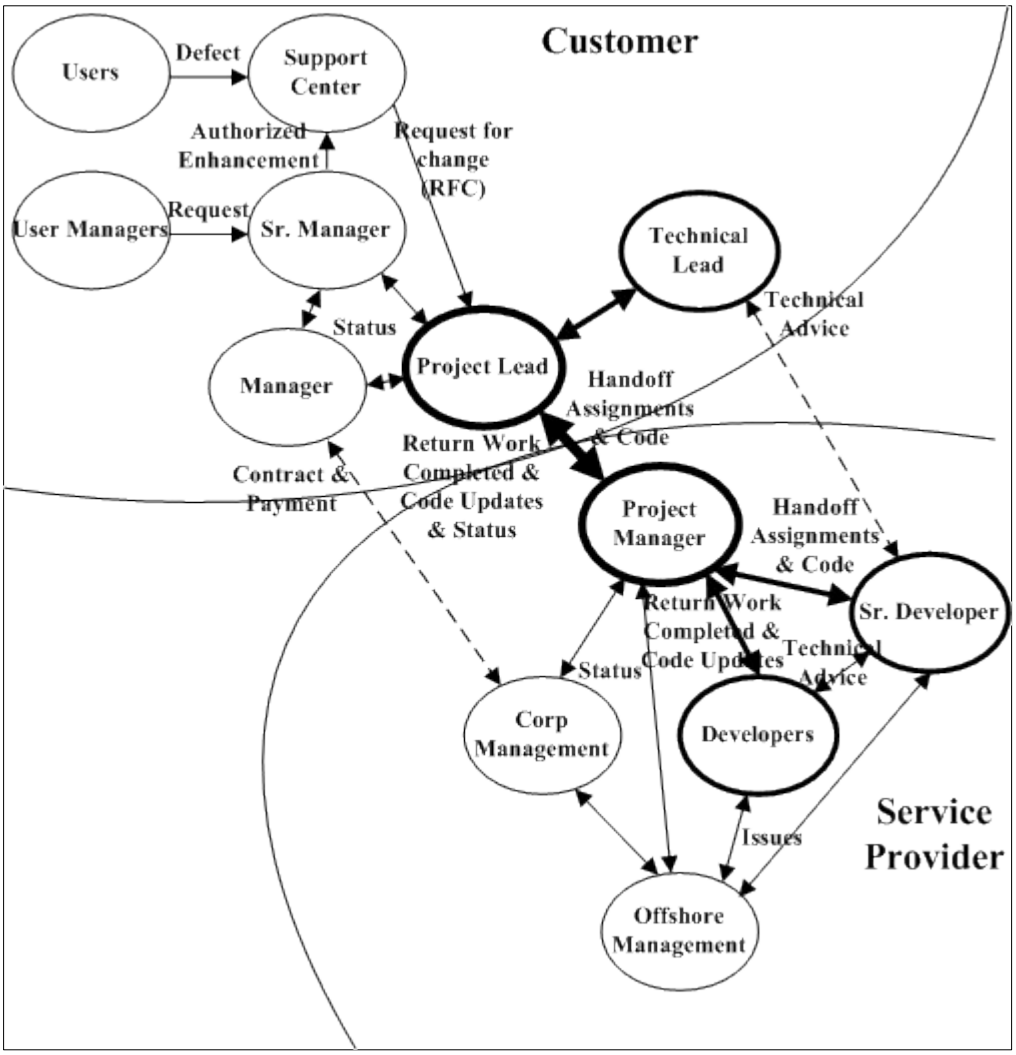
Workplace Ecology: A Case Study in Project Management

By Robert E. Perrine.

Smashwords Edition.



Acknowledgements

I want to thank all of the friends I have worked with over the years who helped me learn about project management and about people. Special thanks go to Brian, Franklyn, Jim and Tom for their support and encouragement while I was working on this book.

Disclaimer

This is a work of fiction. Any resemblance between characters in this story and people in real life is purely a coincidence.

Table of Contents

Introduction

It seems to me that most books written about project management are tall tales. I read about how project managers use their skills and expertise and accomplish great things. In my opinion, most of those stories are fables. I have taught a lot of project management classes. I learned to separate the theory of project management from the reality. My goal in this story is to help you see the reality. The naked truth is that lots of projects fail. Within Information Technology (IT) the number of failed projects exceeds the number of successful projects. Those failures are then typically attributed to technology. I think that denies the truth in the matter. Most IT projects fail because of people.

When I did project management on civil engineering projects I learned that there were vast books of rules that govern how buildings are built, streets are paved and projects are approved. Very little of that applies to IT. So, as you read this story, look for the ways that people are twisting and manipulating the goals of the project. Also pay attention to the way the people who are supposed to be supporting and sponsoring this project often sabotage the effort. I hope that this story helps you understand what project management in information technology actually looks like.

The events in this story are loosely based on a real project. All of the characters and project details have been fictionalized. But the core of the story is intact. I hope you enjoy this story.

Initiating

Here am I back in the consulting game again. The original project that Greg had for me was a business process re-engineering assignment. Then he found this tiny little project that has the potential to become a really great opportunity. My assignment is to go in, take care of this little project, implement best practices and then help grow the opportunity so that we can enlarge the contract. This is typical of Greg's approach to business. He likes to deliver superior customer service and then keep growing the opening one or two people at a time.

Some companies focus on marketing, some focus on engineering and some focus on service. The stereotype of a sales person is someone that promises more than they can deliver. The stereotype for software companies is that they focus on the technology - using the latest and greatest tools. Greg's company focuses on service. We deliver what we say we will deliver. And the company pulls together to make that happen.

As is typical of most projects, the initial negotiations took place while the project manager - that is me - was busy elsewhere. The account manager for this customer found an opening, put a proposal together and then worked with Greg to package a winning deal. Once the contract was signed they called me. My escalation point on this project will be Vijay. Vijay is the technical manager responsible for the programmers on several accounts, including this one. Greg gave me three goals:

Make this customer happy.

Grow our position in this customer site.

Use this little project to set up best practices that we can apply to other projects.

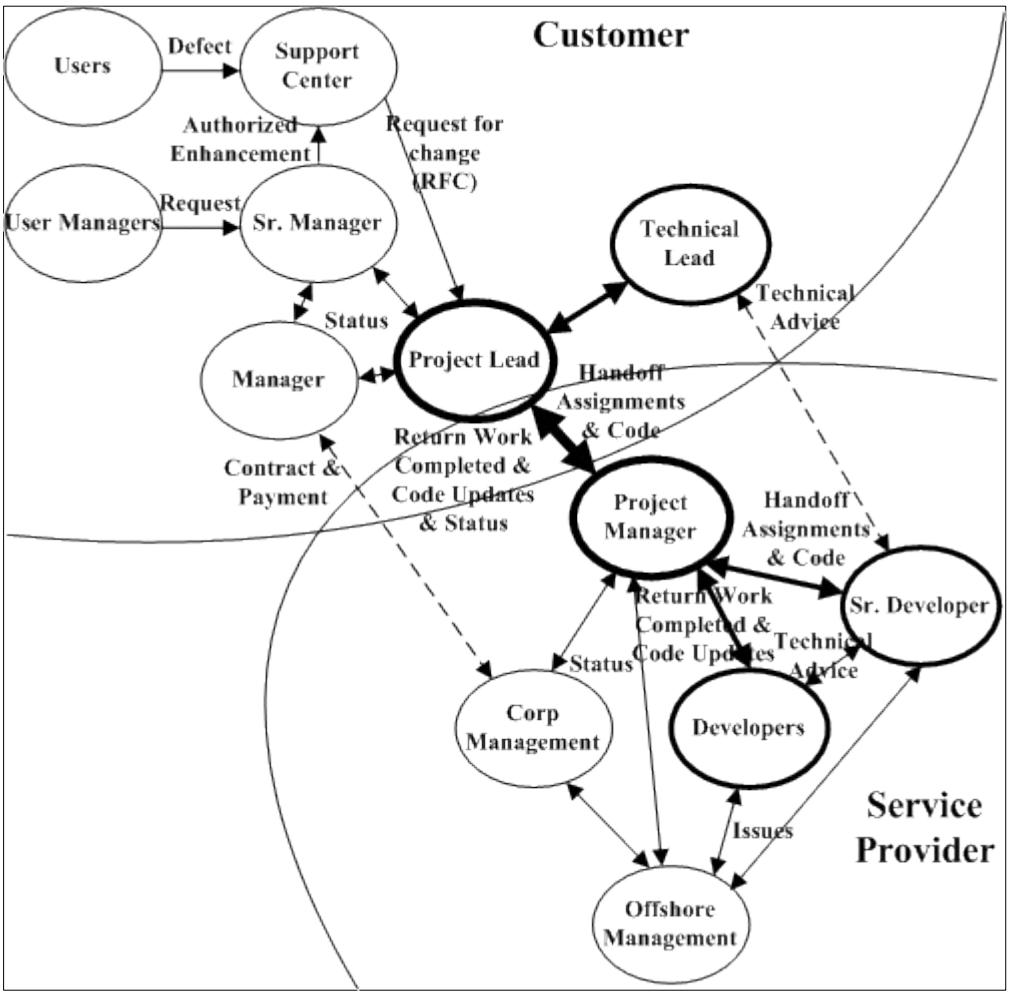
The goals set by the customer are to use contractors to temporarily augment the headcount in a small department. The contract says we will be developing new applications and supporting existing applications. Like most Japanese owned companies, this customer keeps a very tight grip on finances. There are rumors of an upcoming downturn in sales. Hiring a few consultants now will help get some work done. Then, if the downturn materializes, they will cut our contract and reduce their costs. If, instead, they were to hire permanent employees then it would be more difficult to cut costs on short notice. As Lax and Sebenius note in their book on negotiation, the essence of agreement is the perception of differences. (David A. Lax and James K. Sebenius) We are betting that the downturn will not impact this company and that will allow us to not only complete this contract, but to expand from there. They are betting that the downturn might impact their bottom line and thus they are willing to pay a premium for service now bundled with the option to react quickly. This is a win-win agreement.

Planning

The first two programmers that Vijay hired started work today in the offshore location. They are both experts in the programming language specified in the contract. We had a conference call and got to know each other. The key result was agreement on an organizational structure. Vijay will be their technical manager. I will be their project manager. If the team encounters technical issues, then Vijay will either solve them himself or borrow other resources from other projects to find a resolution. My job is to communicate. I need to communicate with the team and I need to communicate with the customer.

