

Evolving Agile in the Enterprise: Implementing XP on a Grand Scale

Michael K. Spayd
Qwest Communications, Inc.
mspaysd@qwest.com

Abstract

How can XP or other agile methods be used in large corporate IT shops? One large company found out, making XP the official corporate software development methodology for all projects over an 18 month period and counting. This paper explores many facets of that experience, beginning with a summary of the eight key determinants of organizational change model. It then follows the organization in implementing XP, using the change model as a yardstick to assess the success of the hoped for change. The paper concludes with key learnings regarding how to implement an agile methodology within a large organization.

1. Introduction

In 2002, Qwest Communications, a large telecommunications company headquartered in Denver, CO, was forced to respond to dramatic changes in its market as it moved from what had been a stable, main street business to one of chaos, uncertainty, and constant challenge. Consolidation in the telecommunications industry coupled with ruthless competition had forced the change. The organization, driven to respond quickly and profitably to its new market situation, instituted an aggressive program of organizational changes.

In response, the IT department began a grand experiment championed by the CIO to convert the entire IT organization to the use of Extreme Programming (XP), ninety day system delivery cycles and the use of 3-day Offsite kickoffs for all software development projects.

A year later, what can be said by way of judging the success of this experiment? If we use the yardstick of faithfully applying each of the XP principles and practices in all development efforts, the experiment might be called a failure. Similarly, if we look for the faithful application of the eight success factors for

organizational change, the effort could appear to fall short. But if we judge the results in terms of helping a 5,000 member IT organization radically shift its thinking about what it means to frequently deliver valuable software, the results may be seen as somewhat astounding.

The organization has moved from one where year plus development efforts were the norm, to consistent 90-day deployment cycles across the enterprise. Along the way, scores of development teams adopted the XP practices of Pair Programming, User Stories, Iteration Planning, Stand-up Meetings and Sit Together. But this is to anticipate our ending.

In this paper, we examine Qwest's 2002 experience by taking a step back, interweaving two valuable perspectives to help us more effectively judge its success: the discipline of organizational change or transition management, and the guidelines of agile software development methodology. We focus on the partnership between these two perspectives, exploring how to most effectively accomplish system delivery (the methodology perspective) as well as how to facilitate organizational change in adopting a new development paradigm.

Part 2 lays the groundwork with regard to how organizations can successfully emerge from change, discussing both the nature of organizational change, eight determinants of success in making change happen and how organizational culture affects change. (It is assumed the reader is familiar with the basics of agile software development and so this is not covered.) Part 3 moves into an extended look at Qwest's experience adopting an agile approach followed by some observations. Finally, Part 4 provides some reflections and conclusions on the overall experience. Throughout, the author attempts a lighthearted approach, in tune with the zeitgeist of the agile community and to soften the rough edge of an otherwise critical voice.

2. How Organizations Change

How do organizations cope with change? We'll look at this important aspect of the Qwest 2002 experience via examples of organizations accommodating a new software development methodology.

2.1. The Nature of Change

Daryl Conner's book *Managing at the Speed of Change* [1] is a seminal work on organizational change. Conner discusses the nature of change, how and when it works, and what leaders need to do to ensure successful change in organizations. Other important work on change includes John Kotter's *Leading Change* [2], and William Bridges *Managing Transitions* [3].

2.2. Eight Determinants of Organizational Change

Based on the work of Conner, Kotter, Bridges, and others [4], as well as fifteen years of practical experience implementing change initiatives for a variety of organizations, we can now think about an organizational change framework distilled into eight critical success factors. We'll call these our Eight Determinants of Organizational Change.

Any change initiative has strengths and weaknesses in terms of the amount of focus and clarity with which it addresses each of these eight determinants. Yet clearly, each factor must be managed to some degree (at threshold levels) for any major change to succeed. The following diagram depicts the relationship among the factors as a circle of interlocking arrows, with *Sponsorship* in the middle to depict its centrality to the entire effort. The order of the factors is somewhat arbitrary, but we'll examine them here in a fashion that suggests which factors tend to be engaged first.

