

From: Steven Sinofsky <[REDACTED]>

To: Jeffrey Epstein <[REDACTED]>

Subject: Fw: [New post] Reaching for harmony in org changes

Date: Tue, 14 May 2013 02:07:41 +0000

Importance: Normal

Sent from Surface RT

Check out [REDACTED]

From: Learning by Shipping

Sent: Monday, May 6, 2013 12:11 PM

To: [REDACTED]

Respond to this post by replying above this line

New post on **Learning by Shipping**

Reaching for harmony in org changes

by [Steven Sinofsky](#)



Reorgs are a part of an organization of any size. As business changes, development teams resize, code evolves, or products pivot, the organization can and should change as well. Given the frequency and challenges of reorgs it is worth looking a bit at the complexity, rationale and some challenges of reorganization. While the first reaction to a reorg could range from a sigh of relief to groan or worse, the most important thing is to keep calm and make sure the work continues.

Be sure to take our three question survey on reorgs after reading this post, [here](#) ([REDACTED]) and to check out survey results below from the last survey about "Meeting effectively".

A first-year MBA student I recently met took the occasion of a reorg as time to career pivot and attend business school, which motivated this post.

Reorgs (this post is about structural and management changes, not changes in staffing levels) are sometimes a popular topic in blogs where they take on a certain level of drama or mystique (for example, some blogs talk about org changes as solutions to perceived design challenges). Lacking context, some tend to see reorgs as either the solution to or the cause of a change in strategy or execution. That itself can be the source of reorg angst. In practice, a reorg should be the outcome of a strategic decision not the decision itself—reorgs don't cause change or things to happen, but are (hopefully) a better way to execute on strategic changes that have been decided upon.

Reorgs can be a natural way to make sure a team is aligned to deliver on a strategy and a tool to allocate resources effectively towards a shared product plan. When done well, reorgs go from something that happens to you to something that happens with and for you, even if things don't always feel that way for every member of the team at the start. At the same time, reorgs are enormously challenging by their very nature—organizations are never perfect and there can always be unpredictable outcomes as members of the team implement org changes.

I've been part of and executed a few "big" reorgs and always find them incredibly challenging, humbling, stressful, and much more work than is often expected. That's why I tend to view reorgs as a tool of final resort rather than a tool to routinely drive change, which was something discussed on another blog a while back (and motivated this post). Executing a reorg involves doing everything you can to "precompute" actions, reactions, and further reactions as best you can while also compensating for them in the plan.

Reorgs are complex and can be thought of from many perspectives. As blunt as they might sometimes seem, there is a great deal of subtlety and nuance to reorgs. While we're focused on product development organizations, the concepts and implications of reorgs are a pretty general topic.

Reaching for harmony is something to strive for in any organizational change.

Do keep in mind, like so many things in the social science of business, organization and reorganization context dependent—there's no right or wrong outside the context being discussed. By definition, reorgs are forward looking and so past history might not always be the best guide.

Perspective and context

Discussing a (potential) reorg can stretch many in an organization. Much like the group describing an elephant, a reorg can mean very different things to different people. A good way to think of things is to refer to a well-known description of [organization dynamics](#) that is often used in training classes: tops, middles, bottoms. We'll return to this often in this blog as it is always a good reminder of patterns and practices that one can generally (emphasis on generally) see repeated.

Bottoms are the folks that do the work. Of course this is an awful moniker, but is the one chosen in the original work (See [starters/books/the-possibilities-of-organization.html](#)). Bottoms also make up the bulk of an organization. In a typical, large, development organization (>100) you usually need fewer than 20% of the team middles and tops, which means more than 80% of your resources are bottoms. Whenever possible, you probably want to be better than that (meaning fewer managers, though one should caution a metric like this should not be abused as a scorecard goal as context matters).

Middles are the line managers in an organization. Middles are where the work and collaboration get defined, where friction is either created or eliminated in getting work done, and where information can flow freely or stop. Healthy middles are an essential part of any organization. It is why practices such as skip-level 1:1s, communication that goes broadly to the middles, and shared view of plans are all such critical tools in a product team—those are the tools of middles managing up and across a team (emphasis on

helping the middles, not the middles helping the tops, which is a common dysfunction). Middles can also be tops. For example, if you are the most senior developer in an organization and your manager is not a developer then when it comes to development stuff you are a top.