One of the lessons that I learned from the prior case studies is that the organizational structure does not always align with the flow of communication. Greg wants me to implement best practices on this project. Well one of the things that I learned from prior case studies is that defining the structure for the communication flow is just as important as is defining the structure for the organization. So the next task I undertook was to prepare a communications plan for this

project. In that document I listed the expected meeting schedule and the expected delivery dates for the status reports. I also included a diagram, shown below, that describes the communication pathways that we will use.



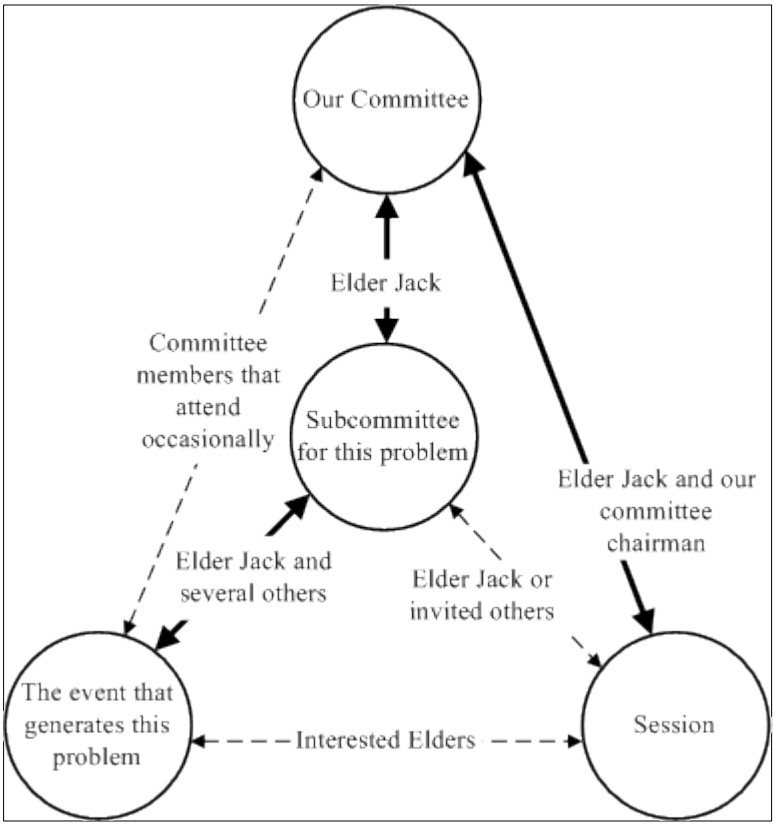
The key to the structure shown above is to compartmentalize the communication. I am the Project Manager. I work for the service provider. The heavy dark line that connects my circle with the Project Lead for the customer is the primary communication pathway we will use. Work assignments will be transferred from the Project Lead to me. Finished work will be given by me to the Project Lead. My purpose for channeling communication like this is to ensure that the developers working offshore do not have multiple people giving them direction. I am interposing myself between this customer and that team.

Note that I am not trying to be their only communication pathway. If I did that then we would slide out of a matrix structure and revert to a projectized structure. On the contrary, the structure I propose actually extends the concept of a two-manager matrix into a three-manager cube. Note that there are three arrows connecting the developers with this virtual organization. I am their project manager. Vijay will take the role of Senior Developer. In addition, the developers also have an offshore personnel manager. This is a three dimensional cube. Vijay is going to focus on achievement, though he also dabbles with power. I am supposed to deliver results - which is achievement. But one of my roles with this customer is to represent our

position in the contract - and that is power. The third dimension is the personnel manager at the offshore location. He needs to focus on affiliation while also being available to intervene if either Vijay or I push too hard on achievement or power.

I am proud of this structure. Clearly I am not the first to implement this cubic structure, but I think I might be one of the first to be able to explain why it works. Well, that is a bit premature. So far we do not even have the requirements for our first deliverable. I am confident, however, that this structure is going to help us succeed.

This is a new paradigm for me. I understand organizational structures and I have adapted them in the past to facilitate my projects. I felt comfortable with Fritz's explanation of the impact structure has on organizational change. (Robert Fritz) I have been working with Roger's explanation of communication for several decades now. (Carl R. Rogers) But this is something new. I am now experimenting with the concept of communication structures. For example, consider the following diagram that I drew when a committee I was on displayed some odd side effects.



I am a Deacon and my good friend Mark is an Elder in this Presbyterian Church. There was a problem that was brought to the attention of the Session - the governing body for the church. A committee was formed and we studied the problem and recommended a solution. But the process of getting there was emotionally difficult for Mark. This puzzled me and I love puzzles - especially when they relate to organizations. I pondered this and drew the diagram shown below. Do you see the stress that this communications structure placed on Mark? My recommendation to the committee was to divide this work between two Elders. The simplicity with which I could illustrate that problem and come up with a recommendation helped me realize the power of

communication structures. Not organizational structures, but the structure used for communications.

Thank you for allowing me to digress. Now getting back to the focus for this chapter, the next thing I did was to document the metrics we would use to measure this project. I setup a spreadsheet with the following tabs:

--

Issues Log: a summary of the issues we are facing and our status on each.

Milestones: a quick overview of what is due when.

Labor expenditures: time tracking with costs.

Budget forecast: estimate at completion versus income.

Invoicing: tally of invoices issued and payments received.

Materials: tally of expenses related to hardware and software.

Cumulative Flow Diagram: tracking the constancy of our work effort.

Service Level Agreement Compliance: a list of incidents, the time required to resolve the incident and comparison of the time required to the time allowed by our service level agreement.

Meeting Value: my tracking of the results from surveys on meeting effectiveness.

--

I also created a procedural guide to explain each of these metrics. Then I met with Greg and Vijay and described this approach to best practices. Greg was impressed and said that this was exactly the type of documents he needed so he could implement best practices on our other projects. Vijay was not impressed and felt that all these metrics would do is highlight our failures and not provide value when we succeeded. And here we see the difference between a focus on power and a focus on achievement. Greg wants metrics and he wants this company to get better and better. Vijay has an aversion to metrics. People who are motivated by power know that metrics are a two - edged sword. If you measure and report impartially there is always the chance that what you report will be bad. Even so, we reached agreement to give this a try.

But one of the key lessons that all three of us learned years ago is to experiment in a controlled environment. We will implement these new structures and track these metrics on this tiny project because this is safe. I believe in this approach and I will use this experiment to work out any bugs in my process. Greg believes in this process but knows it will be difficult to sell governance to his other project managers. And while Vijay is skeptical, he is willing to give it a try as long as we minimize the risk in case his misgivings prove true.

I then scheduled a kick off meeting. I invited Greg and Vijay from our company. I invited the Project Lead and Technical Lead from the customer's organization. The Project Lead is named Luke. The Technical Lead is someone named Kathy. I know their boss from prior work I did for this customer a few years ago so I invited him as well. When the meeting occurred, Luke and Kathy and I met for the first time. They seem nice. I have always enjoyed working for this customer. They have a good ecology. They value people and treat them well. We told them we are ready to start work. We gave them the resumes for the two offshore developers and we briefly outlined my prior experience in using the methodologies that this customer requires. It was a good meeting.

Monitoring and Controlling

The PMI process groups include "Initiating", "Planning", "Monitoring and Controlling", "Executing" and "Closing". (PMBOK) On this project, "Initiating" was the negotiation and bid process that culminated with the signed contract. That contract serves as our project charter.

During "Planning" we hired two programmers, created the communications plan, documented our process and held the kick off meeting. Our process document will serve as the project plan for this project because this is an iterative development effort. We do not yet know what we will build. The business users will process requests and we will deliver code to implement the agreed functionality. There is some risk in agreeing to build an undefined product, but that is just part of what makes this agreement mutually beneficial. The company I work for likes a challenge and is happy to respond to the unknown. The customer does not like risk and does not like the unknown.

At the conclusion of the kick off meeting we transitioned out of planning. I, as the project manager, will now focus my efforts on "Monitoring and Controlling" this project. Our developers will focus on "Executing" and turning requirements into deliverables. We hope to avoid the "Closing" process for a long time by delivering high quality work on time and thus win successive extensions to this contract.

The First Deliverable

A few days later Luke gave me a brief explanation of our first coding assignment. Unfortunately he also told me that his employer has decided not to give our offshore team access to their development systems. I updated Vijay and he was not surprised. This is a fairly common occurrence when working offshore. So he pulled a programmer off another project and assigned that programmer the task of copying the customer's code to a development server we setup in our local office. Going forward I will copy the requirements, test data and updated code to that location, as needed. The offshore developers will work in that environment and either modify existing code or develop new code as required by the customer. Then, I will test the results and copy the updates back into this customer's development environment. Setting up that environment took time.

While Vijay's team worked to set up our development environment I collected the requirements for our first assignment and created relevant project documents. This first assignment is just a few weeks of work to modify a few screens. I created test scripts to describe exactly how to test each update. And once our development environment was setup the two offshore developers read through the requirements for this project.

When the customer put this project out to bid they specified a specific set of development tools from a specific vendor. We hired two developers who were experts in that programming language. By way of analogy, suppose we wanted to open a new office in France. We would need to hire someone to head sales in that office. We would expect that person to be able to communicate with the staff in that office by speaking French. And, buried underneath those requirements is the expectation that someone proficient at sales in Europe should be able to work with people that speak one or two of the other common European languages. When we hire programmers we do not search for specialists that only know how to use one language. We search for programmers that are strongest in one language but ready and willing to use other languages as required. So, if this French speaking sales manager wanted to transfer to Germany, we would expect him to become sufficiently fluent in German to be able to make a sale. I expected these two programmers to be able to fend for themselves with a couple programming languages and I expected them to be willing to learn new languages as required. We missed.

Neither of these programmers was willing to work in the language required by the customer for this first assignment. This put us all in an awkward spot. Vijay had asked these two people to quit their previous jobs and come to work for us to do programming in their preferred language

on this project. I assumed their ability to use other languages had been discussed, but it turned out that it had not been raised in the hiring process. I then went back and I probed what this customer actually expects. I learned that they had changed their plans but did not change the request for bids because it had already been distributed. We bid on a project with the expectation that we would work with one programming language. We hired people to work in that language. But the customer had already changed their plans and decided on another language. By analogy, we hired French speaking experts and now find that they need to learn Romanian.

