in a timely and thoughtful manner. And because of our structure and because we are operating with a plan we are in a much better position to make and stick to decisions. Yet this is a work in progress and I recognize that for many the changes have not been enough or fast enough, or that the changes have not yet erased the challenges in daily work. It takes time for these changes to pay off and time for them to move through the whole organization. With the evidence of early results, such as the Live and Search plans and the progress we have made as a team on Windows and IE I believe we all have reasons to be optimistic and that we will see results of our efforts as we enter execution*

> *mode on our work. Our software will be used by over a billion people by the time we finish this next wave of products—that is an awesome responsibility and an amazing opportunity for us all in our careers.*

Organization Efficiency and Engineering Focus

In many posts to this blog and in many of our team meetings we have talked a great deal about the organization's statistics such as how many people at what level, how many direct reports managers have, how big teams should be, and how many managers and management layers there should be. This is an area where it is easy to be concrete and so was the first one we acted on as an organization. Across Windows and Windows Live (and COSD, Core OS Division) we have worked hard to put in place a team that values the engineering disciplines and streamlines our engineering organization and I think we have made progress.

Last year on average an individual contributor had 7 managers between him/her and me. On average we have reduced this to 3 or 4. I think that is probably the most dramatic change. It is also one that has come in handy as we recruit from colleges this year. It is amazing how sensitive college candidates are to this detail and how often I was asked on campuses and in email exchanges with candidates—being compared to Google (an organization that has 4000 engineers these days) favorably definitely helps. We are also asking more of our managers, but not as much as some folks might have thought—we do have more reports for most managers now, but on average this is only 1 or 2 more reports (because previously many managers had only 2 reports). I want to give a lot of credit to all of our managers—they have been leading a big change even in the face of some uncertainty. And many people on the team have new managers, which is a two-way street of newness, and everyone has done a great job working through those transitions. I've had some conversations with folks where they ask if we are moving to a model where if you are a manager then you are a "pure" manager and only manage people—nothing could be further from the truth and what we have tried to emphasize is the desire to have managers contributing at a much deeper technical level than in the past, and this is possible because of a combination of more managers working within an engineering discipline and a much more focused planning process that frees managers from the burden of "keeping things under control" to focus more on execution of the plan.

The biggest part of our organization's change has been the focus away from silos of product units to a focus on disciplines working together. It goes without saying that this was controversial and also a change that for many amounted to resetting career goals. We have in place about 30 engineering triads of development, testing, and program management representing our 1000 or so developers. Some are still adjusting—wondering who will ultimately decide (they will) or what happens when dev/test/pm don't agree (they will work it out, based on consensus), and so on. Until we go through the whole ship cycle, or at least the full planning cycle, it will be the case that some are still waiting for a VP to swoop in and take away the empowerment of the triad (it won't happen :-)). It is super clear that most people understand the increased level of responsibility that they feel. I've had incredible discussions with GPMs who own planning themes or big bets and they are definitely seeing the reality that the buck stops with them—without their plans we simply won't have features in that area. I've had conversations with SDETs (software design engineers in test) who talk about what it is like to be part of the planning process for the very first time. And in talking to developers who for the first time are taking a step back and looking at the code assets during MQ (milestone for quality) and thinking about

what we should do. All of this is part of taking a discipline-specific view of the work that needs to happen. We are focused on mastering the crafts of dev, test, and pm rather than everyone gravitating to more generalist roles.

I've also talked to some members of the team who think we replaced one type of inefficiency with another—that is we replaced silos of product units with cross-team meetings. On the one hand there is some truth to this, but that comes from the fact that there is no org structure where we develop our products unencumbered by the relationships to other products. On the other hand, we are adjusting to operating in a way that optimizes for executing within a planned framework. We are still building trust across the teams and still learning how to work together. We were used to a manager clearing the way, usually by isolating the team or forging arm's length relationships with other teams, rather than a focus on collaboration and consensus. Some folks have said it feels like we went from 1 boss to 3. My view is that we are now much more accurately reflecting the specializations and skills necessary to create world class software, but I recognize that this puts a high burden on managers staying in sync with their discipline peers and communicating. This is no different than the expectation in a tiny team of one dev (manager), one pm (manager), one test (manager) – except we are trying to scale this to a larger group. It might seem harder now, especially because we only see the work and not the end-result. I've personally experienced this transition before so perhaps that is why I have confidence that once we go through the product cycle it will be much clearer and we will all be surprised at how much more natural collaboration happens.