“Obtain committed executive sponsorship”

Leadership in the form of executive sponsorship is probably the single most important ingredient in major change initiatives. Large-scale adoption of a new methodology, for instance, must have active sponsorship from the CIO or higher (that might come as a shock to really technical people who may not consider there is someone higher than the CIO). Sponsorship—not just support—is required. The ultimate act of leadership commitment in fact would be for the sponsor to put his or her reputation—at least in part—on the line with regard to success of the change (i.e. we succeed or I resign). A clear challenge in implementing a new methodology! You may ask, “Why must a CIO consider a methodology

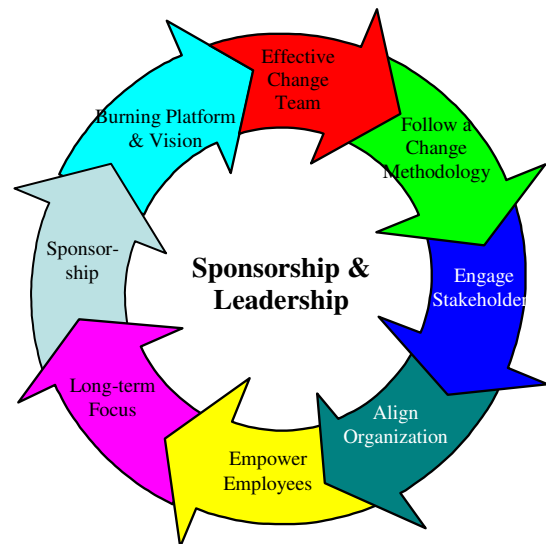
change to be such a fundamental, central focus of his/her agenda?” Well, experience shows that without this level of avid CIO commitment within a large company, real change (as opposed to apparent or superficial change) is quite unlikely.

“Articulate the ‘burning platform’ that demands change and the vision of where you’re going”

The concept of a “burning platform” was originally conceived by Daryl Conner [1] after having watched an interview with an oil-rig worker who jumped some twenty stories from a burning oil platform into the frigid ocean in order to save his life. (Now that's a burning platform!) The choice for the oil-rig worker was clear: likely death vs. certain death.

Executives must take a similar “death leap” stance with regard to their change initiatives. They cannot afford to spend their time and political capital implementing initiatives that seem like “good ideas.” Because organizations do not have the luxury to implement anything less than genuine business imperatives, employees must believe that they've met their “burning platform” decision point: that is, there is no choice other than making the changes associated with the new methodology. Without this executive-injected sense of urgency inertia will typically prevail, employees will balk, and any change initiative, no matter how valuable, will fail. Burning platforms may take the form of poor financial results, very unhappy customers, a poor economy, or consistent failure with system delivery projects. Meanwhile, just screaming at the team does not typically create a good burning platform.

Success Factors of Change



“Form and maintain a highly effective change team”

The change team (often simply called the project team) is the core of a change effort. The first concern is choosing the right people – if the project is important to the company, the people chosen should be the best around. Next, the team must be able to function as a team (calling a group of people a team does not make them one). Central issues here are communication and decision-making. Finally, the team must be accountable. The most powerful “team building” activity for such a group is to define a clear purpose and challenging performance goal for which they are held mutually accountable (this is good news for executives—don’t take them to expensive ropes courses or rafting trips, just tell them you want them to do the impossible!).

“Adhere to key project and change management practices”

Adopting and adhering to a proven set of implementation practices (a change methodology) is essential to success. This includes how a methodology is developed, trained, and rolled out. Monitoring project metrics, managing project scope, and understanding the dynamics of change are all important. The eight determinants of change are the framework for such a methodology.

“Keep stakeholders engaged throughout the life of the project”

A Stakeholder can be anyone affected by the change. In the case of a new methodology, this includes developers, testers, analysts, customers, managers and even executives. The team implementing or shepherding the change may not have time to focus equally on them all, so prioritization is in order (think how XP User Stories must be prioritized, and how priorities change as the project unfolds).

Key Stakeholders are those Stakeholders who are both significantly affected by the change and whose support is critical to adoption. Let’s consider Executives, Middle Managers and non-management “opinion leaders.”

Not surprisingly, Executive Stakeholders, while rarely overlooked, are often mismanaged in a variety of ways. For instance, they may be promised the impossible or misperceived as requiring outcomes that in fact are not that important to them. Meanwhile, Middle Managers are frequently ignored entirely, which will almost certainly doom any change initiative since they are the key constituency to affect organizational change. This group directly

controls key resources, including the best people, jobs, and work practices. Moving from a rigorous to an agile methodology may alter or diminish the power balance within Middle Management, so effective communication and buy-in is vital.

Now consider the non-management “opinion leaders”. These may include influential developers or architects or other highly impacted roles such as testers, support groups, and even Corporate or IT Finance (you know, the people who authorize capital projects in big companies). Stakeholders must be sought out for advice, stroked on the golf course (or pub) and cultivated as allies. Frequent communication, assessment of their position vis-à-vis key issues, and gaining their buy-in (or at times merely their begrudging neutrality) are all essential.

