

Organizing Product Teams for Fast Time to Market and High Levels of Innovation



Martin L Abbott

William Armelin

February 2, 2018

Introduction – A Trip Through Time

30 years ago, in his book Innovation and Entrepreneurship, Peter Drucker identified that:

- 1) Innovation was (and still is today) the primary driver of US economic growth,
- 2) Most innovation comes from small businesses and firms (still the case today),
- 3) Most US employment comes from small businesses and firms (also still the case today).

A good portion of the book is dedicated to how large companies can attempt to put in place the practice and discipline of innovation. One key insight of the book is for large companies to be successful in innovation, they must separate the structure of new endeavors from those of the old.

Here Drucker is identifying something very important. He has identified a correlation between how small, innovative companies organize and their resulting success. He is also indicating that there are organizational overhead (bureaucracy), processes and ways of thinking that stand in the way of innovation in large companies.

50 years ago, at what we argue is the dawn of modern software development, Mel Conway wrote the following in his paper "[How Do Committees Invent?](#)":

“Probably the greatest single common factor behind many poorly designed systems now in existence has been the availability of a design organization in need of work.”

Mel is identifying that poor system design is largely caused by poor organization design, and as we will see later, that the two are inseparable; architectures and organizations must be designed together to work properly.

Many things have changed since Mel’s working career:

- The systems we produce have become increasingly complex. Operating systems in Conway’s time often had less than a million lines of code (e.g. Unix V1.0 in 1971) whereas Mac OSX has over 80 million lines of code.
- In the face of this increasing complexity, the barrier to entry to write software has decreased with the advent of easier programming languages, ORM tools and numerous other advances. In 1968, “software” professionals tended to have advanced degrees in mathematics whereas today one can join the workforce with a 2-month language boot camp.
- Customer expectations in terms of outcomes have increased. Whereas errors in early systems were “expected”, today customers expect flawlessness.
- Purchased on-premise software is on the decline, being replaced by software as a service (SaaS) that consumers and businesses “rent”. Customers don’t buy software anymore, they rent outcomes.

And yet, what Mel said 50 years ago and what Peter argued for 30 years ago are still true today. In 50 years, most companies haven’t heeded the advice of either of these legends.

This paper reviews several pieces of research, including our own, to illuminate a better way of organizing to develop modern internet products, systems and services. As we've experienced across the AKF customer base, this new way results in faster time to market, and higher levels of innovation.

Common Ground

Before diving deeper into this topic, I first want to define a few terms and phrases used within this paper. The first of these is the notion of innovation. I think it's useful to separate innovation into at least two categories:

Big "I" Innovation: This is the type of innovation evident within products that create new industries or significantly disrupt existing industries. Most large, successful companies have experienced this once organically. Unfortunately, as we'll discuss later and as Drucker points out above, these companies typically only experience it that one time organically and then seek to purchase it later as they grow.

Little "I" Innovation: This type of innovation may happen daily even in companies that can no longer create or disrupt markets. It may be the application of a technology in a new way that helps the company meet a time to market objective. It may also be changes to product attributes driven by data insights. This type of innovation ranges from small optimizations to large and meaningful changes to existing products but does not meet the threshold of creating a new industry/market or disrupting an existing industry/market.

Two other terms used later in this document are also worth a brief explanation:

Affective Conflict: Affective, or role-based conflict, is conflict between individuals or teams centered on "who" does something or "how" it gets done. The first is clearly a struggle for primacy or control and rarely adds value. The second is either a struggle for control, or the presence of micro management (if within the same organization). Both of these, as we will see, destroys the opportunity for Big "I" Innovation.

Cognitive Conflict: Cognitive conflict, if properly handled by managers, is magic. It centers on "what" should get done and "why" something should happen. If left unresolved, it can often devolve into "who" gets to make the decision and "how" something should be implemented. Managers should seek to engage in cognitive conflict and quickly help ensure it comes to fruition with decisions rather than letting teams start to fight over ownership.

I feel it's also necessary to put to bed the myths regarding being "conflict averse". Most people who use this term are discussing affective conflict. Avoiding this conflict and forcing managers to do their jobs in creating role clarity is a good practice – not a bad one. Unfortunately, most managers who say someone is "conflict averse" are really simply avoiding their jobs of helping to resolve role clarity problems. Now, on to the fun stuff!

Finally, a quick note on Agile development. This paper assumes that the reader is interested in the highest possible levels of innovation and that as such you are attempting to abide by Agile practices. When I use the word Agile, I associate it closely with the notion of "discovery" and in fact I use those words interchangeably within our practice. While the points in this paper are valuable outside of the Agile development, I feel it is important to clearly call out that discovery (and therefore Agile) are critical

to maximizing the possibility of “Big I” Innovation. The reasons for this go beyond the focus of this paper, but will be something we write about in the future.

Context

The Afghanistan War: 2 Phases

We view the US war in Afghanistan as 2 distinct phases. In the first phase, a small group (thousands) of American Special Operations units were sent to Afghanistan to oust the Taliban. In relatively short order, the units achieved the military objective of largely removing Taliban influence from Afghanistan.

The second phase consisted of occupying Afghanistan with conventional forces and “Nation Building”. During this phase of the war, troop levels surged to nearly 100,000 largely conventional troops. The result was constant conflict with terrorist elements, and the Taliban (and other) terrorist groups moved back into Afghanistan.

How is it that a small military force was successful whereas a force nearly two orders of magnitude larger was unsuccessful? The obvious choice is to point at the training and discipline of special operations units. While that surely accounts for some of the difference, it can’t possibly make up for 85,000 soldiers. Something else is going on.

Amazon vs. Everyone Else

From what we can tell, it appears that Amazon is the fastest company to reach \$100 Billion US in revenue...ever. At least if you exclude the likes of Enron, which clearly cheated on the way to that milestone. Amazon took roughly 20 years to achieve \$100 Billion, 8 years faster than the prior record holder, Walmart. But unlike Walmart, Amazon achieved this growth by continuously innovating on the path to greatness. Whereas Walmart focused on doing one thing well, selling merchandise at rock-bottom prices, Amazon continuously innovated and either disrupted existing markets or created new product segments and industries. Amazon was a pioneer of ecommerce, disrupted the publishing industry with e-readers, was a fast follower in the development and digital distribution of non-network video programming, did the same with music distribution, and created the infrastructure as a service industry. All of these we describe as innovation with a “Big I”. Amazon accomplished these big “I’s” while continuing to innovate with a “little I” (in this case similar to Walmart), by expanding the category offerings of their core commerce segment, offering 2-day and then 1-day fulfillment, and creating the definition of “best in class” for fulfillment systems (on par with Walmart’s famed supply chain solutions).