We are still adjusting to having to incorporate many different perspectives and inputs from our engineering team at each step in the product cycle. We are seeing test involved early on for the first time. We are seeing program management working to get out front while also accountable to development for detailed specifications. It is clear that every discipline is reaching new levels of discipline involvement and if you think about it in terms of the CSPs (Career Stage Profiles) we are seeing more people touch more of those areas than we have had previously. I think this is a good sign for the growth of individuals on our team and our longer term health as an organization.

We start the second year of our organization with a streamlined organization that is much more focused on engineering and collaboration. I know it will take time for every member of the team to experience the positives of these changes and it is easy to focus on the change and see the less than positive elements. It might even be the case that over time some will find this new level of accountability and empowerment to be too challenging. Of all the changes to the team this is the one that will have the biggest long-term benefits for the careers of engineers. If I could pick one thing that we still need to improve it is the communication throughout the team and across the disciplines. We are still not sending around (locally) enough meeting notes and not sharing information more freely. And part of that is asking people to be more receptive to "raw data" and less demanding of "tell me what is important" because with empowerment comes the need to process more data and manage the flow of information. For our process to work smoothly we do need more communication.

Planning

About six weeks after I moved into this new role I remember reading that when asked about the next release of Windows Steve Ballmer said "that will never happen again...it'll never happen again." If I wasn't awake and alert, I definitely was after reading that.

Of course I know that my commitments (as published) were very much focused on planning. The option open to me was to simply put in place the best process we could and hope we could all follow it, without trying to make it seem like it was a “playbook”. That was a big challenge for me. I know many folks thought that I pulled the planning process out of my back pocket, and certainly I had my fair share of stories about Office, but the truth is that the process we embarked on tried to take into account the unique situations of each of our teams. The Live teams wanted to remain as agile as they had been. The Windows team needed to finish Vista. The Search team had monthly improvements and a well-known set of features to work on. Yet it was clear that the team at large was hoping for more of a definitive sense of “what will we accomplish” and “what does success look like”.

A big turning point and in a sense our first real team event was when Marco Iansiti from Harvard came and taught us about the benefits of planning from the perspective of two case studies. While we did not walk away from this day with the magic answers or the specific plans, we did develop a shared sense of the importance of planning and the value that thoughtful planning can bring to product development. Even though that was back in October, people still comment about “not in the direction of goodness” and “we need to be building more keels” in reference to the Challenger case and the Team NZ case. I learned a good lesson about team building in how having those shared experiences, even if not entirely applicable to day-to-day work, is very valuable.

These discussions led to the “WWL Planning Process” slide that we have used dozens of times in offsites and meetings. In a sense this slide made the planning process seem more rote and mechanical than it is in real life. Early on many worried (in email to me) that the process would not yield any creativity and that it squeezes the creativity out of product development. Many worried about the trust we were putting in the teams to develop the plan and had expectations of executives swooping in. And of course the biggest worry people expressed was that it all was going to take too long. We’re behind and need to get ahead so how can we afford the downtime to plan was a common refrain.

The first team to move from planning to plan was the Live Experience team. We made a deliberate choice to focus on getting the plan in place as the highest team priority (perhaps in contrast to the Windows team where we have been focusing more on developing the operational aspects of the larger team and COSD, Core OS Division). In a few short weeks it became clear that we did indeed unleash a lot of creativity. Our first lesson as a team was that the hardest part of the “WWL Planning Process” is figuring out what ideas to decide to do. It was really great to see the prototypes and the detailed feature lists. This creativity culminated in the Vision rollout (jointly presented by the WLP (Windows Live Platform) and WLX (Windows Live Experience) teams) that showed clear commitments to specific scenarios. We have tons of work to do and like any product cycle we will go through milestone checkpoints, recalibration, and so on. But we did learn together than putting the process in place did not constrain our creativity.