“Align organizational elements and neutralize politics”

As a once famous consultant (the reengineering “guru”) was fond of saying: “Any change project has a good chance of success as long as the number of VPs remains constant.” [5] Well said! (Too bad he didn’t understand all of the eight determinants of organizational change, he might have been more successful!) We must ensure that existing HR practices, work design and organizational structure and policies do not undermine the methodology changes being implemented. For instance, do HR policies allow for a team compensation policy, one that could support a team-oriented methodology? Rocket science, right? Or, can people who suddenly find themselves in roles that are now considered part of the customer team be moved out of IT to the business organization? You might be surprised how difficult this is (but perhaps only if you don’t work in a big organization). Such issues may be seen as out of the control of the change team. Perhaps they are; perhaps innovative thinking can help. In any case, failure to mitigate these types of risk creates cynicism and frustration and undermines the change effort.

Counteracting the effects of toxic organizational politics is likewise crucial. For instance, if jealousy and bad blood exist between the development organization and the infrastructure organization, they are unlikely to “play well together” under any circumstances, much less working together to implement a new methodology. An organizational readiness assessment can uncover some of these issues. Similarly, devising an effective deployment strategy can counteract some of the deleterious effects. (Working on the dynamics within the senior management team is also an option, though arguably more difficult.)

“Empower affected employees in adopting the new system”

Supporting employees in changing includes training, communication, and job definition. Excellent training is too often perceived as a luxury, but consider the results of not training people: confusion, inability to perform properly, disempowerment. Timely, frequent and multi-way communication is also essential in making employees allies of the change. Third, designing and filling jobs that support the new process is critical, but frequently overlooked. For instance, if you won't need many requirements analysts with XP, how will you help these folks make the transition? And programmers will likely have to become more user-centric, taking on more traditional analyst roles in the new XP world. Are your developers capable of making this transition? People generally don't like it when the people around them fear losing their jobs, even if that is not what happens. Such an environment creates fear, rampant speculation and slows down the change process.

“Do not lose focus”

One of the greatest dangers in implementing change is the premature declaration of victory, typically based on superficial indicators such as everyone has been trained (almost always a lie, by the way) or everyone seems to be talking the new lingo (important, but clearly not sufficient). Sponsors—and the organizations they lead—cannot give up after 6 months, or 1 year (which is coincidental with the annual budget cycle), or even 2-3 years. Large-scale change frequently takes a long time. Change is a marathon, not a sprint. As with running a marathon you must pace yourself, train well and load up on carbs (e.g., budget) beforehand.

2.3. Organizational Culture

In implementing a new methodology, organization culture is important to consider as an overlaying factor to the eight determinants of change. In the book *Built to Last*, Jim Collins and Jerry Porras [6] establish empirically the critical importance of organizational culture in leading companies to success. William Schneider in *The Reengineering Alternative* [7] articulates four types of organization culture: Competence, Control, Collaboration and Cultivation. Schneider observes that similar to people and their Meyers-Briggs profiles, organizations frequently have different levels of emphasis in each

of the four types, but generally there is a predominance in one.¹

As Jim Highsmith points out, “a particular culture is not necessarily change tolerant or change resistant—but it may resist certain types of change and embrace others,” [8]. The reader is referred to Highsmith's review of the four cultures in regards to software methodology, or to the original material in Schneider. A brief recap of these culture types follows.

The core value in a Competence culture is achievement, so everything is set up around becoming better, organizationally and individually. Training, technical excellence and competition are emphasized (think Microsoft). In terms of software methodologies, there is a natural tendency to employ fact-based measurement and to become process-centric. The Control culture, by contrast, is centered around getting and staying in control (similar to a military organization). Control here is not used in the power-based sense, but rather in creating and sticking with a plan (controlling the variables to avoid surprises). Heavily documented processes with rigid rules and controls are therefore favored by such cultures, as well as forms that help resist the destabilizing quality of change (e.g., rigid change control procedures). No hint needed here. The core need in a Collaboration culture is for affiliation, so teams are a key method for organizing. Consensus is the preferred form of decision-making and collaboration practices find favor. This type of culture is most likely the easiest place to implement an agile methodology. The Cultivation culture is oriented towards individual members achieving their own full potential through the help of the organization. Not surprisingly, this is relatively rare within for-profit businesses.

In summary, there are many complex factors affecting change, most of which few teams have expertise in, nor are they driven to pay attention to these issues. Understanding the conditions for successful change to occur, we turn now to a specific example covering almost two years.