Tops are the big bosses in an organization. The top is where a certain organization function "ends". You can be the boss of product design, the boss of the test schedule, the boss of marketing, or (but not necessarily) the boss of the whole organization or company. It is worth noting that nearly all tops are also middles at some point. It just depends on the context. CEOs are middles relative to the board (and also Customers). Your VP is a middle relative to the CEO even if you don't think of him/her as a middle.

To be complete, the framework also includes Customers. Their role in will be touched on later in the post.

I would encourage folks to check out this framework and book just because it succinctly sums up many of the core challenges within an organization. While there are many insights and many specifics to teams, a key understanding is that members of a team should do far more to understand each other's context (and problems) than they do in practice during times of change—simply walk in each other's shoes. Of course this is blindingly obvious, yet terribly difficult for even the best folks on a team. For example:

- **Tops** should sometimes spend less time worrying about their big strategic views and needs and consider how their choices (based on those needs) can ripple through an organization and impact execution. Tops would do well to listen more (see this [great discussion of 1:1s](#) from Ben Horowitz) and perhaps worry less about what is on their mind.
- **Middles** might spend more time talking to other middles and sharing what they are actually doing, what are their real execution issues, and how they are really progressing. All too often middles get caught up communicating idealized situations and plans which can cause confusion, misplaced bets, or just poor choices in other parts of a team and organization. Middles might spend too much energy on describing problems rather than solutions, or even trying to account for things not going well. Middles can spend more time informing their tops about what is going on, but that also depends on tops spending time listening or asking to be informed.
- **Bottoms** might also spend more time listening or asking questions and a little less time feeling like "victims". It is easy when middles and tops are communicating poorly to assume the worst or to assume folks don't know what is going on. It is equally challenging if the communication that does take place is not taken advantage of, so more listening here can be beneficial as well.

If you think about these typical patterns (remember, this is a generalized sociology framework not a description of your team/behavior), one can see how any discussion of reorgs can quickly degrade. In fact, few things tap into the typical patterns of this behavior framework better than a reorg. Why is that?

Reorgs, by definition, are usually kicked off by the tops. So out of the gate the assumptions that go into making an org change are from a top perspective. The biggest changes in a reorg generally affect the middles since work is reassigned, people's responsibility changes, and so on. Middles have a tendency to view reorgs at the extreme of "whatever" or "oh my gosh this is really messed up" -- as a middle so much of your role

depends on context, connections, and structure changes can significantly impact execution.

For the bottoms, a reorg can appear like a bunch of people rearranging deck chairs on the *Titanic* since ultimately the organization doesn't really change all the work of individuals (much of the same code still needs to get written, tested, maintained and changing the people with that expertise seems the opposite of progress). Throughout the process, communication is less than and often later than many would like or expect.

The process of a reorganization is one where perspectives of each on the team need to come together to define the problem, scope the alternatives, and implement the solution. Absent these steps a reorg goes from a potential solution to a certain problem.

Why reorg?

There are many reasons for doing an org change. In fact, the most important first step of a reorg is to be able to articulate to those who ask why you might do a reorg.

It is often in this very first step where most reorgs hit a snag. The reason is because the tops have a set of reasons in their context about what a change is for and what it will accomplish and then quickly find out others don't share the perspective (or problems) or view it as incomplete. Yet the process often continues.

For the tops, this can be a real pain or just frustrating and worse it can bring out the worst of bottoms and middles in terms of how they dig in their heels and get defensive about the change. They begin to immediately dispense the reasons why a reorg won't work and the bottoms pick up on these and start to feel like victims. All the while the process keeps moving forward.

Reorgs are typically instituted for a pretty common set of reasons, some of which on their own can cause people to retreat to a defensive or cynical state of mind. Some common drivers include:

- **Resource efficiency.** The role of management is to effectively allocate resources and in fact is really often the only tool management has. As a product and team evolve, resource allocations that seemed perfect at one point can seem less than optimal. An organization change has the potential to allocate resources more effectively towards the problems as they are today.
- **Duplication of efforts.** In any organization of size, over time efforts will start to converge or overlap. This is especially true in technology companies. This can be at a very visible level, for example if many groups are working on basic tools for editing photos or user names. This can also be at an infrastructure level such as how many teams have people buying servers or running labs.
- **Individual bandwidth.** Sometimes teams or responsibility grow and the management of the work becomes too challenging or individuals are spread across multiple projects too frequently. Managers at any level can systematically have too many direct reports, for example. Alternatively, the product line can change or evolve over time and folks on the team find themselves context switching between somewhat unrelated projects more than actually managing. This lack of bandwidth becomes a problem for the team overall as everyone evolves to having more overhead than work.