Amazon stands in stark contrast to what we at AKF call “Drucker’s Law”. Large companies, according to Drucker, don’t innovate with a “Big I” (AKF terms, not Druckers) but instead rely on the acquisition of smaller companies for their innovations.

A close look reveals that a handful of other large companies also seem to violate Drucker’s Law and appear to innovate with a “Big I”:

- Netflix innovating in the face of secular changes to develop streaming services as their DVD business dwindled,
- Facebook innovating to continue increasing ad revenues even as viewership changed from PCs to smaller form factor handhelds

- Apple creating the first home PCs (but losing the PC war) and then moving on to disrupt the music industry with iTunes.

Why are these companies so successful when most large companies continue to acquire their innovation?

The answer is quite simple. These companies share several traits in common with Special Operations units, elite sporting teams. These are:

- Team Size
- Team Construction
- Team Duration
- Team Alignment
- Empowerment
- Unifying and Compelling Vision
- Culture of Success

While many of these teams also happen to have highly skilled individuals, these individuals are not a necessity to achieve superior performance. Desirable, yes. Necessary, no. Disagree? Consider the following example:

In his bestselling book “Team of Teams”, General Stanley McChrystal discusses (paraphrasing) five midnight blue silhouettes flowing down an otherwise empty street in September 2004. The image evokes a notion of highly skilled special operators on a path to a deadly outcome. But the silhouettes McChrystal describes aren’t members of a SEAL team. They are al Qaeda insurgents on their way to kill 35 children. The members of this team weren’t part of an elite training program – they were local recruits and fighters smuggled in from foreign nations. Under-resourced and under-trained they were nevertheless successful in their endeavors. Size, team construction, outcome alignment, empowerment and (to them) a unifying and compelling vision trumped superior training and skills.

Its important to note that there are many successful companies that do NOT have the attributes defined above. A few companies have some product teams distributing these attributes, and others that do not. An example is Apple, which has functionally oriented teams for many of its products. Most of these solutions however (outside of the initial Apple and Apple II computers and iTunes) were fast followers. A perfect example is the iPhone stealing market share away from Blackberry and extending the usefulness of the phone as a compute device.

Innovation is not a guarantee for success. Nor is a lack of innovation a guarantee for failure. There are many successful companies that acquire solutions or “build a better mousetrap”. But just imagine how much better these products would be, and how much more successful these companies would be if they could increase the level of innovation within their organizations!

Let’s evaluate each of the dimensions above and apply them to product teams to define a better way of developing highly innovative products.

Team Size

Dunbar Numbers

Robin Dunbar is an anthropologist and evolutionary psychologist who was interested in understanding why primates spent so much time performing social grooming. During his investigation he stumbled upon a correlation between the brain size of primates and group size. Further analysis seemed to indicate the same relationship existed for humans, and that 150 (often called Dunbar's Number) is the largest number of folks with whom you can have a "meaningful" relationship. Somewhat jokingly, Dunbar indicated that this number is "the number of people you would not feel embarrassed about joining uninvited for a drink if you happened to bump into them in a bar".

Dunbar also introduced the notion of "Dunbar Layers". These are ever decreasing numbers of people with whom you can have conversely increasing significant depths of relationships. Those layers exist at 50, 15 and 5 people, with five being the number of folks with whom you can have the greatest degree of social interaction and knowledge – your "best friends" or BFFs.

A close examination shows that these groups, especially 5 and 15, also appear to be the upper bounds of elite teams. SEAL platoons have 16 personnel, US Army Special Forces (Green Berets) Operational Detachment Alphas (ODAs or "A-Teams") have 12, football teams have 11 personnel on the field at any time, baseball 9, basketball 5, etc. Coincidence? We think not.

Conway's Advice

Conway, also in his paper "[How Do Committees Invent?](#)", sets a limit on team size. Quoting Conway,

"...each individual must have at most one superior and at most approximately seven subordinates."

Conway gets this number from both communication theory and graph theory. Computer Scientists know that the number of edges in a fully connected graph (connecting all vertices) conforms to the following equation:

$$\frac{(N \times (N - 1))}{2}$$

This equation also happens to be the amount of communication overhead necessary to coordinate the number of people (N) on a team towards a common objective. Put another way, it's a non-value returning tax on the entire team. As N grows, the amount of time everyone spends coordinating activities with those of the rest of the team grows asymptotically exponentially. With very large teams, more time is spent coordinating than working.

Amazon

By now, you are likely familiar with the "2 Pizza Team" concept popularized and employed by Amazon. But do you know the origination of that approach? Rick Dalzell, who was then the CTO/CIO of Amazon started a reading group with his key engineers at Amazon. One of the first books they read was Fred Brooks' Mythical Man Month. Fred Brooks (who interestingly coined the term "Conway's Law" – more on that later) also indicated that teams should be small and included the above the equation indicating

the overhead in coordinating a team being the square of its size. Rick and his team took this to heart when creating their organizations. But it was Bezos, in his marketing genius, that coined the term “2 Pizza Team”. For more on that story, read the second edition of [Scalability Rules](#).

Key Takeaway

- Teams should be small, no larger than 12 to 15 people.

Team Construction

Small team size is necessary but insufficient to achieve superior performance. An important attribute of all our research on highly innovative teams is that they are constructed with all the skills necessary to achieve their desired outcomes.

Let’s revisit a Special Forces A-Team. The team has everything necessary to conduct unconventional warfare for an extended time behind enemy lines: 2 leaders, 2 medics (each of whom can perform surgery or prescribe medications), 2 engineers, 2 weapons specialists, 2 operations/intelligence specialists and 2 communications specialists. Virtually everything the team needs, including the ability to coordinate and call for close air support or field artillery, exists within the team.

The Tyranny of the How

How would you respond to the following question: “What is more important in your business – what you produce, or how you go about producing it?”. This question does not imply that one of the two (what or how) is not important. Rather, it is a question asking you to indicate which of the two is more important. Most people of whom we ask this question answer with “What we do is at least slightly more important than how we do it”.

Now answer this question: “Do you organize your company around what you build, or how you build it?”. Most of us are organized functionally, along the dimensions of how we do work. Product management is a distinct organization from engineering. Engineering is distinct from QA. QA is distinct from Operations, etc. Why in the heck, if “what” is more important than “how” do we organize by the “how”?