3. Unconscious Evolution towards the eXtreme

Part 3 begins an extended case study regarding implementing XP within Qwest, a large Fortune 500

¹ Schneider has developed a survey-based method for measuring the degree a given organization conforms to each of the ideal culture types described.

business enterprise. Themes explored in this part include how evolutionary organizational transformation occurs, whether it is possible to force the adoption of a new process paradigm, and discovering whether and how XP scales to being used in a complex, legacy-oriented environment.

(Authors note: I take a critical eye—though hopefully a fair one—to the success of the Qwest implementation. Full disclosure: I was part of the original team responsible for implementing XP, so my critique should first and foremost be taken to concern my own performance (*cf.*, it's Chet's fault). Currently I am engaged in the second phase of the experiment: evolving XP into an enterprise scale agile methodology approach.)

3.1. Organizational Background

A brief summary of our organization will provide useful background to explore our themes. Qwest is a 55,000 person company that delivers a broad range of telecommunications services including local business and residential access primarily within a 14 state western region, international long distance, broad band, and other telecommunications services. Qwest's IT organization alone employs more than 5,000 people spread across the United States, but primarily located in six cities and twenty some buildings.

At the beginning of implementing XP, the organizational culture was really two rather distinct cultures: the predominant one was a control-oriented culture (a pre-merger Regional Bell Operating Company, or RBOC), the other a smaller entrepreneurial group (a medium-sized long distance company) who had a mixed competence and collaboration culture. The control culture was enamored of Rigorous Software Methodologies (RSMs) with highly documented process requirements and long waterfall delivery cycles. The original Qwest culture had a more ad hoc delivery process, but with a strong emphasis on change control. The two did not mix all that well.

Long delivery cycles of a year or more were largely the rule, some ending with nothing more to show than documentation. These "documentation only" projects may have been considered a disappointment, but after all only to be expected since "customers never know what they want." The excuse "they kept changing their mind" became an easy shield to protect the executive or manager in charge from meaningful sanction, despite delivering nothing of business value. Business case projections were rarely (never is a strong word) followed up on to determine if the originally projected ROI was achieved.

The technology environment ranged from massive, non-integrated COBOL billing systems to web-based applications written in JAVA or C++, with quite literally everything in between. Each application, some massively dependent upon interfaces from other systems, was managed by a director whose managers, developers, analysts and testers typically used their own development process and often stayed with the same organization for many (many!) years. This environment can be characterized, as Carly Fiorina did of HP, as consisting of a thousand tribes (cited in [9]), each with their own tribal chief and their own way of life. The RBOC organization had been assessed at CMM Level 2, which permits each project to have their own clearly documented software process, but with no requirement for a coordinated process or metrics across the corporation. CMM Level 2 was achieved through very strong (forceful!) executive sponsorship, but with no clear business driver. Despite the lack of continued emphasis on CMM assessments, such rigorous processes were prevalent, if inconsistent, within the organization when XP was introduced.

3.2. The Evolving Methodology

Given that the IT organization had resolved to take on change with a heartfelt embrace, and given that it was the de facto software delivery arm of the entire corporation, one of the first orders of business was to evaluate its software methodology approach and its associated delivery mechanisms. A quick but painful perusal of the process revealed that software was taking too long to develop, too long to deploy, and providing too little assurance that what was either developed or deployed was really going to ferry the corporation through the tumultuous waters of its marketplace.

Recognizing a clear need for change, several options were possible:

- Institute more rigorous controls over all aspects of the software delivery process
- Bring in smarter people who would just deliver the best systems for the corporation
- Rely on inspiring each IT professional to cultivate a sense of change and community for success
- Move to a more agile and collaborative approach that would stress results over tasks

Rigor Begets Rigor

In evaluating the appropriateness of a more rigorous development methodology, we as an IT organization

had to face some hard facts. We had heeded the “Rigor” call in the past when “Quality” and “Quality Assurance” were the watchwords of the day, but thought that adding more rigor wasn’t really going to help in this new marketplace. No one wanted to discount the contributions that CMM and ISO 9000 based approaches had provided; they brought form and stability and a sense of familiarity for the 5,000 people implementing them. People knew what their job was, their role on a project, their tasks day-to-day, and their measures of success.

The problem was people also knew how to circumvent the reams of paperwork in order to “just get it done” while appeasing the folks who felt they were adding value when they created 17 more forms for a particular project process. And while our methodology weight was indeed a bit heavy, the organization could support it and could continually have a sense of accomplishment around it, having specific forms and documents to complete during any stage of the development. Additionally, it provided comfort to the masses of new personnel being hired who needed direction for a software delivery process.

But we felt that the comfort and familiarity engendered in these approaches couldn’t promote the nimble delivery now needed by the transforming organization.