Here is where the cubic reporting structure works. My project has no need for resources that will not program in the language now specified by this customer. Vijay currently has no other work assignments that call for the language that these two people prefer. Their third manager, however, knows that we lured these people out of their prior jobs and thus we owe these people an opportunity. The decision was made to keep them both on our payroll while searching for projects that will make use of their talents.

Vijay then scrambled and found two other programmers that want to work in the language that this customer is now specifying. All of that effort cost us time. We lost time because of the work required to create our own development environment. We lost time searching for replacement developers. Those risks are part of the deal. We agreed to do this work with the risk that the unexpected might happen. This customer is paying us a premium because they like the ability to change direction frequently but they want to transfer the downside of those changes to someone else. In essence, we provide insurance to protect them from the side effects of their mistakes.

A few days later Vijay setup a conference call and I talked with Sanjay and Srinivas - our two new offshore developers. They both impressed me with their willingness to adapt as required by this customer. They had both reviewed the requirements and examined our development site and they were ready to start. We lost a little time, but now, as Jim Collins puts it, we have the right people on the bus. (Jim Collins) A few days later Sanjay sent over the code for the first couple modules and I passed them on to Luke - the project coordinator representing this customer. Luke said thank you but then casually mentioned that he did not need these modules. Luke stated that offshore development has not worked on any of the projects he coordinated in the past and he does not expect it to work now either. He was not going to look at this code because he knew it would not be acceptable. Instead, he had asked Kathy to go ahead and do this work and he knew that Kathy would get it done in a couple weeks. Further discussion was futile.

While in his opinion we had "failed" with our first assignment, Luke agreed to give us another chance and sent over a brief description for our second assignment. I forwarded that preliminary specification to Sanjay, Srinivas and Vijay. Then I set to work to turn those rough notes into a requirements document and a test plan. I will give Luke the benefit of the doubt. They had changed the programming language we were using, but there was some fine print in the contract that said things might change once in a while. He had rejected our first set of code without even looking at it because he has a prejudice against work that is performed offshore. Well, those are exactly the types of risks that caused Greg to think of me when he put his bid together for this work. All I need to do is build the relationship with Luke and clearly communicate the quality of the work we are performing. I cannot expect to overcome Luke's prejudice against offshore work all at once. That will take time. And this is going to require a change to the ecology of this environment. It might be permissible to change the programming language after we hired our staff, but those actions do harm to the ecology. It might be permissible to reject our code without looking at it, but that action harms the ecology. I need to

use the tools I have as a project manager to repair and then revise this ecology. And the two tools that I find most suitable for this purpose are neutral documents and effective communication. I will document our plans and communicate with Luke to make this work.

The Successful Second Work Package

About a week later I gave Luke a requirements document and test plan for this second coding assignment. He glanced through them and said we could proceed. These were fairly small code updates and Sanjay and Srinivas both understood what was required. I setup a shared workspace and asked that they each post an update every day to describe what work they had finished. Then I waited. Nothing happened. I followed up, got their commitment to use that shared workspace to post status updates. I waited but again there were no updates. I asked Vijay why I was not getting updates on status. I reminded him that our first work assignment had been rejected and explained that it is vitally important that I give Luke frequent status updates and send him samples of the code as soon as possible. Vijay replied that programmers are just not interested in status updates. Writing code is important. Providing status updates is too much like doing paperwork and real developers do not do paperwork.

How can you persuade the developers to give you status updates when their technical manager thinks it is meaningless? Ah, here is where the cubic organizational structure came in handy. I sent an email to their personnel manager and explained that I need a daily phone call from Sanjay and Srinivas. Vijay replied to my email that he also supports this approach. The personnel manager then replied that he thought this was a good idea and so I got a phone call that night from Srinivas. We agreed on a schedule and I now get a status call every business day.

This second code change went well. I tested it thoroughly and transitioned the code to Luke. Luke again grumbled and reminded me that he already knows that offshore work is never going to succeed. So I showed him that the code worked in our test environment. He agreed to ask Kathy to set it up in their test environment. It worked. He had Kathy review the code. It matched our requirements, matched their coding standards and matched the testing criteria. Kathy updated Luke and Luke moved our code into production. By over - achieving on the documentation and by insisting on a daily status update we succeeded. Luke is now using code developed offshore. Note that I never confronted him to challenge his belief. If I had told him he was wrong, then he would have been more determined to prove his was right by finding fault with our work. Instead, I accepted Luke as he is and then strove to give him evidence that would change his behaviors. His opinion has not budged. But when he put that code into production he created a dissonance between his actions and his beliefs. That dissonance will change Luke.

Communication for the Third Work Package

Luke and Kathy feel the pressure most developers feel to either stay current with technology or get left behind. There is a new project that the business wants implemented and Luke sees this as a great opportunity to try out a new programming language. Luke invited me and Vijay to a meeting where he outlined the requirements for this third work package. Both Vijay and I have some prior experience with this software and agree it has benefits but we expressed our concern that it might take too much time for Srinivas and Sanjay to learn this new language and then build the application. Luke believes this new language is the right tool and wants us to use it.

Vijay connected with Sanjay and Srinivas that night and both were reluctant to take on another programming language with such tight deadlines. Vijay reached a compromise. Srinivas will learn the new language and start the work while Sanjay continues to work with the other

languages. Extending my prior analogy, we will now be working in French, Romanian and Swahili.

After a few days Vijay realized that we were going to struggle to meet the delivery schedule on this third work package so he began doing a lot more work himself to support this effort. This is taking him away from his other work assignments and his labor is an expense with no offsetting income. After a while he decided that the only way we could meet the schedule defined by Luke was to add another programmer. Vijay assigned Mahesh to work with us on this project and the work began to take shape. I extended our weekly status meeting with Luke to now include a review of the work already completed for this work package. I want Luke to see results quickly.

After one of those meetings Vijay and I talked about my weekly metrics report. First, while our contract says we need to respond to incidents so far none have been assigned to us. Vijay recommended dropping that metric from our weekly status report and I agreed. Then we talked about the cumulative flow diagram. Vijay explained that Luke does not understand this chart and I explained that I expect it to take time for him to absorb the concept. Vijay continued with this theme and it occurred to me that the real issue is that Vijay does not understand this graph. I spent some time telling Vijay the types of things I would tell Luke to help Luke understand. It made no difference and Vijay asked that I drop this metric from our weekly status meeting.

Following that discussion I thought about the value that these metrics provide in creating a common understanding of progress. Greg has asked me to define appropriate metrics for best practices and I have done that. But if we cannot use those metrics then what is the point of tracking them? To me this is a pivotal issue. Do we want best practices or not? I need to probe that issue and find out where this company stands. I put together a proposal for implementing best practices as a way of probing for feedback. I sent that proposal over to Greg and Vijay and to two of the other project managers that I worked with in the past. I got no response. I waited a week and then sent a follow up email. I got no response. I then stopped by the office and asked Greg for his feedback. Greg is an honest, honorable person and he told me he simply had no time to read my proposal. We talked and he raised the implication that if he is so busy reacting to new opportunities then he will never have the time to implement process improvements. He promised to get back to me in the future and I thanked him for his honesty. I had the answer I was looking for. We will not be doing best practices at this time. I then dropped the service level metrics on incidents and I dropped the cumulative flow diagrams from our weekly status meetings.

I updated Vijay and he suggested I also stop assessing the meeting effectiveness. In his opinion, all the meetings are effective. And if a meeting is not effective then we do not want to tell people that it was not effective. I explained that these metrics give me immediate feedback and allow me to take corrective action before it becomes a crisis. He does not understand that concept. From his point of view if you measure you might find something you do not like, thus it is better to not measure. From my point of view, the reason for measuring is to find things to change so that you can get better and better. Vijay has the power approach - claim victory immediately and then avoid testing that reality. I have the achievement approach - probe until you find something to tweak and then work to make it better and better. Greg is caught in the middle. He is achievement focused, but he is overloaded. And, most importantly, there is one detail I forgot to mention earlier. Greg is an employee while Vijay is one of the owners. And that tips the balance. I will stop assessing meeting effectiveness and do what Vijay requests.

Meanwhile, Srinivas and Mahesh (our newest programmer) are making good progress but their efforts seem scattered. I took some time to meet with Mahesh and go through the project

schedule with him. As is typical, there is something about a project schedule that seems to create a mental block. So I copied the tasks out of Microsoft Project and pasted them into a spreadsheet. Then we met again and like I have seen before, the light went on. From now on I will communicate our schedule to Luke, Vijay and Mahesh using a spreadsheet.

I was beginning to get concerned that something similar was happening to my weekly status reports as well. Fortunately the standard template this customer uses for their weekly status report is a word document and it is fairly simple and easy to read. But I have noted a couple issues there and raised them in our weekly meetings and nothing seems to be happening. So I followed up with Luke and found that he does not have the time to read the status reports that his company requires that I submit. Nor has he had the time to read the requirements and testing documents that his company requires that I prepare. Here we have a disconnect.

When we next met I took the time to walk through all of the items where I needed a response from Luke. This annoyed him but it opened things up a bit. As a matter of fact, it opened things up a lot more than I had expected. It turns out that the requirements that we had originally discussed no longer aligned with what Luke was expecting. Somewhere during the prior weeks Luke had decided on a fairly significant design change and simply forgot to tell us about it. This worries me.

There are four standard means of communication: written or verbal, formal or informal. Examples of each are shown listed below.

--

Written and Formal: A contract

Written and Informal: An email

Verbal and Formal: A command

Verbal and Informal: Conversation

--

Because our focus is on doing as much of this work as possible offshore, I need to rely on formal, written documents - things like project schedules, requirements documents, test plans and status reports. Luke, on the other hand, really prefers informal verbal conversation. He likes to just spin his chair around and tell Kathy to go do something. When he gives me informal verbal instructions like this I need to turn those instructions into formal written instructions so that the offshore team has a way to understand what is required. I always copy Luke and Vijay when I send out those communications and I have been following up with Luke and Vijay regularly to ensure that what I have been documenting is what they think we are supposed to do. What was becoming clear in this conversation is that Luke has not been reading those documents. I pushed on this issue to make it clear how catastrophic that can be to our offshore effort. To my surprise Vijay came to Luke's defense. It seems that Vijay has not been reading any of my documents either.