The core of the planning process is the idea that good ideas can come from anywhere and that the role of those leading the planning themes is to make sure they go out there and ferret out those ideas, and those with ideas need to seek out those working on the plans. The idea of “middles-out” planning is definitely new. JulieLar and ChrisJo and I have talked many times about how those on the team and others all think we have “the plan” in our Drafts folder waiting to be sent out, and sometimes there are those on the team that are waiting for us to hit send regardless of what others think up. We are deep in the

process on the Windows product now and it is definitely the case that people are feeling the sense of accountability for the plans. One GPM who used to be a PUM told me that they feel way more responsible for the plan of their team then they ever have before. They know that the plan will fit within the vision and that they alone are responsible for that, and previously they felt that the "worklist" would just sort of get created and before they knew it their primary job was to avoid having too much work put on them making it impossible to get done the few things they thought should get done. This is just one case, but it is a general theme. Of course I still have to earn the trust of the team and so do all of my direct reports. I was once asked if I plan on [personally] redesigning the [Windows 7] Start button 80% through the project and I took an oath not to do that in front of the WEX (Windows Experience) team, but I still have to follow through on that. Our team is learning how to consider all points of view and bring those together into a plan. It might look a bit chaotic right now and we will iterate and get better. More than most changes, the "what goes into the plan" is likely to be the one that is the most opaque while it is transpiring. If you have any questions and are not sure about what is going on, please ask rather than assume nothing or something worse is going on.

We are also seeing the benefits of "downtime" despite the misnomer. The time is not idle at all—all the teams have done (or are in the progress of doing) some excellent MQ work. Each of the leaders of development has worked hard at documenting development practices that will be used for quality, security, performance, etc. Our test discipline has done some great work on automation and frameworks. On the Windows team we continue to dive into postmortem feedback around test tools and bug tracking and using the MQ time to be concrete about our goals and exit criteria. Yet it is fair to say that some are still frustrated by our lack of "coding" right now—some believe that "we know what features we want to do so we are wasting time not coding". Of course it could very well be that we might end up doing precisely some of those features, but we also know that developing these features with up front thinking in terms of specifications and test plans will at the very least make delivering those features more reliable, if not make those features better. I know I have to ask for patience for some.

The most challenging area for me in terms of planning has been the expectations of the end result, the plan. By talking so much about the plan, I believe I have inadvertently raised the stakes of the plan (the vision document) too high. I think many are expecting the plan to be the most amazing document with detailed killer features that describe how we deliver a knock-out blow to the competition. The plan will not have that (sorry). I expect the plan to be detailed and to be clear in the value to customers. But the real details of what are home runs or merely solid hits are up to the team. If you could really mastermind a product plan to know you would have a hit then there would be many more successful companies (or maybe many fewer, just the ones with hit products). Perhaps this is the most counter-intuitive part of the plan. The plan is about removing risk (the schedule, the ability to get the work done, the resources, etc.) but it cannot remove the uncertainty (did we pick good features to do, will people like the features we pick, will we have expressed the features well, etc.) In the past few weeks I have definitely heard from people that they expect the plan/vision to be more "exciting" or more "killer". I've never seen a plan that looks like a winner before you start that also looked achievable. The Office 2007 UI was just one pillar of the product and if you would have asked me in 2004 if the UI would be as paradigm shifting as it has been perceived I definitely would not have said that (until beta1, that is). We will all learn together the value of the plan and also how to recognize

good plans and feel pumped to begin development even if we know that what we have accomplished is outlining the goals, if not the actual results.

The other area of planning yet to be tested, and the source of concern, has been flexibility. The reality is that when working with a plan our team will be more agile than if there was no plan in place, as counter-intuitive as that sounds. We will have a clear sense of our priorities and the costs of making changes and new decisions. We will be able to adapt to changes in the marketplace or in our progress because we will know what remains to be done. Some are still skeptical of this and so together we will learn and adapt going forward.

There is work to be done. We have to write the vision for Windows and Internet Explorer. We have to develop prototypes and make sure our release meets the business goals we have. In Windows we have had two planning checkpoints and we are zeroing in on the product vision and we can feel excitement building. Perhaps the biggest challenge ahead of us is in doing a detailed development schedule that we require and one that has very high scheduling integrity. This is going to be critical to our success and will ask the most of our disciplines—pm needs to have specifications (that can never be detailed enough), dev needs to have schedules (that can never be granular enough), and test will need to be forgiving that pm and dev are both not 100% certain. I am sure this will stretch the team and our triads in new ways, but we are all in this together.

Decision Making

No topic caused more angst in the Vista product postmortem than decision making. Often this is a code word for “random executive” or “deadlocked managers”. It also described the feeling that almost everyone has when you feel like you know the right answer but can’t get something done. Improving decision making cannot be done by putting in a decision making process (in fact many groups put those processes in place only to see those processes get overridden). Improving decision making takes organizational trust. It takes a common framework. It takes clear roles and responsibilities.