An analysis of some of the most innovative companies uncovers a similar attribute: teams are cross functional in nature and organize more around “what” they produce – not how they produce it. There exists in these teams not only the skills necessary to write code but also to support the software engineers in their endeavors: infrastructure skills, devops/support skills, product ownership skills, quality assurance skills, etc. In fact, Brooks’ Mythical Man Month (1975) indicates the same need in teams when he uses his surgical team analogy: teams must have everything necessary to make the surgeon successful in achieving her desired outcomes.

AKF Research – Conflict and Innovation

Our own research helps describe why cross functional teams (instead of functionally aligned teams) are simply better. Cross functional teams enjoy higher levels of morale, faster time to market, higher levels of innovation and lower levels of conflict.

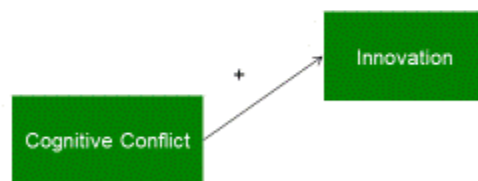
First let's define two terms. Researchers like to split conflict into two types: Cognitive and Affective. Cognitive conflict is "beneficial conflict". It often addresses what a company will do, or why a company will do it. An example of cognitive conflict is brainstorming, in which teams build on ideas and generate something, as a result of team interaction, better than that which any one person could accomplish. Generally, anytime we discuss options regarding what we will do and why we will do it we are engaging in cognitive conflict.

As cognitive conflict increases, so does the level of innovation. The reasons are obvious. Cognitive conflict helps us increase the range of options about what we will do and helps generate compelling reasons as to why we should do those things. Here we have the first part of a multi-step structural equation model denoting how team interactions and constructs drive innovation.

Cognitive Conflict



As cognitive conflict increases (what and why), so does innovation

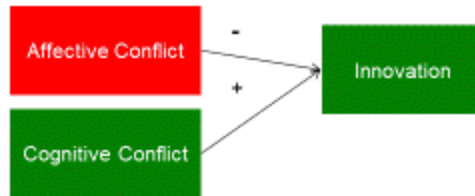


**De Dreu and Weingart 2003

Affective (or role based) conflict within organizations is directly correlated with time to market and inversely correlated with levels of innovation within teams. As affective conflict within teams increase, time to market increases (slows) because teams spend more time in resolving who should own something and how it should be done. Increases in affective conflict also reduce morale, which in turn significantly reduces a team's innovation. When morale decreases, teams become disengaged and follow orders rather than seeking innovative solutions.

Affective Conflict

As affective conflict increases (who and how), innovation decreases

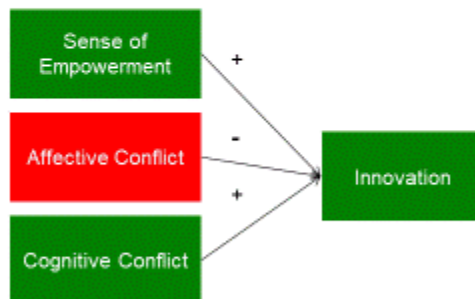


**Amason and Mooney, 1999; De Dreu and Weingart 2003

As feelings of empowerment increase within a team, so does the level of innovation. Teams feel that they “own” or are “wed” to outcomes and thusly are more likely to engage in innovative activities. Time to market shortens and quality generally increases.

Empowerment

Feelings of empowerment increase innovation



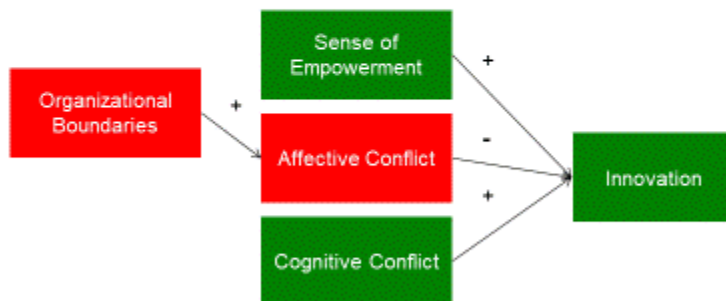
**Spreitzer, De Janasz, 1999

Continuing our model from right to left, we start to explore the drivers of various forms of conflict. Organizational boundaries across which work must travel or over which collaboration must happen

increase affective conflict. Organizations help reinforce the notion of social identity, in which is embedded certain notions of ownership. Who should be responsible or own certain tasks or decisions? How should those activities be performed? These types of questions, if unclear organizationally and if not resolved immediately fuel innovation-destroying affective conflict.

Org Boundaries

Organizational boundaries across which collaboration must happen increase affective conflict



** Tafel 1974; Hogg and Terry 2000; Cummings and Kiesler 2005

Functional organizations and waterfall development methodologies are largely at fault in this domain. These structures and processes create moral hazards (defined as a lack of incentive to guard against risk where one is protected from its consequences) for other participants in the value pipeline, especially where goals, objectives and outcomes are not equally shared by each of the organizations. A Product Management team owning certain revenue targets may make product specifications or define approaches that limit the ease with which an outcome may be achieved. The product engineering team then inherits that impact and, being incited by time to market, finds that they cannot achieve their objective. Conflict results. Product managers feel they “own” how the solution should work regardless of the impact downstream. Contrast this with a product owner and the remainder of the team jointly feeling they must accomplish an outcome together and making trade-offs between product approach and design fluidly.

Similarly, an engineering team may make software decisions that have an impact to the availability of the service to an end customer. The Operations team then becomes upset at the engineering team as the Ops team is the “owner” of the availability goal. Both teams may engage in unhealthy discussions about who gets to own availability related decisions, or how something will be implemented. Contrast this with the engineers, operations personnel, and product owner jointly owning all results (revenue, time to market, availability) and making tradeoffs in real time to jointly achieve their objectives.

Now we come to the notion of cross functional, or experientially diverse teams. As experiential diversity (in terms of skillsets and approaches) increases, so does cognitive conflict. With more skills involved in a

decision, the probability of coming to the best possible result increases. Unfortunately, if not well managed, individuals may also start to engage in discussions around “who” owns a decision which in turn increases affective conflict.

Here again we come to the notion of identity. When groups of individuals identify as the same team, affective conflict tends to decrease. It’s when team identities (QA vs Engineering, Engineering vs. Product) are distinct that probability of affective conflict increases. As such, having disparate skillsets on the same team creates the highest possible cognitive conflict and lowest potential affective conflict. The identity test can be performed as follows:

Ask an individual what they do. If they respond with the function they perform while they are embedded and working within an outcome focused team, you will likely have high levels of affective conflict and low comparative levels of innovation (against what is possible). If the individual responds with “I am a member of this outcome focused team”, you will likely achieve high levels of cognitive conflict and high levels of innovation.