Luke likes to give informal verbal directions and then verbally change directions a couple times a day to get the results he wants. This is fast and efficient. But Luke's company wants a cheaper price for services and they want this work done offshore. Luke is not going to phone the offshore team and even if he did we are out of synchronization with their time zone. The way to make this work is to take the time to document what we are going to do in writing. Now it is clear as to why Luke "knows" offshore will never work. Unless you communicate effectively with the offshore team they can never succeed. This is going to be an ongoing problem. Fortunately Mahesh and Srinivas were able to correct the misalignment between my documents and Luke's expectations before the due date for this work package. We got lucky this time.

The Failed Fourth Work Package

While Mahesh and Srinivas were busy on the third work package Sanjay was doing fantastic work to finish the fourth work package. I had documented this work package far more extensively than I had the prior work packages. But based on what I had just learned about our third work package I am now concerned. I asked Sanjay to send me all of the code that he had finished so far and I passed it over to Luke to test. Luke installed our code, tested it for a couple hours and told me he was pleased. He added, however, that as he was testing he came up with ideas for a few changes. I was not surprised, but as he described those changes I began to worry about the size of the change compared to the amount of time we had left. I wrote up a description of the required changes and sent it over to Sanjay. When we talked later that night he was also concerned but promised to do everything he could to finish on time.

Sanjay did exactly what he said he would do. He delivered. He gave me the complete package nearly a full week before the scheduled delivery date. I passed this code over to Luke and asked that he verify that all of his changes had been incorporated. The next day I asked Luke if he had any issues with our code and he said none so far. The next day I asked Luke for an update and he said he would get back to me if there were any problems. We continued this dance right on up to the date when the code was to be put into production.

At 10:00am that morning Luke said that he had decided upon a fairly significant design change and asked that I have the offshore programmers make the required changes sometime in the next couple hours. I reminded him that they were asleep. He was a bit frantic, but he said he was sure Kathy could take care of it. About an hour later Luke called me and Kathy into a conference room. Kathy explained that none of our code worked and that she was going to delete it and start over again. I offered to show both of them that our code worked just fine in our test environment. They did not have time right now to look at those screens because they had a tight deadline to make. I reminded them that Sanjay is a very good programmer and it had taken Sanjay several weeks to make the changes he had already completed and thus I was skeptical that Kathy would be able to start over again and finish in two hours. Luke agreed and asked that I have Mahesh assist. Again I tried to get them to slowdown and look at the fact that our code was working correctly in our test environment and then troubleshoot from there. I lost on that one.

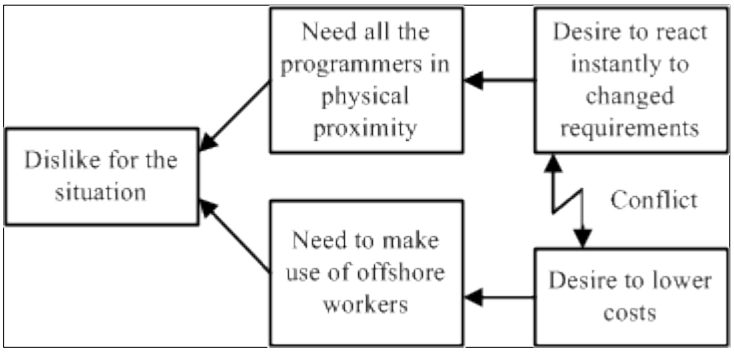
Mahesh looked through the work that Sanjay had done so that he could understand the approach Sanjay had taken. Mahesh then tried reasoning with Kathy that the work Sanjay had finished was quite complex and not easy to do over again in two hours. Kathy and then Luke told both me and Mahesh that they had already made their decision and what they needed now was for Mahesh to get busy writing new code.

Two hours later Kathy realized that she was not going to finish in time to move the code into production that night. She and Mahesh kept coding until late that night and then they started again early the next morning. When he got in that morning Luke tested Kathy's code and found significant issues. As Luke started explaining the requirements to Kathy it dawned on her that the work Sanjay had already done was, after all, the best approach. So Mahesh copied Sanjay's code from our test server and overwrote the code that he and Kathy had just spent ten hours frantically writing. He and Kathy then started to work on Sanjay's code, not to redo everything, but simply to apply the changes that Luke had requested. In a few hours they had things working well enough to push it into production. Luke then called me, Mahesh and Kathy together for a meeting.

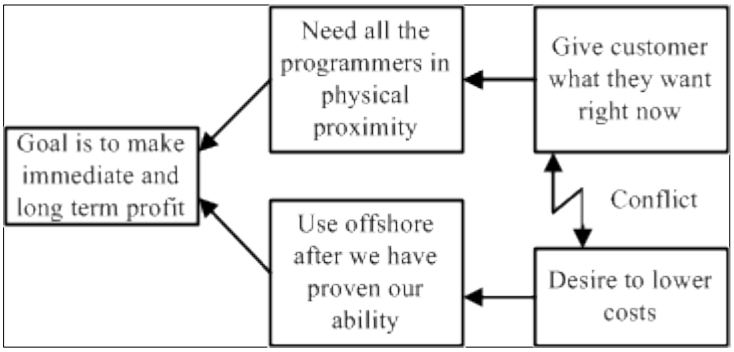
Luke again reminded me that offshore development never works. I countered that the code that had been delivered exactly matched the requirements we specified. Luke responded that this is the heart of the problem. All he wants is a couple programmers sitting here so that they can do what he tells them to do. He does not want programmers working offshore and he wants nothing to do with project managers. "Project managers add no value. Programmers are the people that do real work."

Earlier in this chapter I explained that the purpose for contracts like this is to insure the customer against risk. The primary risk that this customer faces is continually fluctuating requirements. In my opinion, the best way to mitigate against indecisive requirements is to use iterative coding. This customer, however, also wants lower costs. And the magnitude of the cost savings they want can only be obtained by working offshore. Poor Luke is caught in the middle. His users can change the requirements on his project right up until a few hours before the code is scheduled to deploy. But his management wants the cost benefit that comes from working offshore. Luke cannot change the users and Luke cannot change his manager. Luke can, however, express his anger at the situation.

I learned a technique called “evaporating clouds” by studying Eliyahu Goldratt. (Eliyahu Goldratt) Consider the evaporating cloud shown below. This is Luke's dilemma. His managers insist that Luke must react immediately when the users decide on a change. And his managers insist that Luke use offshore workers because they cost less. Luke wants workers sitting a few feet away from him so that he can immediately redirect them in reaction to changed priorities. Management did not give Luke what he asked for, yet they still hold him accountable for the results. The only way Luke sees to escape from this conflict is to help his managers see that offshoring did not work in the past, is not working now and will not work in the future.



Once I understood this cloud I called Vijay and I explained the reasons as to why Luke does not want us to use offshore workers. Vijay replied that it is my job to make this work. I asked Vijay which was more important - prove that offshoring works or get a renewal on this contract? I should have known better because he replied that it was up to me to ensure that both happened. Thus, my cloud is only slightly different from Luke's cloud.



Even so, I persisted and soon the conversation got a little loud. We will not get the benefit from lower offshore costs if we need to do all the work over again two or three times. We will not get the contract renewal unless we make Luke happy. And the only thing that will make Luke happy today is to put more programmers on - site and help Luke deal with the chaos that his managers have created for him. We then both reverted to our primary orientations. I quoted from my metrics report that shows that our labor expense is over budget and our forecast went negative a couple weeks ago. I emphasized that these metrics show that we are losing the advantage of lower offshore labor rates because we need to keep doing the work over and over again and Luke is not motivated to solve this problem because he believes his is better off demonstrating that offshoring does not work.

I should have known better. Achievement speaks metrics. Power is adverse to metrics, especially metrics that are negative. Vijay told me to stop distributing the financial metrics and to stop collecting the data. That ended my gambit to use metrics to make this operation better and better. Now I am still going to collect that data anyway, but I will no longer be able to use it to help Greg understand what is happening here.

Next it was Vijay's turn to propose a solution. True to his preference for immediate impact he decided that all I need to do was to update the procedural document and our problems will be solved. I asked if he had finished reading the prior version and he said he had not. I reminded him that Luke has not read the prior version either and I asked him how updating a document that no one reads is going to fix the problem? He replied that this is the solution and it is up to me to make that update and then get results. This is linear thinking. Consider the following statements of cause and effect. Which are true and which are only partially true?

--

If all the programmers work in the same building they can get more work done.

Using offshore labor costs less than using on-site labor.

Increasing the size of the team means we will get better results?

--

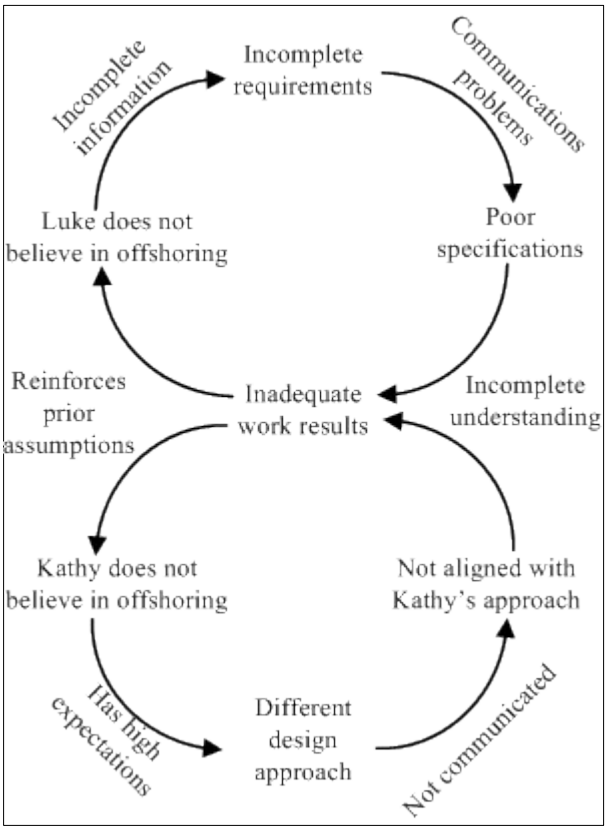
In my opinion, statement one is only partially true. I have worked with offshoring efforts where the time zone difference was an advantage. I have worked with offshoring efforts where the isolation was an advantage because that team was better able to focus. That, after all, was the key element in the proposal I gave Fred in the prior case study. In general, there are advantages to having everyone communicate verbally and there are advantages to requiring that communication be formal and written. For example, if Luke would help us revise the requirements document that I prepare and if Luke would sign those documents then my offshore team could work is isolation and be judged solely by how well their results match the requirements.