- **Structural challenges.** Organizations evolve over time in a way that suits the time, problem space, and skills. Sometimes when you take a step back, the current state ends up being suboptimal going forward. The alignment of resources, decision making, even core roles and responsibilities are not yielding the results. More often than not, this type organizational pain is felt broadly by the team or by customers.
- **Synergy / Strategy.** The notion of increased synergy or strategic change generally drives the most challenging of org changes. Many are familiar with these challenges—the effort to move large blocks of work in sort of an architectural view. Motivation is this sort often is about “proximity” or “relationship” and has the feel of architecting a product except it is about the team that builds the product. There's a tendency to create "portfolios" of products and teams when organizing along these lines.
- **Alignment.** Alignment is slightly different than synergy/strategy in that it speaks to how the organization should be viewed moving forward. A long time ago, for example, the Office team at Microsoft shifted from building Office "apps" to building the Office "suite". Alignment also could include many mechanical elements of businesses/products like customer definition, business models, ship dates, and so on.

Even though these have the potential to sound *Dilbert-esque*, the reality is that when problems are identified that most people on a team share, then these can form the basis of not just a useful reorganization but a reorg that people want to do. Each one of these motivations (and others not listed) can serve as the basis of a successful reorg. That might not reduce the stress, uncertainty, or even dislike of a change but it does say that reorgs do not have to be *a priori* negative or random for a team.

Ultimately, changes to an organization should be rooted in getting more and better work done. Few would disagree with that. The question is really whether the team believes an org change will do that. It sounds easy enough.

Challenges

Even with the best of initial intentions, reorgs can (and often) do hit rough spots. Rarely are reorgs stopped once started (just as it is rare that products are stopped once under development). It is a good idea to have a taxonomy of why reorgs can hit snags or challenges, since it is likely they will.

The question is not how do you avoid these necessarily, but how do you identify a specific *hiccup* the reorg is going through (much like how you identify problems in product development and address them) rather than just stopping. This preparation should take on elements of chess-play as changes and reactions are mapped out and reconsidered based on feedback. Some potential challenges include:

- **Rushing.** A potential failure with any reorg is rushing. The funny thing is that the tops usually don't think they are rushing and everyone else feels things are going too fast. During a reorg process most tops think it is dragging on forever and are just in closure mode simply because tops have likely been thinking about the reorg for quite some time already and most other people have not. In practice, most people only get a short time to hear, absorb, and reflect on the potential change. Skipping a communication and feedback step or skipping deep 1:1 conversations in a consistent and thoughtful manner can make for a very tricky reorg. When people feel the changes are rushed, the process loses structural integrity.

- **Reasoning.** Failure to effectively communicate the rationale commonly plagues reorgs. Think of a reorg like any “launch” in that you want to be clear, concise, and appeal the folks with your message. If your message is not the problem your customers have then only challenges follow. The reasoning should appeal to the people who will experience the changes—the organization is what most people in a job and on a team experience day in and day out so reasoning needs to resonate with them. Reorgs announcements that leave too many questions as “exercises for the reader” might be viewed cynically and folks might believe that not enough thought has gone into the change.
- **Strategy.** Sometimes a reorg is being done in place of a strategy— “when all else fails, lets reorg” is how victims of such a reorg might characterize things. Reorgs are not a substitute for a strategic choice an organization must make. In fact, a reorg is a tool to use *after* you have made a strategic choice. Hearing objections to reorgs based on differences in strategy is a real warning sign that the first order problem has not been addressed. If the team has a strategic choice to make (less people, fewer managers, align products, etc.) then first make that choice, then decide if a reorg is needed to accomplish the choice. More often than not, clarifying and then making a strategic choice is the more difficult, but useful, way to spend energy.
- **Timing.** A complaint bottoms and middles might raise about a reorg is when it happens—“the timing isn’t right”. A complaint many tops might have with reorgs is that everyone is always telling them the timing is wrong. In practice reorgs can be like a “stand down” for a product team. For some period of time, proportional to the number of people who change managers and/or responsibility, the team will effectively stop working. Therefore no matter how urgent the rationale, the timing of a reorg needs to minimize the impact on the work. On big teams, org inefficiencies trickle on to a team throughout a product cycle (no matter how long or short) due to people coming/going or even things like acquisitions. Unless the point of the reorg is to pivot the product, the potential loss of time to market due to a reorg is a high price to pay.
- **New problems.** Any reorg can and will introduce new problems. A common technique for middles is to quickly identify the things that get “more difficult” or for bottoms to ask “well who will do X now”. From a top driving a reorg these often look like self-preservation rather than constructive input. It is a safe bet that almost everything one hears at this time is going to come become issues as middles and bottoms know their jobs. Even if it is presented in a selfish manner, the reality is that tops are not in touch enough with all the details of the work to just keep moving without adjusting. There’s a real balance to understanding what new problems are introduced in any org change and the impact those problems might have on the work.
- **Too much change, too little problem.** If the reasoning of a change is not sound for most people or there is a lot of feedback about strategy then there’s a chance that the reorg being executed is outsized relative to the problem. The feedback loop in this case is really pointing to an incomplete problem definition or simply a solution that doesn’t match the problem. This is a case where listening to the feedback can be especially enlightening.
- **Fatigue.** Reorgs can also be too much of a (good) thing. Teams can grow tired of the churn that comes from reorgs and enter a state of reorg fatigue. Finding the right cadence for org changes and finding the ability to get the reorg done and over with are important parts of an effective process. When more than one person starts sending mail saying how many managers or office moves they have had, then it might be time to consider this challenge.