Social Identity is a powerful concept. Imagine what would happen if NFL teams organized primarily by the skillset family rather than the “outcome” (offense and defense) team. Player identities would be primarily rooted in their family – NFL offensive and defensive linemen would argue with offensive and defensive backs. Struggles would ensue regarding which team gets to own decisions, and how something would happen. Coaches would argue nonstop about the same things.

Social identity is also a “tricky” thing. We all have multiple identities: Mom, Wife, CEO, Harvard Business School graduate, Director of a public company. These identities are contextually activated. When you are home, you may be a Mom or a Wife primarily (though only thinking of one primarily at a time as it turns out). At work, you are the CEO. When going to a reunion, you see yourself as an HBS graduate.

It’s when these identities conflict with your context that affective conflict ensues. The NFL offensive lineman who affiliates first with other linemen while in offensive team discussions, the engineer who views herself as an engineer first when in outcome-oriented team discussions, etc. As a side note, this notion of conflicting identities is one of many reasons why it makes so much sense to split the role of CEO from that of Chairman of the Board.

Instead, the teams are properly constructed around outcomes – scoring points and preventing teams from scoring. “Families” (linemen, backs, receivers) still exist, but they are subordinated to the offensive and defensive (outcome) coordinators.

The same holds true with special operations units in the military. The team commander within A-teams has everything at his disposal. Job families where mentoring is performed still exists – but the team identifies primarily with their ODA (again, Operational Detachment Alphas or A-teams) and secondarily with their job families. One member of an ODA will stick up for his teammate before siding with a member of another ODA within the same job family.

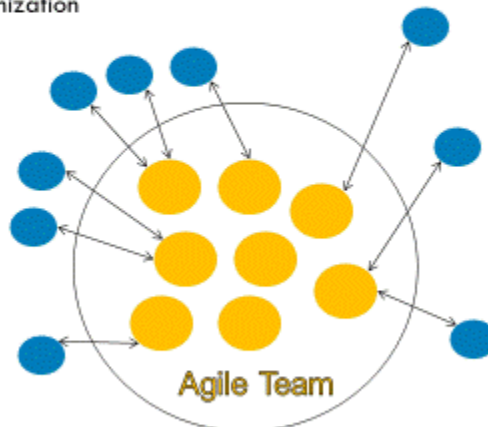
Finally, we come to the research on serial entrepreneurs – those individuals who have successfully started multiple companies. One of the traits these folks have in common is the notion of a very diverse network of “loose ties” – individuals between the Dunbar 15 and 150 range. These loose ties become both sources for, and validation of, ideas and approaches. Early social network researchers (before our

notion of online social networks) have also found that teams of diverse skillsets also often have a large network of diverse individuals from whom they can source information and against whom they can “bounce ideas”. Increasing the sources for information helps these teams introduce new best practices, approaches, and technologies. Similarly, having more people with whom these teams can discuss ideas helps to quickly validate or invalidate various paths. Both are incredibly useful for increasing team innovation.

Network Diversity

Best when high number of **non-overlapping** network links (loose acquaintances) outside of the organization

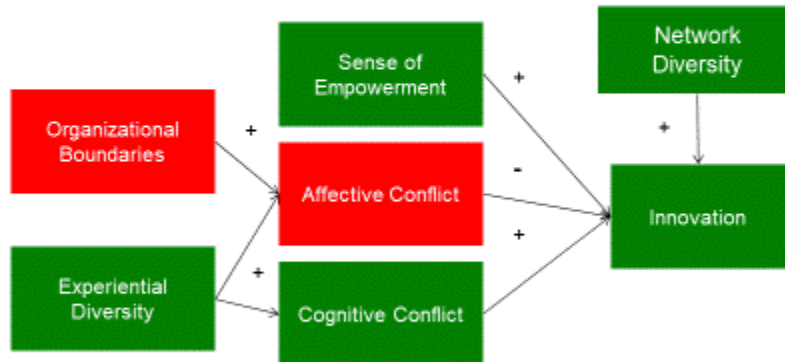
Network
Diversity



** Burt 2001; Bassett Jones 2005; Abbott, Lyytinen et al. 2010

The entire AKF model of organizational innovation now looks like this:

Driving Innovation



Our research concurs with Fred Brooks' analysis back in 1976 – teams should be formed around the outcomes they desire to achieve. Further, from our research we know that the identity of an individual is a good indication of the level of innovation the organization is likely to achieve. If someone performing a product management function identifies themselves as a product manager, rather than as a member of a team associated with an outcome, conflict is likely to ensue.

Key Takeaways

- Teams should be constructed cross-functionally, with each team having all the skills necessary to produce a group of customer-specific outcome.
- Individual identity should be rooted in the cross-functional team with which they work – not the functional family to which they belong.
- Note the notion of team and family. Families are collections of people with something in common – teams share outcomes.

Team Duration/Tenure

In 1965, psychologist Bruce Tuckman published his theory of group dynamics. This theory describes the stages (or phases) through which a team progresses enroute to optimal productivity. While generally useful for any organization, and prescriptive as to what leaders should do to boost performance, it has profound impacts to Agile development practices and how we build organizations them.

Forming

Tuckman's first stage is forming. This is where the team first comes together. Here, the individuals are trying to get to know each other. They tend to be polite and cordial, but they do not fully trust each other.

In this stage, team productivity and team conflict are low. The team spends time agreeing to what the team is supposed to do. This lack of agreement of the team's purpose can cause members to miss goals because they are individually targeting different things. Team members rely on patterned behavior and look to the team leader for guidance and direction. The team members want to be accepted by the group. Cautious behavior on the part of the team starts to depress overall team outcomes. Good leadership, emphasizing goals and outcomes is important to set the stage for future team behaviors and outcomes.

Storming

Once the team's goals are clear, they move into the next stage, storming. Here, the team starts to develop a plan to achieve the goal and defines what to do and who does it. Friction starts to occur as members propose different ideas. Trust within the team remains low and affective conflict rises as people vie for control. Cliques can form. Productivity drops even lower than in the first stage.

Once the team agrees on the plan and the roles and responsibilities, it can move to the next stage. Without agreement, the team can get stuck. Symptoms include poor coordination, people doing the wrong things and missing deadlines, to name a few. Good leadership here focuses on fast affective conflict resolution and serves to help reinforce team goals and outcomes in order to quickly move to more productive phases.