This option was rejected.

I Can Do Anything Better than You Can

Interestingly, we did consider an approach of simply bringing in smarter people (strategic hires) who would, by their mere presence bring in a new model of successful development, delivery, and deployment. With 5,000 people in place and innumerable legacy systems to evolve, however, this plan seemed implausibly small in scope.

This option too was rejected.

Collaborative Tiger, Hidden Piranha – The Qwest Agile Decision

Ultimately, we resolved that the way to speed up both systems delivery and recognition of business value was to take up the agile battle cry. We decided to groom highly collaborative teams around fixed timeboxes (our infamous 90-day delivery cycle) so that we could “fail fast” to converge on better and faster systems delivery. Our guides through this fast agility were to be expert teams of XP Coaches, known as “The Piranha Team.”

The next section will lead us through what we may now see as our parable of methodology success.

3.3. The Methodology in Action: The Parable of the Six Blind Men

You likely know the story of the six blind men and their differing views of the elephant. Here, the idea is that there existed a variety of perspectives on what XP really was depending on to which of the thousand tribes you spoke. We will explore several of those perspectives. Clearly none of these views is complete—as with the elephant. In our case they may not have even necessarily been “true” in any meaningful sense, but each was certainly real for at least certain members of the tribal community.

View 1: XP is Offsites

The first view of the XP elephant is that XP is really just Offsites. So, what’s an Offsite? Did Kent describe them, or was it added on later?

It turns out, nobody in XP-land ever heard of Offsites except those at Qwest. Offsites are three to four day intensive software development sessions where users work together closely with developers and testers to create working prototypes for a significant piece of new system functionality. The original Offsites were brought to Qwest by the CIO, who basically helped invent them at another company.

The Piranha team, in consultation with a leading XP coaching firm, created an enterprise XP rollout strategy using “immersive” style training and coaching sessions to wrapper the prototyping efforts. Thus, the XP Offsite was born. Four days with a 7:00 am to 9:00 pm schedule, and barely a break for meals! Not exactly a sustainable pace, but they did deliver results. More than one team went through significant transformations, both in terms of their understanding of XP and in their ability to work together as a team. Customers tended to be of two types: those that didn’t really have a vision for the system—who tended to get confused and frustrated during the Offsite. The other type were those that had a clear vision for the system. They loved XP. These customers got into the driver’s seat and started driving the project like only a good customer can. Developers loved it too, even pair programming (after the initial resistance of course).

The difficulty was that XP is not just Offsites. Delivering a system takes at least 90 long days of well-coordinated effort of a whole team. And guess what? You can’t have any of these well-trained XP coaches who worked with you during your Offsite, because they have another group coming into an Offsite next week! Sorry.

View 2: XP is 90-day delivery

Others believed that XP was simply delivering software in 90-days. (Well, that wouldn't be so bad, would it?) The managers of such teams sometimes invoked a death march strategy to ensure rapid delivery, since not all teams were good at scope management in the XP fashion. Or co-location. Or many of the other ten or so practices.

There was tremendous pressure from senior management to deliver rapidly, and to not push back too hard on the customer's preferred scope, whatever the estimates or velocity tracking might indicate. As always, some leaders did not betray the spirit of XP, while others did. In general, the 90-day delivery mandate became a tremendous organizational pressure cooker that forced the rethinking of software development approaches, regardless of whether the team believed in XP as the way to solve their dilemma. Change ensued. This was good.

View 3: XP is the Right Methodology

Some teams had great success. The developers adjusted to programming in pairs, to frequent integration and rapid delivery. (As a note, very few teams—with some notable exceptions—ever seemed to adapt to test-first development. Writing unit tests before writing code seems to be a talent few people can easily learn, even though it makes great sense.) Customers adapted to driving the iteration and writing stories. Testers worked in partnership with customers and did pair testing.

One VP in the business community was an avid fan of XP. "I will never go back to developing software any other way," he told an internal IT leadership conference. He was not alone. For such practitioners, XP was a saving grace. It saved them from hide-bound rigorous methodologies where most of what was delivered was paper. In the past, analysts had frequently been a barrier to customers sitting down and talking directly to developers about what they wanted. A great sense of shared appreciation was cultivated in such teams between developers and customers. Both saw how hard the other's job really was, and neither wanted to switch. For this type of team, XP was the one right methodology—couldn't the rest of the organization just come to their senses and see the truth?

View 4: XP is Chaotic

This view of XP was often espoused by people who worked on legacy systems and/or applications with multiple dependencies on other applications and/or

who did not attend an XP Offsite (but did tune into the rumor and innuendo hotline.)