- **Org distance.** Getting work done every day is how most people will evaluate an org change. The "org distance" between routine collaborators and resources is one measure commonly used. Org changes can potentially run into resistance when people perceive the changes mean they are "further away" from those they work with routinely. Commonly people will just count the org intersection point and see how far it moves or how different it becomes.
- **Accountability for the present and future.** Ultimately any organization needs to land clearly with who is accountable for what. This is a statement about specific people, code, and job functions. Every accountability has a "30,000 foot" view as well as an "on the ground" view. It is usually accountability at the detail level that matters in terms of selling through an org change. People will naturally want to know who "decides" which is another way of asking who is accountable. To answer who is accountable also requires one to answer where the resources are that "own" the code, designs, tests, etc. The transition from the present and all the work in flight to the future is a key part of any reorg effort.
- **Leadership and people.** One of the most challenging aspects of reorgs, particularly those that are about restructuring, is the ripple effect on staffing. At each level of the change, leaders need to be put in place. Some might be the existing leaders and others might be new. The image of [musical chairs](#) can come to mind, which is always stressful. Alternatively it is entirely possible to create an organization where there are more jobs of a certain type than people to fill them, which is equally stressful. As is always the case, making sure that when roles are created the people filling them are truly the right choice for the intended role is paramount. A new organization that is poorly staffed gets off to a challenging start.

In addition to these conceptual challenges, there are always potential pitfalls with respect to the process of reorganization. The tools of communication, listening, planning, empathy, adapting, are all absolutely critical. My own efforts at blogging started as part of the learning, sharing, and feedback loop for the team as we geared up for Windows 7 development (see [our book](#)) and re-organization. Blogging was one tool of many, but an effective way to drive a two-way dialog about changes (many posts were the result of questions or follow-up).

Finally, accountability for a reorg rests with management, specifically the line manager driving the org change. Reorgs are not something HR does for or on behalf of management. HR has valuable tools and a position of objectivity to assist, but they are not accountable or there to drive the process, pick up the pieces, or otherwise appear out in front of a reorg. A way to think of this is that as a manager resource allocation is your primary tool, therefore you can't really delegate org design and implementation because it is a primary job function—it is like a developer outsourcing coding (wait didn't we recently read about the [dev](#) that did that?). A common source of frustration is when someone is referred to HR when they raise issues about the goals and execution of a reorg.

Tools

While there are many human resources and management tools to support the communication, feedback, and discussion of a reorg, there are also some specific work management needs to do in order to drive an effective process. A big part of the use of these tools is the contribution from a large set of people on the team who are enrolled in driving the change.

The initial burden for getting things going well falls to the tops to communicate clearly. The **reasons for implementing an org change need to be clear and resonate with the team and discussed separately from the solution.** This is the problem statement and explains the why behind a reorg. The first sign of skipping steps in a reorg is that the first words, slide or paragraph show a reporting structure. Any reorg that leads with reporting structure is likely to be hit head-on with resistance. Of course most people will be anxious and want to know the structure first anyway, but as a leader of a reorg there is a real responsibility to explain the problems being solved first. This is not burying the real news because the real news is management waking up to a problem that needs to be solved.