Norming

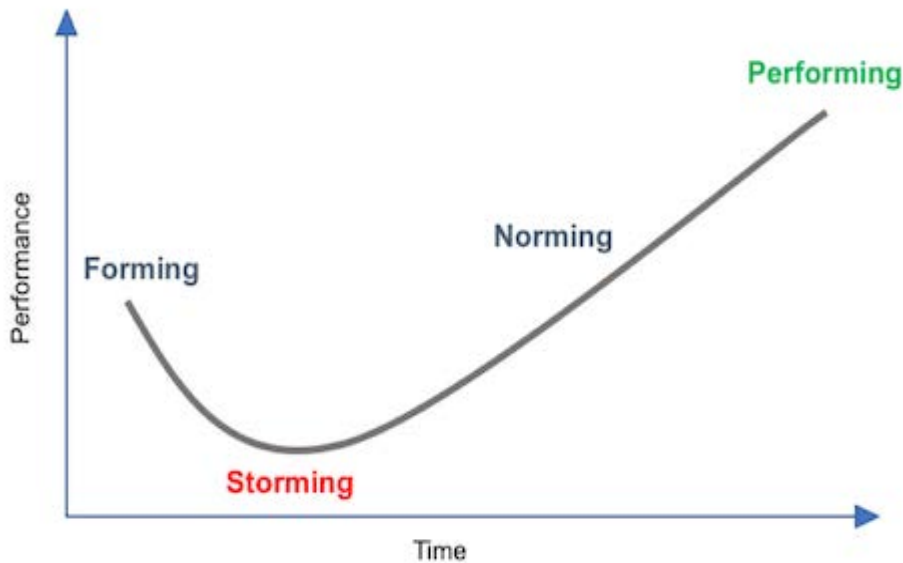
Once team members agree to the plan and understand their roles, they enter the norming phase. Affective conflict goes down, cognitive (beneficial) conflict and trust increase. The team focuses on how to get things done and productivity begins to increase. The team develops "norms" about how to work together and collaborate. A lack of these norms can cause issues such as low quality and missed deadlines.

Leadership within the team becomes clear and cliques dissolve. Members begin to identify with one another and the level of trust in their personal relationships contributes to the development of group cohesion. The team begins to experience a sense of group belonging and a feeling of relief from resolving interpersonal conflicts. Team identity starts to take hold and innovation and creativity within the team increases. The members feel an openness and cohesion on both a personal and task level. They feel good about being part of the team.

Performing

The final stage, performing, is not achieved by all teams. This stage is marked by an interdependence in personal relations and problem solving within the realm of the team's tasks. Team members share a common goal, understand the plan to achieve it, know their roles and how to work together. The team is firing on all cylinders. At this point, the team is highly productive and collaborates well. They are trusting of each other and "have each other's back." Healthy conflict is encouraged. There is unity: group identity is complete, group morale is high, and group loyalty is intense.

Not all teams get to this phase. They can get stuck in a previous phase or slide back into them from a higher phase. Leadership that focuses on affective conflict resolution, team identity creation, a compelling vision and goals to achieve that vision is critical to reaching the Performing phase. It is usually not easy for teams to quickly progress through these stages, and it often takes 6 months or more for a team to reach the Performing phase.



Impact to Agile Development

We often see companies make the mistake of coalescing teams around initiatives. Sometimes called "virtual teams" or "matrixed teams", these teams suffer the underperforming phases of Tuckman's curve repeatedly, especially when these initiatives are of durations shorter than 6 months. But even with durations of a year, six months of that time is spent getting the team to an optimum level of performance.

Tuckman's analysis indicates that teams should be together for no less than a year (giving a 6 month return on a 6-month investment) and ideally for about 3 years. The upper limit being informed by the research on group think and its implications to creativity, performance and innovation within teams. Teams then should become semi-permanent and we should seek to move work to teams rather than form teams around work. To be successful here, we need multi-disciplinary teams capable of handling all the work they may get assigned. Further, the team needs to be familiar with and "own" the outcomes associated with the solution (or architectural components) with which they work. More on that in future articles discussing Conway's Law and Empathy Groups.

Key Takeaways:

- Keep teams formed for at least 6 months and ideally as close to 3 years as possible.
- Move work to teams
- Do not form teams around work

Team Alignment

Conway's Law – the "Law" that Almost Wasn't

Conway's law had a rather precarious beginning. Harvard Business Review rejected Conway's thesis, buried as it was in the 43d paragraph of a 45-paragraph paper, claiming he had not proven it.

But Mel had a PhD in Mathematics (from Case Western Reserve University – Go Spartans!), and like most PhDs he was accustomed to journal rejections. Mel resubmitted the paper to "Datamation", a well-respected IT journal of the time, and his paper "How Do Committees Invent?" was published in 1968.

It wasn't until 1975, however, that the moniker "Conway's Law" came to be. Fred Brooks both coined the term and popularized Conway's thesis in his first edition of the Mythical Man Month. It has since been one of the most widely cited, important and unfortunately *incorrectly understood and applied notions* in the domain of product development.

Cliff's Notes to "How Do Committees Invent?" (the article in which the law resides)

Conway's thesis, in his words:

... organizations which design systems (in the broad sense used here) are constrained to produce designs which are copies of the communication structures of these organizations.

Conway calls this self-similarity between organizations and designs "homomorphism". The preamble to the thesis helps explain the breadth and depth:

... the very act of organizing a design team means that certain design decisions have already been made, explicitly or otherwise

...

Every time a delegation is made ... the class of design alternatives which can be effectively pursued is also narrowed.

...

Because the design which occurs first is almost never the best possible, the prevailing system concept may need to change. Therefore, flexibility of organization is important to effective design.

...

Example. A contract research organization had eight people who were to produce a COBOL and an ALGOL compiler. After some initial estimates of difficulty and time, five people were assigned to the COBOL job and three to the ALGOL job. The resulting COBOL compiler ran in five phases, the ALGOL compiler ran in three.

There are 3 very important points, and one very good example, in the quotes above:

- 1) Organizations and design/architecture are intrinsically linked. The organization affects and constrains the architecture - the opposite is not true.
- 2) Depth of an organization negatively affects design flexibility. The deeper the hierarchy of an organization, the less flexible (or alternatively more constrained) the resulting architecture.
- 3) We will make mistakes and must organize to quickly fix these.

Important corollaries to Conway's law suggest that if either an organization or a design change, without a corresponding change to the other, the product will be at risk.