I would say we have the start of all of these in place. But this is also an area where it is easy to find many examples where things are not friction-free. We make big products. We have many competing needs. We will always have challenges in having an environment where one person has 100% control over everything they might do. I know that either frustrates or surprises people. I have been asked by many what it is like to be a “business leader” and I remind people that even at my job I have no say in tons of things that matter a lot to whether our work is successful (when thought of from the 4 [redacted] of marketing: product, price, promotion, placement I only have say in 25% of the business, at best). But we have built our team, both in terms of feature teams and in terms of engineering structure so that we can have a balance if we all operate from a shared plan. Decision making has the key ingredients of a plan and an engineering structure that reinforces roles and responsibilities. I know the jury is still out.

The first order of business for our team was putting in place the engineering-focused organization so it is super clear that decisions about engineering get made by engineers, and that those decisions can stick because they are grounded in the technical realities of our products. When it comes to the schedule, the test plans, and the architecture I know for sure that we will not be second-guessing these down the road.

The second priority is to establish a common framework for making decisions or a common set of priorities. The product vision that outlines business goals, tenets, and clear scenarios makes it a built-in prioritization tool for the product. The key here is that we agree on the priorities and on what it takes to get done. We agree at the outset that these goals, while aggressive, are achievable and that everyone has a deep understanding to begin the project and the resources necessary to complete the project in a timely manner.

Taken together we have moved from a model of “butts on the line” or the “hero” model of working. We won’t look to a single person to go above and beyond to accomplish our work. In terms of decision making this is very important because those modes of working have a way of tearing down any orderly decision making processes and replacing them with “out of my way this needs to get done” work styles that upend even the most robust plans.

And finally we are going to be clear about work that crosses the whole product we make. We are seeing this as we create the vision where we are going to prioritize the 2 or 3 “big bets” (as we have done with Live Wave 2) and making sure we share these bets and staff them accordingly. We have moved from a model where groups believe they decide their own set of features only to see that burdened by endless inbound reprioritization.

Without a doubt the biggest part of decision making that we are looking to change, and one that still is somewhat “foreign” to people is escalation. It is still the case that we have too many things where people think they need VP approval. If you want to spend $1M then that is probably VP approval, but if you want to decide on the features for your area you have the authority to do so. We also still have many decisions where people do not agree with the decision and escalate to reverse a decision—I hope folks understand that escalating a decision is not a way to reverse things. The hope is, again, that those making decisions have done the work necessary to understand points of view and take into account the full context of a decision. If you learn something new or have new data, then change your mind rather than force members of the team to drive an escalation process. The way to avoid escalation is in communicating and making sure you are taking into account all the stakeholders—freedom to make decisions is not freedom to ignore the landscape.

With our engineering focused organization we are also expecting more from our group managers with respect to decision making. They are on the hook to drive the processes of engineering and do so without pushing the process down the organization. We have minimized the “staff” functions for running our team. We can do so because we are working to make the process of following the plan less challenging by being more consistent and less confrontational. We are still early in this regard, but watching the visions come together gives me reason to believe these changes will come naturally from the work we have done to date.

This next year will be a year of execution. We are about to roll out the plan for the next wave of Search investments. We are approaching M1 completion for Live Wave 2. We will soon have a vision for Internet Explorer. And we are entering MQ for the next release of Windows. The best organizational learning happens while doing actual work and we are doing that now. So we have clearly left the realm of theoretical and meta-discussions and are turning the crank. It is exciting.

A big part of execution mode, so to speak, is that it is now up to the team to succeed. Management, process, and memos can only take us so far. Taking the development foundation we have built together and delivering products is up to every member of the team giving 100%.

This has been a challenging year for just about everyone on the team. It has also been a rewarding year. We are a team ready to execute. We are a team ready for the challenges of execution. And above all, ready to take those on together as a team.

--Steven

Steven Sinofsky | July 13, 2013 at 6:00 pm | Categories: posts | URL: http://wp.me/p31nkB-9Q

Comment See all comments

Unsubscribe or change your email settings at Manage Subscriptions.

Trouble clicking? Copy and paste this URL into your browser:
[redacted]

Thanks for flying with [redacted]