Similarly, in my opinion, statement two is only partially true. We are not getting this advantage today because this is a fixed price contract and we need to keep doing the work over and over again. It is also clear that as long as this customer minimizes the requirements process we will need to continue to do the work over and over again.

And I also think that statement three is at best only partially true. First, we need to remember Brook's Law: “Adding manpower to a late software project makes it later.” (Frederick P. Brooks)

Then, beyond that, adding more gasoline to a fire is typically noted as a way to increase, not decrease the spread of the flames. Putting more resources into an effort that already allows the users to redesign the application hours before the production release is only going to encourage the users to be even more irresponsible in their actions. If anything, we need a dampening effect.

That is the magic word - dampening. This system is currently running as an amplifying loop. I need to dampen the negative feedback. Consider the diagram shown below. Luke does not believe in offshoring. Therefore he sees little incentive to give us a full description of what is required because he assumes the effort will fail anyway. This means we are working with incomplete requirements. Then magnify those communication gaps with the inherent communication problems that come from working offshore and the net effect is that the work is poorly specified and incompletely understood. The natural result is inadequate work results.



Every time we deliver a product that does not meet Luke or Kathy's expectations we reinforce their prior assumption that this is a doomed effort. Add onto that the fact that Kathy has very high expectations - expectations set so high that even Luke cannot do work good enough to please Kathy. Add onto that the fact that Kathy often has her own design in mind for the code but never participates in the requirements sessions and never tells us what she was thinking. Every

step of the way we add to the problem. What we need to do instead is to dampen - we need to start subtracting. Vijay wants me to work on the top most step in this double loop. He wants me to enhance the requirements. At best that would help us better align our work with Luke's expectations but it does nothing to change Kathy. Thus I reject this suggestion.

There is an approach to artificial intelligence called neural networks. The premise for that technique is that pathways that lead to success are reinforced while pathways that fail to deliver should atrophy. You represent those pathways through nodes with branch points and assign probabilities of success to each branch. My mental model of this situation is being adjusted and the probability of success I assign to our current pathways is being decremented. I then began searching for another pathway.

The approach that I came up with is that I will do the opposite of what Vijay recommends. Instead of trying to get better requirements I am going to spend less time trying to guess what Luke wants. After all, I only have a fifty percent accuracy rating so far. So instead of working on the top part of this system, I am going to interpose myself into the juncture where Luke and Kathy collaborate. My plan is to bring inadequate work results to those sessions and allow Luke and Kathy to critique. Thus, I will finally be able to get requirements from them. And they, in turn, will see that the offshore effort increases their importance and is not a threat to their jobs.

Today we refer to software development like this as iterative development. Few developers, however, are able to intentionally strive for inadequate results. Our pride gets in our way and we put more and more time into trying to make our code perfect for the requirements that we understand. In this situation that effort is a waste. I cannot get adequate specifications when all I have is a five minute verbal exchange with half of the team that will judge our results. My problem is going to be forcing that offshore team to give me code that they know is inadequate. I am going to need to force them to give me junk that they are ashamed of so that Luke and Kathy can tell us what it is that they are actually expecting. I will fail in that effort unless I dampen the communication between here and there. I need to absorb the insults and prejudice in these meetings and suppress that tone in my conversations with Sanjay and Srinivas. I need them to focus on the value they are delivering. I need to continue amplifying the positive reinforcement that they receive so that they understand how much I value their effort.

Years ago engineers used iterative techniques to solve problems that were too complex for their slide rules and calculators. One such technique was the Hardy Cross technique for analyzing a grid of water pipes. The core of that approach is to accept our inability to know the results in advance. I entered the engineering profession at the juncture where computers were just beginning to penetrate the industry. One of my first programming jobs was to put the Hardy Cross technique into a computer program. I spent considerable time studying the process and the implications. What I found is that inadequate initial guesses had little impact on the eventual solution. Perhaps the computer had to repeat the iterative cycle a few more times, but those few extra cycles took less time than it would have taken the engineer to prepare better initial approximations.

That is the technique that I am going to use here. I am going to spend the minimum time required to document my initial guess at our requirements. I then need the offshore team to spend the minimum amount of time possible to give me their first pass at an implementation. Next I will show those results to Luke and Kathy and they will gladly tell me just how far we are from the target. They still might not tell me where the target is, but if they just keep on telling me whether we are getting closer to the target or further from the target eventually we will get it. But I need help on this. I am going to focus on the people. I am going to enlist Mahesh to focus on

the technical. In essence, I am going to exchange Vijay - one of the owners of this company for Mahesh - our newest employee. We will still have a managerial cube. But the three managers will be the personnel manager, myself and Mahesh.

Consider the parable told in Matthew 13.

“That same day Jesus went out of the house and sat beside the sea. Such great crowds gathered around him that he got into a boat and sat there, while the whole crowd stood on the beach. And he told them many things in parables, saying: ‘Listen! A sower went out to sow. And as he sowed, some seeds fell on the path, and the birds came and ate them up. Other seeds fell on rocky ground, where they did not have much soil, and they sprang up quickly since they had no depth of soil. But when the sun rose, they were scorched; and since they had no root, they withered away. Other seeds fell among thorns, and the thorns grew up and choked them. Other seeds fell on good soil and brought forth grain, some a hundred fold, some sixty, some thirty. Let any one with ears listen!’” (NRSV)

I cast the seeds of best practices in several prior organization but as soon as I left all of my work ceased. It was as if the birds came in and ate up all my metrics and all my processes. I came here to work with Greg to implement best practices. But, once we encountered resistance the effort was abandoned. Greg's desire for best practices fell on rocky soil, withered and died. Here my effort is being choked by thorns. The way around that is to find the good soil where these efforts can succeed and nurture that soil so that it will support the effort. Mahesh, Sanjay and Srinivas want to grow. They represent the good soil that can yield a hundred fold. If I can invest the time into teaching them how to make this work then they will carry that message with them to other projects and our success will grow. Now, by implication, that implies that Luke and Kathy are the soil that is choked with thorns.

“He put before them another parable: ‘The kingdom of heaven may be compared to someone who sowed good seed in his field; but while everybody was asleep, an enemy came and sowed weeds among the wheat, and then went away. So when the plants came up and bore grain, then the weeds appeared as well. And the (workers) came and said to him, “Master, did you not sow good seed in your field? Where, then, did these weeds come from?” He answered, “An enemy has done this.” The (workers) said to him, “Then do you want us to go and gather them?” But he replied, “No, for in gathering the weeds you would uproot the wheat along with them. Let both of them grow together until the harvest; and at harvest time I will tell the reapers, Collect the weeds first and bind them in bundles to be burned, but gather the wheat into my barn.”"" (NRSV)

Luke, Kathy and Vijay are not thorns. They are just caught in soil that produces thorns. And any action that I would take to destroy the thorns is going to do harm to the wheat that they can produce. What I need to do is to continue to fertilize and water this soil and allow the thorns to grow along with the wheat. Later the harvesters will separate the wheat from the thorns. That is not my job. On the contrary, my job is to try to minimize some of the stereotyping that I see going on here. Even though the environment here is much better than it was when I worked a couple blocks down the road at their competitor's facility, there is a class structure here that corrodes the relationships. Employees consider themselves superior to contractors and contract programmers are superior to contract project managers. Similarly, on-site work products are presumed to be better than are products from offshore.

We create these environments. We nourish these ecologies. When I worked in construction, we rewarded workers for their muscle power and reinforced the stereotype that construction workers do not use their brains. In data processing we encourage people to focus on the

technology and reward them with labels like geek and nerd. We reward people for distancing themselves from who they are. We amplify the feedback loops that cause people to abandon their efforts at integration and wholeness. Vijay is caught in this spiral. If he was the technical manager for this company then he would soon find himself challenged from multiple directions by people who are technically his superior in one narrow discipline or another. But as part owner of this company he is immune from those challenges. He does not get the feedback that would tell him that he needs to learn to listen and collaborate.

I have made a few allusions toward the barbarian behaviors of IT professionals. Here is the reason those behaviors exist. We reward those people for amputating their personalities and help them stunt their social and emotional growth. We treat programmers like a machine that can only produce one product and we work them twenty-four hours a day to create more and more of that same product. We need to change this.

I need to change this ecology and help these people grow out of the shells they now inhabit. I have some ideas, but today I do not feel like I have answers. Therefore I am going to first change myself and then try to change this ecology. I am going to water and fertilize this ecology and let time tell where the wheat grows and where the thorns dominate.

The first change that I need to make is to start running again. You might wonder what running has to do with resolving our project management issues. The connection is that I am not acting like an integrated whole. I have given up important aspects of my life in order to put more time into this project. But that is counter-productive. I resent the time that this project takes and I resent that I am putting time into this effort to the exclusion of other parts of my life. Tonight I am going to start running again and make it a point to do so at least five nights a week.

Re-Planning

It is time to go back and check my alignment with the goals that launched this journey. Please indulge me while I do a quick mental comparison of where I am now. After all, the reason for doing a case study is to assess whether or not the processes that I propose can survive in the real world. So here we go.

We need to start with a vision of what this project team can be. What I now see is that this system perpetuates negative behaviors. We reward people for seeking dominance to the exclusion of cooperation. We reward people for focusing narrowly on achievement to the exclusion of their own maturity. We freeze people out of the decision processes and then act surprised when they feel alienated. These behaviors are normal responses to the environment.

I believe that a properly defined vision has six elements:

--

A concise expression of the vision.

A champion to embody the vision.

The vision must be aligned with reality.

For success, the vision must be bought into at three levels.

The vision must be focused and specific.

Implementing the vision requires hard work.

--

When I started this project I thought that the vision was to do a proof-of-concept implementation of best-practices. The pattern I see repeated over and over is that people want someone to hand them a certificate that says they have achieved "best practices". They are willing to pay money for that piece of paper and there are lots of consultants willing to take them

up on that offer. But sticking a plague on the wall that says "best practices" does nothing. Best practices are achieved through hard work. Greg wants best practices for this company. Does Vijay? Does Luke? If you want to live a life focused on best practices then you are going to need to do it yourself. Most of the people with vision have either already been defeated or they have turned their talents to the pursuit of short term rewards. If you want someone to define the vision and be the champion then you need to look in the mirror. If you want best practices badly enough to get this far in this book then you are the person who is going to need to make that happen.