Common Failures

There are three very common failures in organization and architecture within our clients, the first four of which relate directly to Conway's points above:

- 1) Organizations and architectures have been designed separately. Given the homomorphism that Conway describes, you simply CANNOT do this.
- 2) Deep, hierarchical organizations. Again – this will constrain design.
- 3) Lack of flexibility. Companies tend to plan for success. Instead, assume failure, learning, and adaptation (think “discovery” and “Agile” instead of “requirements” and “Waterfall”).

Key Takeaways

- Team organization affects the architecture. Design the two in parallel.
- Organizational hurts flexibility and agility – create as flat an organization as possible.
- No design is perfect – expect to have to respond quickly.

Empowerment

Earlier when we introduced the model for the drivers of innovation within a team, we briefly discussed the notion of empowerment. Empowerment is so important to time to market and innovation that we feel we needed to further emphasize it here.

Too often leaders say that they are “empowering” a team or a person to act without ensuring that the prerequisites of empowerment are in place. The most important component of empowerment is the “power” portion of the word. This alone is worthy of further discussion illustrated through a handful of examples.

A carpenter isn’t “empowered” to hammer a nail into a 2x4 simply because the general contractor directs her to do so or delegates to her the responsibility to do so. The carpenter can’t perform her function without having the necessary tools (a hammer), and supplies (nail, 2x4s, etc) in place.

Similarly, a surgeon isn’t empowered to perform an operation by the surgical director simply because the surgical director schedules the surgery and delegates the task to the surgeon. The surgeon needs to have a support team around her to be successful in her endeavor which includes, among other things, an anesthesiologist, nurses, etc.

These two examples are meant to differentiate the notion of delegation from that of empowerment. One can easily delegate a task by simply directing someone else to do it and holding them responsible. In so doing, they delegate the responsibility for completion (though as we indicate in [The Art of Scalability](#), *never* the accountability for results). But empowerment requires additional actions. It is meant to indicate something more than delegation.

To truly empower a person or organization, one must give them all the tools and supporting capabilities to complete the desired outcome. In the world of services development, this means ensuring that the team has all the skills, tools, process and appropriate budget to complete the outcome. The academic research relating empowerment to innovation all rely on validated psychometric scales. These scales measure the respondent (the person reporting whether they are empowered) along dimensions including their self-competency, locus of control, skills and access to the proper support to achieve an outcome.

Truly empowering teams through ensuring that they have the right skills embedded within a team to achieve an outcome is one of the most important drivers of team innovation. Having those skills in a separate team from which the person to whom you’ve delegated must request service negatively impacts empowerment and as a result, innovation.

Key Takeaways:

- Empowerment and delegation are different concepts.
- Empowerment requires the movement of all of the required skills and tools to achieve an outcome into the team being empowered.
- Confusing delegation and empowerment reduces innovation.

Unifying/Compelling Vision

As Richard Boyatzis points out in Resonant Leadership, vision is more than just a destination. Vision must inspire teams to ever greater and increasing accomplishments. Ideally, a compelling vision tugs at the heartstrings of the team constituents and is the thing that motivates them to come to work in the morning. Further, a vision needs to be the thing, that in the absence of other stimuli, moves teams to the right actions.

In extreme cases, such as with US Special Operations Units, vision not only drives teams, it is a reason people with the right behaviors and organizational alignment to outcomes join teams. People know how difficult it is to make the cut within these units and that only the best need apply long before they consider joining. The result is a virtuous cycle of elite individuals, ultra-dedicated to a mission they've aspired to achieve for significant portions of their lives. The people who make this cut then drive each other to ever higher levels of performance.

The same is true for certain high profile and hyper growth companies that publicly proclaim their vision – at least when that vision is voiced and believed in by employees. eBay, in its early years, had such a phenomenon. Executives claimed, and employees felt, that they were changing the world for the small “mom and pop” seller. In extreme cases, eBay helped people who could not otherwise hold a job (e.g. due to illness) make a living from home. Knowing that doing a great job building an effective product helped people make a living or achieve their dream of being their own boss was one of the many reasons employees worked so hard. One extreme example of this was eBay's “Auction for America”, launched shortly after the World Trade Center fell in 2001. Employees donated time beyond that necessary to run the business to create a platform on which people could buy and sell goods, proceeds from which went to benefit the victims of 9/11. While I would not put eBay in the category of highly innovative (it misses many of the other attributes), the compelling vision was certainly there.

Key Takeaways:

- Vision should go beyond a destination – it should “speak to the hearts” of the team constituents to turbo-charge innovation.

Culture of Success

It's hard to think of a culture more intent on success than that of Amazon. In Jeff Bezos' 2016 letter to shareholders he writes:

"Day 2 is stasis. Followed by irrelevance. Followed by excruciating, painful decline. Followed by death. And that is why it is always Day 1."

Jeff is talking about Amazon's "Day 1" culture – a culture he strives hard to create. The building in which his office is located is named Day 1. In fact, when he moved buildings, he moved the name of the building to his new building. Day 1 is about hope, drive, and the thrill of what might be. Day 1 is all risk with the hope and dream of high rewards. Day 1 is about a small team, working to achieve something very, very big.

Consider also the fifth portion of the Ranger Creed (the "E" of Ranger):

"Energetically will I meet the enemies of my country. I shall defeat them on the field of battle for I am better trained and will fight with all my might. Surrender is not a Ranger word. I will never leave a fallen comrade to fall into the hands of the enemy and under no circumstances will I ever embarrass my country."

These words have driven many a Ranger hero to run through enemy fire to retrieve a downed Ranger. This one short paragraph explains some of the most important "dos and don'ts" of the Ranger ethos.

First let's address the culture statement. As with the Vision of a company (or team), the statement must be equal parts aspirational, inspirational, and achievable. The statement of culture should help you attract and retain the right people – and be consistent with the outcomes you want to achieve.

With that aside, let's also recognize that the statement alone is useless without leaders who embody the statement and make it "real" through their actions every day. This is absolutely the case at Amazon. So much so, that one of the most common reasons people depart is that the company is so insanely focused on getting things done. Amazon acts like the world's biggest startup. Put frankly, startups are hard work and they simply aren't for everyone.

Similarly, Ranger Battalions live their creed (only a piece of which is quoted above). The creed embodies the Ranger culture, and the Rangers spend a lot of time evaluating individuals against that culture. Some of the very best soldiers and officers in the US Army would never make it a month in a Ranger Battalion.

Vision and culture interact to create the incentives for innovation and growth. You drive to help come up with innovative solutions because failing to do so will let down your constituency and your peers. In all our research, we've never seen a highly innovative company that did not have a compelling vision or a culture of success.