With those two tools in place (the **why** and **what**), there are a few other tools that can help smooth over what is bound to be an emotional change for a team.

- **How.** The next thing to identify is how the work will get done. This is not the job of the top at a very gross “whole product” level but at the level down to some granular level that shows the implications of an org change are understood. Even in the largest organization, understanding at a level of 10-15 developers (engineers, marketing people, etc.) is really an acid test for knowing if an org change has been thought through.
- **Who.** The funny thing about reorgs is that the success of them depends on the most local of variables—individuals want to know what they work on and how the org affects their work, their career, and their place on the team. This is the “who” of a change. In an information based team (software!) this is your asset, not the code. So failing to really understand the who of an org change is going to make it rough. For this tool you need to enlist the help of managers throughout the team to make sure everyone is clear on who does what.
- **When.** The timeline of a reorg is critical for everyone. You need to take the time and yet not drag it out. How you balance this depends on the scope of the change and size of the organization.

Whenever a reorg is taking place, whether people agree or not, ultimately the members of the team will want to know about their own careers, skills, knowledge, and place in the new structure. As much as reorgs are about the big picture, successful reorgs are about the individuals that do the bulk of the work on any product team.

Or not...

One more tool for reorgs is simply not to do them. As strange as that sounds, the reality is that no organization is perfect and even if an organization is perfect it won't remain so for very long just because of the dynamic nature of product development and teams. People move around, features become more or less important as the technology landscape changes, some areas require more resources than planned or some require less, business models change and more.

This is why more often than not a reorg might not be the best place to spend the team's limited energy. Reorgs have the potential to substitute activity for progress and can cause an organization to be looking inward right when it needs to be outward focused the most.

That's not always the case, but it certainly is worth considering.

Yet that doesn't cure any problems a team or organization might be having. What are some changes tops can initiate or help to drive that can be substitutes for addressing the root cause of challenges that might be equally challenging but perhaps focus on the root cause more than an org change? Here are some examples for tech teams:

- **Align planning and execution.** Any time an organization has more than two products (or projects) that connect to each other (two unique products, front end/back end, etc.) there could be a need for alignment. The easiest way to have alignment is to align the planning and execution calendars. Teams that are joined by a calendar have the easiest time working together when it comes to hard decisions like what code to write and when. This alignment needs to be supported by tops—meaning once the bet is made to align, then you have to work within the constraints of release cadence, scope of product, external communication, and more. The converse of this is that putting teams together that have different schedules does not bring alignment—alignment in product design and code sharing essentially requires some degree of schedule alignment (at least in my experience).
- **Process alignment.** Teams that do the same things but do them differently will always have a hard time working together. From even abstract things like roles and responsibilities to extremely concrete things like how to categorize bugs or deliver daily builds, differences in processes can really make it hard to work together. A good thing to do is pick the processes that matter most to your orgs and just align (perhaps see [Managing through disagreement](#)).
- **Strategic choice.** Perhaps the real problem is not one that can be solved by organization at all and an org change is a substitute for a strategic choice (exit or enter a business, combine businesses, etc.) In this case, as painful as it may be, the org change only pushes accountability and delegates responsibility for something that should just be decided.
- **Decide to share code.** The hardest thing for dev teams to do is share code with other dev teams—50 years after the invention of the subroutine. Yet it is magical when teams do commit to doing so. How to share code effectively and how to manage the *provider* and *consumer* roles, especially in a complex org in many businesses, is an art form, but one that needs to be perfected, locally. As we all know, sharing code is great and also constraining—so again support from the broader perspective regarding additional constraints is critical. Sharing code is also a lot easier if teams are aligned on planning and execution timelines.

Implementing a reorg is a big step. It is always wise to think first about your problem statement and decide if you can attack the root cause in a much less disruptive way. This is especially true in a large organization where changing things “at the top” has much less of an impact on product evolution than many believe.

Customers

The Oshry framework also includes customers. Customers of course define the reason for making products in the first place.

The biggest challenge any multi-product organization faces is that customers want products and technologies (relevant to them, keeping in mind many products serve many different customer types) to appear to work together. From the outside, that is the customer perspective, when products don't appear to work together or appear to have arbitrary differences/redundancies then the obvious culprit is the org. The org was not

structured to work on that problem as an integrated whole. This can be seen as “shipping the org chart”.