I came here to use the best project management tools and prove that they work on projects like this. Vijay and Luke see no value in doing this. My life would be simplified if I just gave up as well. Fortunately I had lunch with my good friend Jim this weekend and he told me about a project that he just finished. He started out doing all the right project management stuff and hit resistance. The culture in that company devalued meeting minutes, project schedules and other simple tools. Jim decided to go along with the culture and then found that he had nothing to fall back on when his project ran late.

My friend Rick told me a very similar story. Rick began work in a new company that impressed him with their commitment to project management. He created minutes, used a project schedule and created a work breakdown structure to help define the requirements. Then he found that the while the company proclaimed their adoption of best practices the culture in that company was actually hostile towards documentation. Rick kept creating all those documents anyway, but he learned that it was best to not distribute them. Then Rick, like Jim, found another job where people understood the value in using those simple tools. I plan to do the same. I am going to continue to track expenditures and create my financial forecast. I am going to continue to use a project schedule and publish status reports even if no one reads them. And if Vijay continues to tell me to stop doing what I know is right then, like Jim and Rick, I am going to search for another job.

Another best practice is change management. Change management and configuration management are two disciplines shared by projects and operations. This company has a rigorous change management process, but they ignore it. Requirements are in constant flux. Rick, Jim and others have asked why my employer accepts fixed price contracts with no restrictions on scope. My response is that this is a natural reaction to the environment we inhabit. Every behavior that we choose is chosen for a purpose. Vijay put this proposal together and he is convinced that we will make a profit. The numbers show otherwise, but Vijay is content to ignore those numbers. Vijay, like many executives, has developed the ability to selectively hear only good news. And when there is bad news the fastest response is to shoot the messenger. I have worked for numerous companies that behaved exactly the same. Those companies see no advantage to best practices and they never will. Unless and until you see a clear distinction between where you are and where you can be there is no motivation to change. As long as all of the problems that arise are simply swept under the rug nothing will change. As long as we adopt the attitude that there is a linear causality between the person and the results then we are doomed. Consider Vijay's response to the problems on this project. He wants me to update a process document that no one uses. First, best practices are meaningless if all they are is paper sitting in a file cabinet. Second, this is a system. Pushing on Luke will not solve the problem unless we can also influence Kathy. We do not need more processes. What we need to do is follow the processes that have already been defined - starting with change management. Change management would allow us to manage the requirements. And if we could manage the requirements then we could deliver the right product.

By now you probably sense my frustration. We are not striving for vision. We are not following best practices for projects. We are ignoring the processes that are defined for operational consistency. But the topic that is the most frustrating to me is the way we ignore all the other rules and then blindly feel compelled to pretend like governance matters on this project. Governance, on this project, means that I am obligated to devote hundreds of hours to the creation of thousands of pages of documents that will never be read. Process, solely for the sake of having process is worse than meaningless. All of those hours I am now spending on useless documents is taking me away from tasks that could add value. I hope that nothing that I wrote in my chapter on governance is ever used to justify such nonsense.

Now, if I was not spending so much time on those useless documents I would instead spend more time on metrics. I look back now at the metrics I used on prior projects and I dream of putting them to use here. Failing to do so is, I believe, the root cause for most failed best practice initiatives. My goal in analyzing metrics is to find early warning indicators. Seven points on a control chart trending upward is an indicator. Lines on a cumulative flow diagram diverging or converging is an indicator. Find the warnings and take action before they become problems. The key reason I do not believe that best practices will ever occur here is that we are resistant to metrics. I used metrics to warn that our budget was trending negative and I was told to stop publishing. I persisted anyway in trying to get Vijay to accept the reality that we are no longer trending towards negative numbers but we are rampaging through red ink. The result was a direct order that I was to stop collecting the metrics.

This is a personality weakness in people who are strongly motivated by power. They do not want to see the evidence that their decisions do not get translated into the desired results. There is nothing wrong with that motivation. It is a part of the human mix and it serves a purpose by allowing visionary optimism to persist in the face of statistical forecasts of doom. But managers need the ability to discern. Managers need to be able to listen to the bad news. To an extent this means that managers need to be versatile enough to use the different styles of management such as telling, selling, participating and delegating. To an extent this means that managers need to have a high enough emotional quotient to be able to know to whom they should listen. To an extent this means that managers need to be mature enough to seek the good of the group instead of focusing on self.

Now I am adding to that list. Now I see that we each need to step out of our comfort zone and engage with the variety of motivations. Always taking the power path leads to the fad of the month initiatives. Always following the achievement motivation leads us to seek a state of perfection that is as meaningless as an unread report. And if any of us neglect the affiliation motive then we will find it difficult to lead the alienated.

What I want to find is a work place ecology where the parts are aligned and people are valued. I guess that means I am going to need to build it myself. I have a vision that this tiny little project can demonstrate best practices. I now realize that the best way to achieve governances is through roles and communication structures not through documents that go unread. I am going to use the best tools that I know of to manage this project - but I will not distribute those documents to those that do not have the ears to hear what I am saying. I am going to continue to measure the indicators that are the most critical to me. I will not, however, share those metrics with anyone besides Greg.

Most importantly, I am going to set as my goal the creation of a repeatable process that will allow this project to work smoothly and deliver consistent results.

Testing the Fifth Deliverable

Mahesh struggled to get closure on the fourth deliverable and he did not understand why I did not send more of that work back to Sanjay and Srinivas. I explained that this is a critical path issue. There is a lot of work required for the fifth deliverable and if I disrupt their concentration then Sanjay and Srinivas will not finish on time. Instead I am disrupting his schedule. He is not on the critical path for this phase and thus I can allow his work to be delayed without impacting the project completion date. As we talked about this I realized that the concept of critical path is a learned concept. I began using critical path when I was a teenager and I just assumed that everyone else learned it at about the same age. In the past I was surprised when people would pull me off my assigned work and give me responsibility for coordinating their projects. Forty years later I finally understand that the concept of critical path is not universally acquired. Now that I understand that the concept of critical path is a mental model that comes with some assembly required, I will make it a point to help Mahesh learn this concept. He is one of the people that will shape the future of this industry and now is a good time for him to begin to visualize a critical path.

I then initiated a campaign to change the direction of this project. Sanjay and Srinivas like to do excellent work and they like to hold onto their code until they are satisfied. I need them to give me their code while it is still premature. I lobbed over a few polite requests and we talked about it on several phone calls. But when they politely ignored my request I lobbed over a more volatile email. It got their attention and the hornets started buzzing around looking for who had dared to disturb their solitude. I spent time talking with Vijay and lots of time talking with Sanjay and Srinivas. Still they held onto their code. I wacked the hornets' nest a couple more times and tried to help them understand that I was turning their world upside down. And then they got it.

One of my favorite management books is Spencer Johnson's Who Moved My Cheese: In that book he tells the story of a few individuals that adapt quickly to change and a few individuals who prefer to opt-out. Sanjay and Srinivas, like the heroes in Johnson's book caught on quickly. I had to prod them a bit, but as Johnson put it:

“One morning they arrived at Cheese Station C and discovered there was no cheese. They weren’t surprised. Since Sniff and Scurry had noticed the supply of cheese had been getting smaller every day, they were prepared for the inevitable and knew instinctively what to do.” (Spencer Johnson)

Sniff and Scurry - the heroes in Johnson's book - knew immediately that it was time to go search for fresh cheese. Srinivas and Sanjay needed a few gentle reminders but within a couple days they were off and running as well. The old process was dead. The new process required a different approach to the work. They got it.

Vijay, however, still did not get it so Greg called a meeting. I explained that the way we had been working had led to the failure we saw on the fourth deliverable. I explained that rapid turn-around of code was the best way we could hope to get accurate requirements and thus deliver what was required. This puzzled Vijay. He could not see how doing poor work quickly was going to lead to good results in the long run. We talked for a while and then Vijay decided that the solution is to convert my test scripts into a simple checklist. Rather than listing all the steps required to execute each test what I need to do is just make a list of tests and count on people to figure it out for themselves. This puzzled me but eventually I could understand his reasoning. He felt that Luke could not understand the steps required to follow our test plan but Luke would be able to understand a list of the tests we proposed. As far as that is concerned I must agree. But I fail to see how this is going to solve the problem.

My inclination is to stereotype Vijay as being narrowly focused on power. My initial reaction is that converting a useful document into a trivial document is just repeating the same exercise that we went through earlier when I changed the process document that no one read. But that is not an accurate representation of Vijay. So my next inclination is to assume that Vijay is thinking linearly rather than thinking in systems terms. Perhaps my discussion with Mahesh about critical path has predisposed me to search for gaps in mental paradigms. Perhaps this is also an unfair assertion about Vijay. I need to be careful that I am not projecting too much of the case study fictional Vijay onto the real person that is one of the owners of this company. Even so, I believe that linear explanations for complex systems is such a common problem that it is worth a few paragraphs of explanation. Consider the following:

--

I like to ride bicycles and I have been knocked unconscious several times when I have crashed. My worse crash caused me to plunge down a twenty foot embankment while traveling downhill at about thirty miles per hour. I argued with the paramedics that I wanted to ride my bicycle home but they felt it best to take me home in an ambulance.

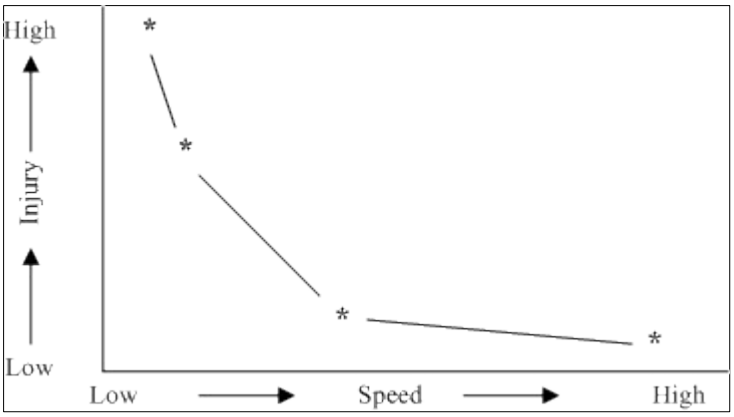
My brother used to ride bicycles but he crashed once while riding at a walking pace and broke his arm.