Key Takeaways:

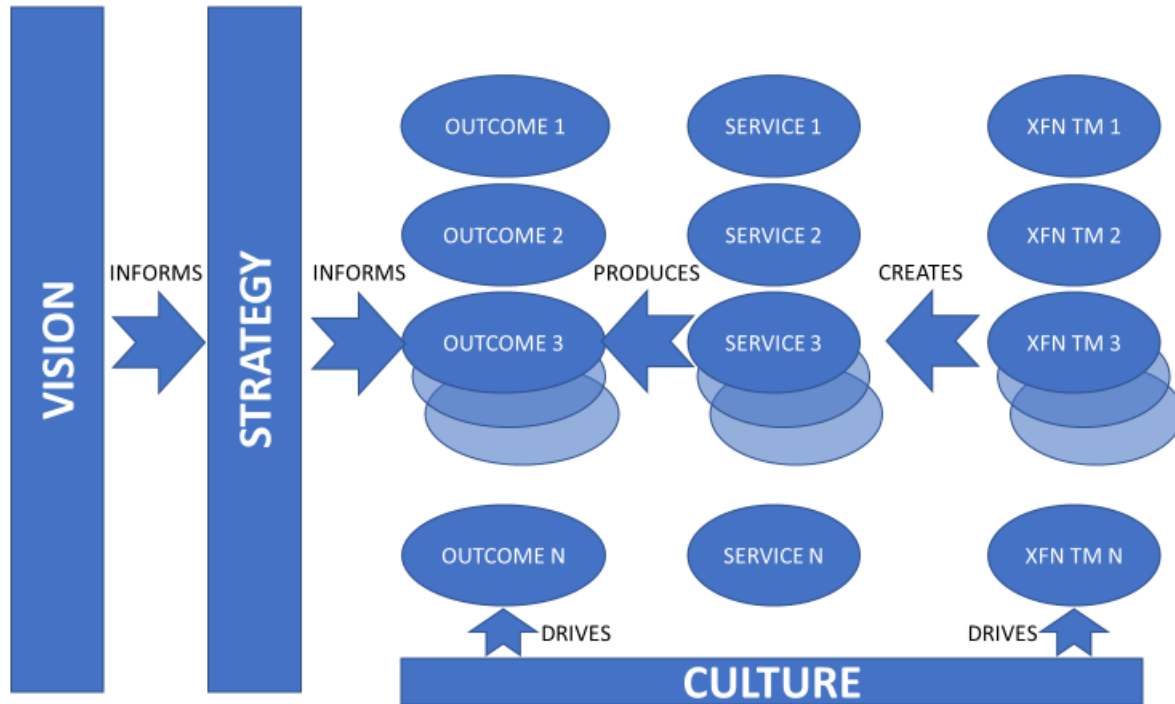
- Culture creates teams that expect team participants to perform at their highest levels.
- Culture isn't a statement – it must be "lived" first by the executives, and then by the employees.

Putting it All Together

Having covered the extant research of which we are aware, we'll end with a set of principles (somewhat prescriptive) as to how product teams should be constructed and how product development lifecycles should work.

- 1) Team Size: Small teams – ideally no more than 15.
- 2) Team Construction: Cross functional. Everyone necessary to develop a solution included in the team including:
 - a. Product Managers (Owners)
 - b. Engineers
 - c. Infrastructure/DevOps
 - d. QA
 - e. Designers as appropriate
 - f. Etc.
- 3) Team Duration: No less than 6 months, no more than 3 years.
- 4) Team Alignment: Completely aligned to outcomes.
 - a. Outcomes informed by vision.
 - b. Architecture aligned to outcomes.
 - c. Organization aligned to architecture – owning specific components.
 - d. Organizations own outcomes.
- 5) Team Empowerment: Teams own outcomes and have the skills to achieve those outcomes.
- 6) Vision: Creates the reason teams come to work and work hard to achieve results. Informs product outcomes, which in turn are owned by teams.
- 7) Culture: Established around success and driving each other to higher performance.

The alignment of outcomes, architecture (vis-à-vis services) and cross functional teams is depicted below.



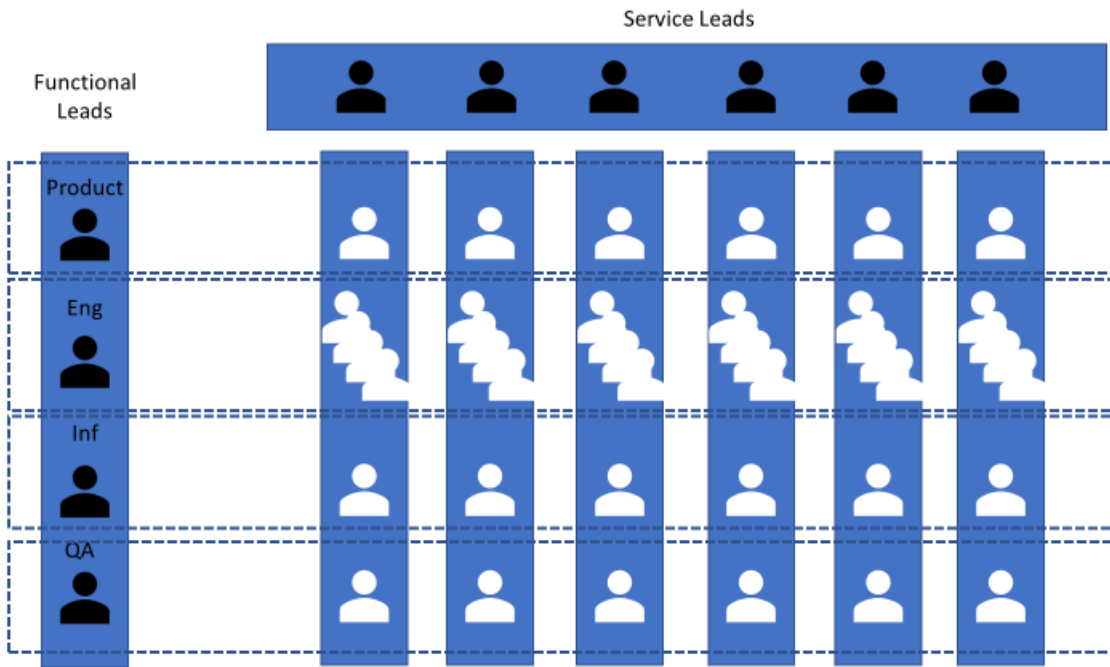
Consistent with the recommendations within Conway’s Law, each cross functional team owns a component of the architecture called a service. No two teams own the same service, but one team may own several services. The architecture and organization structure are now properly mapped.