There was some real truth to this view. What happens when five project teams all start two or so weeks apart, each with 90-day delivery mandates, but all having functionality dependent on the other teams and each needing to go through end-to-end testing together prior to deployment? Chaos.

Or, what happens when the business or requirements analysts that have always worked in the IT organization are suddenly asked to be part of the customer team and not included in the IT director's resource plan? How about "stand-up" meetings that last more than an hour, over multiple sites, on the phone, with primarily project managers, managers and others who don't touch code participating?

We must remember that this implementation potentially effected hundreds of projects and thousands of people, only a few hundred of whom ever went through the official training program—an XP Offsite. This type of complexity was not managed systematically, but rather evolutionarily. Fail fast. They did, then they reverted to tried and true ways.

View 5: XP is Irrelevant

For some, XP was simply irrelevant. Working on a maintenance effort, or on small enhancements to mainframe applications where the customer was involved in only the briefest and most peripheral way (and rightly so), XP was a lot of hoopla with little payoff or meaning. When you are trying to get an entire organization to change its stripes, however, every nail (or tack or screw) seems to call forth the same hammer. Differentiating when a given tool is required—and when it is not—tends to be at a later stage of learning evolution. Such teams went quietly along—below the radar—delivering as in the past.

View 6: XP is Evolving

Truth be told, we really weren't implementing XP. Rather, it was "XP^e". The extra 'e' originally stood for evolutionary, later evolving into meaning XP for the enterprise. XP Offsites were clearly one of the key marks of this evolution, as were mandated 90-day delivery cycles. For an organization oriented towards reporting financial results quarterly, 90-day cycles clearly made sense.

Other evolutionary innovations included pair programming across distance (using Net Meeting), stand-up meetings over the phone (everyone is required to stand, even when in their car!), and user stories documented (briefly) in a requirements analysis tool, a vestige of former rigorous days.

Teams that took the view that XP was evolving, that they would practice local adaptation, tended to fare quite well. Moving past the rather dogmatic rhetoric of the Piranha team, they figured things out for themselves, connecting with the spirit of XP and doing what they needed to do to deliver software in a large, complex IT organization with multiple customers, dependencies, needs, leaders and methodologies.

4. Reflections on our Evolution

Given each of the six blind men's views of our 2002 XP implementation, what is the real truth? What can we learn from having carried out this experiment about how to implement a new methodology and about XP? We need to explore whether and how change really occurred within Qwest. Further, we will reflect on what lessons we can take from this adventure to implement such a massive methodology change in other organizations.

4.1. Did the Thousand Tribes Change?

It is no mean feat to get 5,000 people, divided into highly variegated tribal groupings (most of whom were quite comfortable with rigorous methodologies that synched well with their control culture) to adopt the collaboration driven XP methodology. It might even be described as nearly impossible. For the new CIO of Qwest, who loved to attempt the impossible, such a challenge was fantastic. (To be honest, the CIO was driven less by wanting the organization to adopt XP than by wanting faster delivery cycles and stronger collaboration with the customer. XP was the obvious industry trend to carry that message.)

Such CIO, top-down driven change represents excellent executive sponsorship—a singularly necessary, but not sufficient one of our eight change factors. So, while the difficulties in implementing XP practices in Qwest were many, lack of executive sponsorship was not one of them. Overall grade for Sponsorship: A-.

As an aside on top-down driven change, XP and most other agile methods typically have a significant component of bottom-up motivation (collaboration, remember). In fact, most often XP appears to be introduced one project at a time, by developers who tell other developers, who convince management to support them in using XP, or to at least look the other way. This was attempted in some small way at Qwest through inviting Kent Beck in for a consultation to drum up support and to consult on how to bring this

methodology into a large organization. Regrettably, the team found that little was known by the XP community (including Kent) about how to work in a big company. We were pioneers and would have to figure it out ourselves. Rats. (We rationalized that one day our experience might turn into big consulting fees!)

The second change determinant is the burning platform and vision. The burning platform articulated by the Qwest CIO was the failure of the organization to deliver software reliably and to satisfy the internal customer. The critique hit home, but also ruffled feathers in the way the message was articulated. The vision outlined was that of an IT organization that worked very closely with their customer; delivered prototype software in 3-days during an initial Offsite; reacted to the customer's changing requirements without flinching; then delivered deployable production software every 90-days. Wow! Overall grade for articulating the Burning Platform and Vision: B-.