In this case, the org chart for the products is not right—some things need to be “closer” or “one person” needs to be in charge of a couple of products. This goes a step further. When the design or quality is not right according to customers then the org is not right because the designers or testers were not organizationally working closely enough with developers.

You can see this multi-dimensional problem. It all boils down to graph theory and how you can connect all the parts of all of the products with the highest bandwidth and always connected flow of information, decisions, and more. This means it is much more difficult than it appears to use organization to address these perceived challenges. The side-effects of moving some things closer include moving other things farther apart, and the implications of the solution might be worse than the problem.

In the idealized world of small teams you can get everyone in the same conference room and decide everything. This tops out at about 40 -50 people. For example, Excel 5 had about that many developers. After that, organization is a tool that can help you to overcome this limitation. While it would be great to work on product families that always take fewer people, that isn't always possible just on the basis of the number of features it takes to be competitive in the market place over any period of time.

The substitute of anointing someone to oversee all aspects of a product is also a scale challenge. There are just so many hours in a day and only so many people that might fill such a role (if that is even possible to do). Once a person is managing a large number of related, but different, projects or just a large number of people then the ability for the large/complex team to act like a small team is limited. In other words, just joining two entities at the top does not necessarily mean they will appear to work better together for customers.

Yet, what everyone wants to avoid is a dynamic where your collective efforts result in “shipping the org chart” to customers.

Since you have to have an organization, which might be divided by geography, discipline, products, architectural layer, product release timing, business models, or more, the real tools to avoid shipping the org chart are planning, communication, and accountability. You can really never solve the multi-dimensional matrix of responsibility without making teams so large or structurally complex, or relying on a superhero manager that any value that might come from being on one team is lost. The converse to this is that designing products by a committee doesn't work either. Just taking a lot of complexity and sort of saying “work it out” usually fails to be optimal for any customer.

Because of the complexity of org changes in a large team, the best lesson I have learned is that a culture that adapts to solving problems turns out to be the best organization structure. Combine that with common views of roles/responsibilities, clear and reliable plans, and accountability and you can have the makings of an agile and flexible organization that can move work around, partner across projects, and deliver without using org structure as a high-order bit for strategic change.

—Steven Sinofsky

Be sure to check out this week's survey on org changes

Thanks for everyone that responded to our survey for "Using meetings to be more effective". In this survey, we hoped to learn together about the tools and characteristics that make meetings successful.

Here are the results:

- About half of our most recent meetings include a phone bridge, with about one third connecting via Voice over IP (i.e. Skype)
- In about one in six meetings, at least one person joins via a cell phone
- About half of our meetings take advantage of screen sharing and about half involve PowerPoint, though only in about one third was a projector used
- When asked about whether our last meeting was a success, on average (mean and median) we "neither agree nor disagree" that it was a success

In looking at drivers for what made us rate a meeting a success, there were some interesting findings:

- Regarding technologies, of the technologies queried (phone, cell, VoIP, screen sharing, PowerPoint, projector, and meeting software), only the use of a **projector** had a statistically significant impact on our success rating. However, **meetings with a projector ranked half a point lower** on a five point scale, than those without projectors
- Interestingly, presenters rated meetings with projectors lower than members of the audience, with a difference of about a half point, it's worth noting this was not correlated with slideshow software like PowerPoint
- Of the tips for success discussed, "**a fully understood context**" drove the success factor up a third-point, and a "**concise**" meeting (brevity) drove success up nearly a half-point.
- Interestingly, presenters rated meetings with "**a fully understood context**" higher than members of the audience

Bottom Line:

Modern meetings leverage online tools like to get everyone on the same page, though care should be taken during in-person meetings to not let the audio/visuals detract from your message as a presenter. Taking time before and during the meeting to create a **shared sense of context** and **keeping your message concise** seem to drive the best outcomes for everyone, presenter and audience alike.

Thanks, Cameron

###

Steven Sinofsky | May 6, 2013 at 9:00 am | Tags: [management](#), [organization](#) | Categories: [posts](#)
| URL: <http://wp.me/p31nkB-3O>

Comment

[See all comments](#)

Unsubscribe or change your email settings at [Manage Subscriptions](#).

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

[Redacted URL]

Thanks for flying with

[Redacted]