I used to ride motorcycles at absurd speeds. I crashed a lot because I was always pushing the limit of what I could do. My worse crash was in the mountains when I was traveling at about ninety miles per hour and I hit a patch of sand. The motorcycle and I separated and both tumbled end over end down the highway. This crash destroyed my leather jacket and damaged my helmet and yet I walked over to the motorcycle, checked it for damage, and then continued on my trip as if nothing had happened.

My brother was once sitting on a motorcycle and when he attempted to start the engine he fell over and the motorcycle landed on him. He was hospitalized and to this day has impaired use of one leg.

--

Now if we plot this data we end up with a graph that looks like the one shown below. As you can see there is a relationship between speed and magnitude of injury. Clearly traveling at higher speeds results in fewer injuries. If you doubt that, then just look at the four data points that were used to create this graph. This is valid scientific data.



Well, not exactly. Here we see two problems. First, we have too little data to draw such a broad conclusion. Second, there is little relevance between the four data points that I chose. Thus

the best hypothesis that I can now come up with is that Vijay is exercising linear thinking. He has seen situations where giving a customer a simple of list of tests has worked. He has seen situations where giving the customer a complex requirements document has failed. Thus he now concludes that there is a correlation that implies that short lists work and long lists fail.

I think it is understandable that people think linearly. Peter Senge's The Fifth Discipline would not have received as much attention as it did if he was preaching things that everyone already did. (Peter M. Senge) And yet the concept has been around for a very long time. Darwin's work on evolution is an expository on distant causes of nonlinear events. (Charles Darwin) Also consider a work that I recently encountered by Brooks Adams.

In 1913 Brooks Adams published The Theory of Social Revolutions. (Brooks Adams) Adams, in 1913, was already describing the nonlinear effects of remote causes in complex systems. I hope I am accurate when I attempt to modernize his themes. From what I understand, he suggests that the need for power creates a feedback loop. The existence of a disproportionate distribution of power causes resentment from the disenfranchised and fear in those holding the power. The natural reaction to a perceived injustice is a desire by the down-trodden to topple those in power. The natural reaction from those in power is fear of the subjugated. In reaction to that fear the powerful tighten their control which increases the perception of injustice. And thus an inequitable distribution of power initiates a self-perpetuating cycle that eventually leads to a revolution. I think this has relevance in two ways. First, the fact that I have only recently encountered this work means that Adams' explanation has not permeated society. Second, I believe that the fact that Vijay is one of the owners of this company means that he is at risk of initiating a similar feedback loop. I feel that I am not being heard. I feel that Vijay is immune to advice because of his position. In my opinion, the greater the distance between Vijay's world view and my world view the more strident I become and the more resistance he becomes. I need to change this feedback loop.

I thought it interesting that Greg stayed fairly quiet throughout this meeting. He has something else in mind but he did not let us in on his plans. I, however, left that meeting and proceeded to do exactly what I said I would do. I began my efforts to implement iterative development. And, since Vijay owns part of this company I also created the lists that he asked me to create. I gave Luke that simple list of the tests we proposed for this deliverable.

Still I felt that something was missing and I felt like Greg had something else in mind. Thus I was not surprised when he came over to visit Luke's boss. After that visit he told me that he helped Luke's boss understand that they cannot benefit from the economies of offshore pricing if they continue to reject our code. Luke's boss explained this to Luke and I now see a change in Luke's attitude. The first few modules for this deliverable passed Luke's review. There is hope after all. Iterative development coupled with a little friendly persuasion just might turn this affair into quite a pleasant experience.

I told Vijay the good news about our latest modules getting a passing score on Luke's code review and he seemed surprised that I was surprised. He reminded me that he knew all along that creating the list of test cases would solve the problem. And thus we reinforce his perception of linearity, and increase my sense of frustration. I am working with a systems approach and trying to inoluence the relationship between Luke and Kathy. Greg is working with a systems approach and trying to influence the relationship between Luke and his boss. I believe those approaches might work. But Vijay still believes that creating procedure documents that no one reads or converting test plans into bullet points is going to change the environment. He does not perceive the actions that Greg and I have taken as having value and thus he attributes the change in Luke's

behavior to the creation of a list of bullet points. This is difficult for me to deal with. I am puzzled as to how I can change his view of the world.

My head was spinning. As I pondered the immense gap between my world view and Vijay's, it finally occurred to me that Vijay is right after all. He short circuited a few steps, but he is right that the checklists are going to get the credit for changing this ecology. When I gave Luke that checklist it gave Luke an excuse to change his behavior. That checklist allowed Luke to save face. He cannot admit to himself that his prior behaviors were costing his company lost time and my company lost revenue. His self image does not allow him to see the impact his behaviors were having on us all. But he feels the pressure that his boss and I are applying. And here he has a way out of our vise grip. All he needs to do is take that list of bullet points and then tell the world that it is the list of bullet points that makes the difference. He now has an excuse to go to his boss and say that while he had been rejecting our code in the past now that he sees this list of bullet points it is safe to assume that our code is good.

I need to remember this lesson. Luke could not change his position without losing face unless we gave him a scapegoat. Changing this one document allowed him to transfer the blame to our document and thereby avoid accepting his role in this convoluted system.

Soon the rest of the code began trickling over. One by one the modules piled up. Kathy still had objections but Luke tempered those objections to a manageable checklist. Luke still changed the requirements, but he did so only after calculating the feasibility that we would still finish on time. I think this just might work.

A Few Small Diversions

We are getting better results with this iterative approach. We also benefit from having two people on-site rather than just one. I can take care of the project tasks such as requirements, schedules, budget and communication. Mahesh then takes care of quality assurance and risk mitigation. When there is a quality issue, we now have two options. Time permitting, we send the code back to the offshore team and they address the issues. But when we are short on time then Mahesh tries to address the issue immediately. The problem is that we do not have enough budget to pay for both me and Mahesh. Greg's solution is to spread my time over multiple projects.

There are weeks when I am out of town working on a project for some other customer. There are weeks when I am in five cities in five days working on five projects. This makes it a lot more difficult for me to stay focused but it means that we are spending at a rate that is closer to our budget.

One of the projects that Greg asked me to work on was a response to a Request for Proposals (RFP) from a customer seeking our advice on how to build an application to address one of their business needs. Within the RFP they stated that we need to provide descriptions of the standard practices we use and explain how our standard practices align with industry best practices. I talked with Greg about this and expressed my concerns. First, we do not have consistency in our approach to projects. I put together recommendations for metrics but Vijay does not want them implemented. I put together recommendations for organizational structures on our project teams but Greg has not implemented it elsewhere. I created a prototype portal for project status updates but neither Vijay nor Greg supported that effort. So I asked Greg to write the section of our response document that explains our best practices and our alignment with industry recommendations. As he struggled with that task he realized that we have a long ways to

go. Together, however, we came up with a suitable answer that described our lack of process as a flexible approach to meeting customer needs.

After we submitted that RFP I talked with Greg about the process. Greg reminded me that most of the people in most of the companies that have best practices do not actually follow their own guidelines. His point is that it is better to not adopt policies that we will not use than to burden people with processes that do not work. There is merit in that approach. When I work in Luke's office I am surrounded by a project team from one of our competitors. They have highly standardized processes and they are a CMM Level 5 software development shop. As I listen, however, what I hear is continual conversations about how to make what they are actually doing look like it fits in their process documents. They have best practices but what they do is no different than what we do. They just package it to look like they follow their process.

It turned out that our words about being flexible worked better than did this competitor's response to that same RFP. That customer does not have standardized processes and they did not want someone coming in and telling them how to run their shop. Thus our description of "flexibility" was actually the type of response they wanted.

With that task finished, Greg asked me to next write a response to another RFP. This customer, however, specifically states in their RFP that we must give them screen images from our project management portal. I asked Greg for his advice on how to deal with that requirement. Greg sent our sales team on site to meet with the customer and get more information. The results were ambiguous. The sales lead came away from that meeting convinced that the portal was not going to be that important. Still, I had to leave that section in our response blank because we do not have a portal. As the deadline loomed Greg became more anxious about that gap. But before it reached a crisis, the customer retracted their RFP and cancelled the project.

I talked with Greg and reminded him that sooner or later we are going to need to catch up with our competitors. Sooner or later we will be squeezed out of this business if we do not implement best practices. This last RFP convinced Greg that we must act. We talked about options and then Greg set up a conference call with Ram, the manager of our offshore programming team.

Ram had been waiting for this opportunity. He has contacts who can come in and arrange for us to be certified with CMMI or ISO. He already negotiated discounts with those firms. All he was waiting for was authorization to act. Ram assured us that we will be CMMI certified by the end of the year. I probed. First, how will Ram get Vijay to agree? Ram assured me that Vijay can continue to do as he wants and we will still be certified. Next I asked how certification will help us respond to an RFP that asks for a description of our project status portal. Ram assured me that the portal is not important. I probed on a few additional points and then we closed the call.

I asked Greg how it is possible to be certified as a best practices vendor without changing the way we do business. Greg explained that those certifications are just a business transaction and for the right sum of money we can achieve certification in whatever we need. I said that to me best practices are a journey. I described my friend Manny. Manny knew he was dying and yet Manny took on the studies to become a Project Management Professional (PMP). Manny only had a few months to live and yet he volunteered to help others pass their PMP test. Manny strived to do better and better and understood that the journey was the goal.

This is what best practices are all about. Best practices are a tool that helps us grow. First, our species had to evolve physically. Then we needed to mature mentally. Now we are evolving our social constructs. Within IT we begin with personal knowledge. Then we document that

knowledge so that it is shared. Next we form a community. The resulting social structure is greater than any individual.

I explained that it is not the obtainment of a certification that changes the company, it is the process. During the journey people learn to work together. They learn to align their individual efforts to the corporate goals. It is not the certificate that changes the company; it is the journey that changes the people.

Then I returned to the disconnect between standardization and the entrepreneurial approach that Vijay likes. What I see is that the journey is also a selection process. The people that enjoy the journey put in the work that it takes to achieve the results. The people that do not want to change then self-select themselves out of the company.

The deeper I go into this search for best practices the more I return to a search for meaning. More and more I am realizing that all of this is a façade. Companies want certificates without effort. People want to claim best practices without changing any of their own behaviors. But even those generalizations are too granular. What I see now is that "humanity" is on a journey. Humanity is now striving to construct a social order that works. Best practices are just a tiny part of a mammoth whole.