The team nominated to implement the change was a self-identified elite group of developers who had some homegrown experience implementing XP without management support. The Piranhas. Perfect. The team was very familiar with object-oriented programming and the tenets of XP. Also perfect. Unfortunately, they also had no experience as trainers, in project management disciplines or with organizational change. Rats. From an outside view, the team was clearly lacking some key expertise, but was otherwise a very good team. Additional fallout resulting from the choice of the team was that XP was implemented in “true believer” fashion—with a rigidly dogmatic tone that helped some “connect” to the power of the ideas, while alienating many others. Overall grade for Effective Change Team: B-.

The team's background as developers unfamiliar with implementing change made following a change or other implementation methodology a low priority (some darn good user stories never get coded, either). The team-scaling strategy to drive implementation was to train 50-plus XP coaches by bringing them onto the team for a 90-day rotation, have them lead XP Offsites, coach 90-day projects, then turn these trained coaches loose to infiltrate the organization and create change. Overall grade for Following an Implementation Methodology: C-.

Perhaps the most serious shortcoming from an organizational change view was the lack of engagement of a Key Stakeholder constituency: Middle Management. Since the only training mechanism for IT employees was their attendance at XP Offsites, many middle managers were cut out off

from knowing what was happening, or why, and how they should respond. Further, there was no written documentation on the process, a key method of both control and understanding amongst the Middle Management ranks within the Qwest culture. This group was never actively involved or talked to, thus creating alienation and ultimately a form of management undermining, sometimes intentional, other times merely out of ignorance. Overall grade for Engaging Stakeholders: D.

Aligning the rest of the organization with the change was also significantly problematical. Due to HR practices, there was no ability to meaningfully compensate the team as a whole—a possible “fatal flaw” in such a team-oriented endeavor as XP. Perhaps an even greater problem, however, was the lack of budget to create the proper “pod environments” to allow XP teams to co-locate. At first, small amounts of money were creatively used to redo the space on a few floors, resulting in several teams moving into pods. Some additional teams took matters into their own hands, taking down and rearranging cubicle walls in their preferred format. (The Piranhas even gave Allen wrenches as awards at Offsites.) Unfortunately, an incident or two where the power was taken down in adjoining cubes got the real estate organization involved and the self-created pods were shut down. As a result, pod space was unavailable for all but a few early adopter teams. Overall grade for Align Organization: C-

As mentioned, there was no meaningful training in XP except for those projects that went to Offsites. Even though some fifty or more Offsites were held, this represented perhaps 30% or less of all project teams. In order to empower employees, training is key; its lack prevented the majority of employees from truly understanding XP, let alone using it effectively. Training certainly could have been done “on-the-job” by experienced XP coaches, but there were virtually none at Qwest available to work with project teams. Offsites became the only form of such training and it was clearly inadequate to drive large-scale change.

An additional problem was the unclear way with which changes in roles and responsibilities were articulated. The message that seemed to be heard by the organization was that developers and customers were king. Project managers and analysts in particular were judged to be “not long for the work world” within the Qwest environment. Not good. In addition, the shift in responsibility of the business analyst role from IT to the business organizations created confusion and organizational responsibility

gaps. Again, not good. Overall grade for Empower Employees: C-

Taking a long-term focus is one of the most difficult lessons that organizations need to learn with regard to change. Qwest was no exception. As with many change initiatives, premature success was declared in implementing XP within Qwest. Vice Presidents and everyone else were talking about user stories, stand-ups, and release planning, while perhaps only 10% of teams were meaningfully using XP practices to a considerable degree. To their credit, IT leadership eventually recognized that success had not been achieved and took steps to refocus the organization. Overall grade for Long-term Focus: C

What is our chief takeaway from this analysis of Qwest’s methodology change? Organizations do not change merely because the boss says so, at least not in the way that is intended.

Even though we didn’t do everything right to ensure successful change, changes did occur. There was no question that the organization was delivering software more frequently. It was debatable whether the software was of higher quality or whether more software was coming out. Still, release cycles were shorter, and customers were much more involved than in the past. In fact, customers generally loved the new approach. In the CIO’s address to 200 members of IT leadership at the end of 2002, he stated, “The business likes XP better than you do.” He was probably right.

4.2. Turning Point: Breakdown and Renewal

The term “malicious compliance” is one used in the field of change management to denote circumstances where change has occurred on a superficial level (the letter of the law) but not bought into by the people that matter. Much of Qwest IT was in such a state. Many project teams tried to circumvent Offsites and the practices of XP while giving lip service to trendy XP terminology. The feedback to senior management was quite clear: XP^c was broke and needed fixing. What was the problem? Employee feedback was that:

- Not all projects are the same and can use XP as it was written or communicated at Qwest
- XP is only a development process, not an end-to-end system delivery methodology (which is what a big company like Qwest requires)
- Project team roles are confused, especially the role of project manager and other management personnel

There were other, much more specific pieces of feedback, but the themes above were both consistent and prevalent. This outpouring of feedback led to the formation of a new methodology effort that attempted to redress these problems and is summarized below.