Furthermore, each service is developed owning one or more outcomes. While multiple services may ultimately affect the same or similar outcomes, no service is developed without having a specific outcome for which it is at least partially responsible. Any development on this service should be predicated on a change in one of its many outcomes. Put another way, we should never modify a service unless there is a measurable outcome associated with the effort.

The Agile Organization Chart

While we’ve spent a fair amount of time arguing for why functional organizations are bad when used as the primary template for organization structure, we are not implying that they should go completely away.

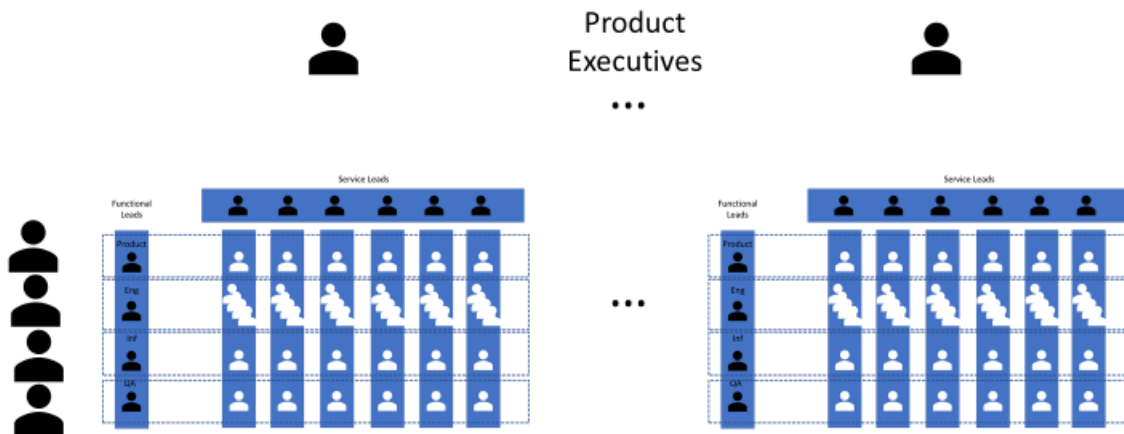
Functional organizations still play a role in defining the standards by which something should be built. In the case of product management, each member participates in a service team and is also spending time helping to define the product management roadmap. They move between the two teams, but their primary identity is embedded within the service team. The organization chart then looks like this – with service teams being at least slightly more important than the functional team.



The graphic above is not meant to cover every skill necessary within a cross functional team but rather be an indication of how you should organize.

Remember, every component of an NFL offense has a “family” from which they learn skills. The fact that those families are subservient to the offense is a good thing – not a bad thing. Special Forces teams also make the A-Teams the most important component as they win or lose the battle – not the families to which the individuals belong. Finally, the companies that display the highest level of innovation in our research are all attempting to make teams more important than families – and are being successful in doing so.

In larger organizations, it is sometimes necessary to start adding hierarchy to the Agile Organization. One functional manager may simply have too many people to try to mentor and coach, and as we add services we may need additional “leads” that own logical groupings of services. This is easily accomplished by adding additional managers and or executives to those areas and forming product groups as appropriate. The resulting organization looks like this.



Functional Executives

Common Resistance to The Cross Functional Model

Make no mistake, changing organization structure is difficult work. Unless you have worked in a startup, most of your experience until you reach the executive ranks is working within a functional group. Before becoming CTOs, most of us have spent a majority of our time developing software, working on infrastructure or the like. Before becoming CEOs, most of us have been within a single functional domain (finance, engineering, marketing, etc).

The following are the most common reasons we hear for not implementing a cross functional model, and our response to each of them:

“The Model Destroys Technical Excellence”

There is nothing about the cross functional model that stands in the way of technical excellence. It simply recognizes that “What we do” is more important than “How we do it”. It does not say that technical mentoring, coaching and development isn’t important. In fact, it recognizes its importance, and the fact that it is less important than the right skills to work together to develop a solution.

In fact, we argue that the functional model destroys innovation for all the reasons we’ve indicated here. Remember – startups have proven themselves more innovative than large companies. This model is in effect a way to create a group of startups and leverage the power of their innovation driving organization structure.

“The Model Creates Anarchy”

This is also false. We still have executives developing strategy, functional teams (product management, etc.) defining subservient strategies and roadmaps. But the primary identities of these folks are embedded within the teams that implement these solutions.

“The Model Destroys Career Paths”

Quite the opposite is true. For most of us in functional models, the first exposure we have to multi-disciplinary management is at the executive level. For many folks, that’s 20 years into a career. This model promotes multi-disciplinary understanding early in a career and demands that employees develop both breadth and depth.

“The Model Requires More People and Costs More”

Absolutely not true. In fact, this model tends to require less people as communication and collaboration within traditional functional organizations declines significantly. This means less middle management and less “project managers” responsible for trying to coordinate virtual teams comprised of functions. In fact, in our experience in implementing this model, we’ve seen many product organizations decline in total personnel by 3% for large organizations while increasing throughput by more than 15%.

“Sometimes a Team Doesn’t Require a Full Headcount for One or More Disciplines”

That’s happens frequently, especially in the case of user experience or design folks. But make sure that the same person attends any given service’s standup meetings. If you need .25 of a person for team, spread that person across four teams and stagger the standups such that they can attend.

“This is Just a Matrixed Organization”

From a literal perspective, yes, it is. But it differs from the way we’ve traditionally used that term in that it recognizes that “What” we do is more important than “How” we do it. Again – not that both aren’t important, but one is more important than the other.

“Our HR Systems Won’t Support This”

Well, isn’t that just the tail wagging the dog?

“This Means the Product Owner is Always the Leader”

No, it doesn’t. The product owner is certainly a critical portion of the team. However, the question of who “leads” the team should be answered by who the best leader is – not which functional role is best.

The Path to Implementation

We see great benefit to applying Agile concepts to organizational design. While we've included a blue print of where the most innovative teams are, or to which they are in the process of moving, we do not believe you should attempt a wholesale turnaround of your organization tomorrow.

As with software, you are going to make mistakes in your organization. First, define the metrics you hope to improve by changing the organization and start to measure them before any change. These may include:

- 1) Identifying a measurement of innovation.
- 2) Identifying a measurement of empowerment (likely psychometric)
- 3) Baselineing your velocity
- 4) Determining a way to measure time to market and cost of development.

Now design a prototype organization and try it out with a portion of the company team. Assume that you are wrong in how you've constructed it and modify it to achieve your desired results.

Lastly, iteratively roll out the proto-typed organization once you have achieved your desired results.