4.3. Lessons Learned

How can we summarize the lessons that this experiment in organizational change can provide us? There are five:

- 1) Top-down driven change is likely to be only partially successful with an agile methodology. Top-down change worked well in implementing CMM within Qwest because CMM is consistent with a control culture orientation. Agile, on the other hand, is a collaboration-oriented approach and therefore requires grassroots level buy-in.
- 2) Full XP does not scale well to an enterprise level—it is too inherently rigid both because it is “extreme” and because its practices are highly interdependent. Such an approach does not lend itself well to the large-scale customization required in a complex enterprise. XP practices could conceivably become the inspiration for a flexibly implemented agile development approach, however.
- 3) If a methodology implementation does not have significant buy-in from Middle Management, it will almost certainly fail. Middle Management is critical in driving the day-to-day behaviors within a large company that embody change. If they are not strong advocates change will not occur.
- 4) Creating a rigid performance benchmark—in our case a 90-day delivery cycle—forces organizations to change. The problem is that the organization may not change in all the ways you want it to, or even in the most desirable ones. In Qwest’s case, there were shorter delivery cycles with more customer feedback and involvement. However, there was not widespread use of a significant portion of what most think of as XP.
- 5) Implementing XP without coaching will not work unless the team environment is already very favorable, such as having developers who have already used XP effectively and others who really want to try it.

4.4. Looking Forward: The Evolution to an Agile Methodology Framework

Moving forward, the new methodology team (dubbed Q.EXE for Qwest’s Enterprise XP End-to-End)

wished to employ an evolving, more sophisticated approach that would be adaptive to the variety of projects within Qwest while still keeping a strong XP influence on development practices. The team was heavily influenced by the work of Alistair Cockburn [10] and Jim Highsmith [8] in formulating this thinking.

The new approach emphasizes our analysis of what was left out: XP^e had insufficient methodology scope (e.g., it was not an end-to-end delivery process including intake, deployment readiness, and deployment); XP^e was not implementable consistently across IT; XP^e was not adaptive to different project profiles; and XP^e did not include practices for IT program governance or project management.

The architecture of the new methodology includes a required component (principles and outcomes) and an adaptable component (practices, roles, deliverables). The adaptability dimension pays attention to the project’s profile in a way that is very similar to Cockburn’s Crystal methods [9]. Implementation is underway and the results could become the subject at a future conference. Currently the feedback from employees is hopeful.

5. Conclusion

The willingness of Qwest to embark on such a bold experiment, be open to critique (feedback), learn from its failures and to allow others to hear about them (communication) and to adapt quickly to that learning should be appreciated and applauded. The most significant learning the reader might take from Qwest’s example regards that most challenging of XP’s values—courage. We at Qwest IT attempt to embody that value in our new spirit of service.

Acknowledgements

The author wishes to thank the following people who contributed to this paper: to Bill Quirk and others on the Piranha team for their diligence; to Bob Morgan and Qwest IT for their support, and to Rebecca Wirfs-Brock for her guidance. Most especially I wish to thank my colleague Jean Tabaka for her considerable patience, strong editing and her unwavering and kind advice throughout six months of writing that helped me stay true to my own voice. Thanks all.

References

- [1] Conner, Daryl, *Managing at the Speed of Change*. Random House, New York, 1992.
- [2] Kotter, John, *Leading Change*. The Free Press, New York, 1995.
- [3] Bridges, William, *Managing Transitions*, 2002.

- [4] Unpublished work done at General Electric and StorageTek on how organizations change.
- [5] Michael Hammer, address at technology conference, Boston, MA, June 1993
- [6] Collins, James C. and Jerry I Porras, *Built to Last*. Harper Business, New York, 1994.
- [7] Schneider, William, *The Reengineering Alternative*. Irwin Professional Publishing, Burr Ridge, IL, 1994.
- [8] Highsmith, Jim, *Agile Software Development Ecosystems*, Addison-Wesley, Boston, 2002
- [9] Lissak, Robin and George Bailey, *A Thousand Tribes: How technology unites people in great companies*, PricewaterhouseCoopers, New York, 2002
- [10] Cockburn, Alistair, *Agile Software Development*, Addison-Wesley, Boston, 2002.