Consider the longitudinal study captured in the Christian Bible. Adam and Eve were concerned about themselves and did not share responsibility for their actions. Joseph took the concept of family and enlarged it to join many households into one nation. Moses separated the us from the them and displayed developmental level three. The prophets described the actions of foreign nations as part of God's grand plan to shape the nations of Israel and Judah and gave us a vision of developmental level four. Jesus took the Jewish holy words and began to share them with people from other nations. And in the final book in the Christian Bible we hear that the acts of all the peoples of the world are part of God's massive plan to create a new society. The Bible contains a longitudinal study that describes the evolution of our social constructs.

Perhaps this is what Martin Buber meant when he wrote:

“Meeting with God does not come to man in order that he may concern himself with God, but in order that he may confirm there is meaning in the world. All revelation is summons and sending. God remains present to you when you have been sent forth; he who goes on a mission has always God before him: the truer the fulfillment the stronger and more constant His nearness.” (Martin Buber and Ronald Smith)

I still find those words cryptic, but I sense that I understand them. Kramer and Gawlick say that the core of Buber's philosophy hinges on the first chapter in Genesis:

“In Buber’s mind, Adam mistook God’s original intention in creation. Adam didn’t realize that he had to work toward perfecting the image of God placed inside him. Buber came to recognize in these verses (Genesis 1:26-27) his life purpose. And not just his alone. He understood this to be the task of all people who recognize God’s address to them. The task is to become a partner with God in creation.”

“First we must cut away our mind-forged perceptions of God, what Buber calls in Philosophical Interrogations our ‘passionate devotion to a fantasy image that one regards as God’. In a book of essays, Israel and the World, Buber writes that we enter a creative partnership with God by ‘imitating God,’ by ‘cleaving to God’s ways,’ and not by constructing God from our imaginations.” (Kenneth Paul Kramer with Mechthild Gawlick)

We construct all "reality" in our minds. We decide that best practices work or do not work and then we resist evidence that conflicts with our preconceptions. Vijay resists metrics. Ram knows that certifications can be acquired. Manny looked at life as a journey that he could only

enjoy briefly. I look at all of this as an opportunity to make this a better world. I look at all efforts to achieve best practices as a small drop in an ocean of socially constructs. Together we are creating a new social order. To me, it no longer matters if Ram buys certification in best practices or if Greg ever adopts my recommendations. Best practices will happen. Those who swim with that tidal wave will thrive. Those who fight the current will drown.

Seeking the Sixth Deliverable

The sixth deliverable was the most complex. We spent a couple months on a proof of concept project and then we negotiated an addendum to our contract to fund two more resources. We proposed to do some of the work offshore but Luke wanted one additional person to be local. We gave him the pricing for that type of work and he offered to pay for one-half of one salary. We offered to bring someone on-site to work half time with the understanding that we would need to find other customers that would also pay him or her part time so that he or she would get a full paycheck every week. After a few days of haggling we agreed on a price and schedule. Meanwhile I was busy documenting the design and seeking ways to trim the scope and schedule to match our resources.

About ten days later our staffing team found a possible candidate for our half-time position. Vijay talked with him first and said that he seems to be an expert in this programming language, but he is a bit argumentative. Vijay noted, however, that experts of this caliber are often rather opinionated. This is both useful information and damaging information in that it prejudiced my own interview. The candidate's name is Brett. Greg and I did a phone interview with him a few days later. I found him to be argumentative, but he seemed knowledgeable. We need someone who is knowledgeable and can probably live with a bit of conflict if it is for a brief duration.

Three weeks later Brett showed up and I gave him his first work assignment. After a few hours I checked to see if he had access to the right systems and if the goals seemed clear. Later that day I asked if he had been able to read through our design document and asked if he understood what needed to be done. Brett explained that he was not going to read our documents and had already started coding. This distressed me because it had taken weeks of negotiation with Luke to get the subtleties of what was required down on paper. I probed to see if he understood the complexity of the requirements and found that he did not. I asked for an explanation about the code he had already written and found that he was coding based on assumptions that were not valid. Eventually I convinced him to glance through the design document and then use it as a reference on this project.

During the remainder of that week I checked with Brett and Mahesh a couple times a day to ensure we were getting things done on schedule. I continued the daily conference calls with Srinivas and Sanjay and used emails to coordinate the offshore and on-site efforts. This was a significant effort and we did not have enough staffing or time to do anything twice. I also worked with other project managers to find work for Brett so that he could spend half of his time on our project and half of his time with another customer.

Early in the second week there was an argument between Brett and Mahesh. I took them into a conference room and mediated. I asked each to explain their perspective. Brett explained his view while Mahesh listened. Mahesh started to explain his view and Brett stopped him and told him he was wrong. I explained that the rule for this process is that each person gets to explain their view before we begin the discussion. I asked Mahesh to continue and Brett again stopped the conversation. I explained again that we had given Brett a turn to speak and now we were going to give Mahesh a turn to speak. We tried a third time and failed again. It is my goal in

these situations to dampen the antagonism and get people back to polite discussion. Since the first tactic was not working I suggested we instead move ahead to the discussion of approaches. I offered to let Brett go first and he succinctly told us that we either do it his way or our project will fail. Mahesh reminded Brett that we had already spent time on the proof of concept project and had already solved the key issues that Brett said were pertinent. The discussion degenerated from there rather quickly and I asked Mahesh to leave while I discussed this with Brett.

This was the first of many confrontations with Brett. I told him that we had already pooled the knowledge of Vijay, Srinivas and Mahesh, explored the literature on the internet and scoured through the books that Vijay, Srinivas and Mahesh had purchased. We had code that now worked and all we needed Brett to do was expand on that code base in order to meet the additional requirements defined by Luke. He refused. I told him to stop work on that task and proceed to the next task on his list while I discussed this with Vijay. I also told him that arguments like this were not acceptable.

I called Vijay, explained the situation and suggested that first, Brett is not the expert that we thought he was and second that these types of arguments cannot be allowed to occur within six feet of Luke's cube. Vijay assured me that I was over-reacting and promised to get back to me next week. Brett then called Vijay and explained that I was giving him bad advice and told Vijay that he needed to do something about this. Vijay promised Brett that he would get back to him next week. We are at a standoff. Both of us know that Vijay is not going to get back to us.

I sometimes volunteer to work with homeless people. The key thing I have learned from that experience is that we all isolate ourselves from reality. One of my homeless friends is named Tommy. Tommy tells those he meets that he is a wealthy businessman who is dressed casual so as not to attract attention while he explores the neighborhood for a business expansion. This is his outer personae. If you penetrate that layer you next learn that Tommy is an out-of-luck businessman searching the neighborhood for a location that can use his skills. If you persist, then over a matter of months you will learn that Tommy was a successful businessman, has spent time working for other businessmen and is now frustrated by his inability to get the type of job he wants. Now, if you have patience you will learn that Tommy has gone through several jobs in the past few months and has a hard time holding onto a job because he tends to get angry when he does not get his way. If you can stay in touch with Tommy long enough you will learn that Tommy gets angry because he often ends up in prison when he gets into a fight with someone he works with and thus he does not trust the people that he works with because they might send him back to prison. Layers upon layers of projections all designed to protect the person who Tommy really is. There are so many layers, in fact, that I still do not know Tommy. Nor do I know Brett.

The key difference, however, is that I volunteer to work with Tommy but it is my budget that is paying Brett. When it is my personal money that helps Tommy find a place to stay for the night, then I do not expect anything in return. Somehow, however, when it is my customer's money that pays Brett then I expect him to do what we are paying him to do. Another key point is the impact on the ecology. If I help Tommy get off the street and into a job then it seems to me like the world is a better place. What, however, would I think if I was the employer that hired Tommy and then found him threatening to hit one of his customers? In that situation the priority is to protect the ecology of that work place and remove Tommy. Here I am in a dilemma. How can I improve the ecology of this work place when the people who shape this ecology are not getting along?

My first reaction was an appeal to Vijay that was rebuffed. My next strategy is to "turn the other cheek". I will accept Brett as an expert who is opinionated and try to change the work

assignments to avoid overlaps between him and Mahesh. It will be my goal to find Brett's strengths and help him focus on those areas. At the same time, however, I hope to show him alternatives. Perhaps he can learn if he understands that Mahesh is succeeding even while taking an approach that Brett rejects. After all, that was my approach with Luke. I allowed him to continue to tell us that offshore development never worked even as he moved the results of that effort into production.

The goal of that approach is internal dissonance. The Bible contains several references to a building block that was rejected. In brief, the quarry workers selected the very best stone and sent it to the building site to use as the stone that would have the most visibility. When it arrived, the number on that stone did not match any of the numbers that the workers were looking for so the stone was rejected. People continued to stumble over this discarded stone as the work continued. Finally, only one stone was missing from the completion of the building. Only then did the workers realize that the stone they had been cursing all those years was the most beautiful stone of all and the key to the completion of their building.

We do the same. Anything that does not fit our view of the world is rejected. My design document, Mahesh's code and Luke's requirements do not fit Brett's view of this project so they are rejected. Brett's personality does not fit my goals for the ecology of this work place and so I want to reject him. This is human nature. We grow, however, when we see the bigger picture and understand how the parts we had rejected fit into a better whole. Here is a key to best practices:

--

We need to teach people to integrate the parts they want to reject.

--

Stating this in another way, we need to help people mature from being focused on themselves to the place where they integrate multiple cultures into one concept.

The way to do that is to create dissonance. Mahatma Gandhi created dissonance by refusing to behave like a colonial subject should behave. Martin Luther King Jr. created dissonance by refusing to behave like a good colored person should behave. My goal is to hold the dissonance of Brett, Mahesh and Luke in close proximity and allow each to change the other. This will be a journey. We are going to need to seek not only the solution to this technical problem but also the solution to our relationships. If I can hold this together we just might not only build the customer's application but also help each other to grow in the process.

Six Scapegoats

There are six of us now locked into delivering this complex set of code - Vijay, Srinivas, Sanjay, Mahesh, Brett and myself. There is a lot of friction and I find it hard to hold this team together. Vijay vanished for a couple weeks. Once he finally showed up I convinced him to do a code review on Brett's code. The result was sadly predictable. The code is amateurish. Vijay politely explained that there were better ways to get the same result and tried to mentor Brett. Brett assured Vijay that he already knows the answers and does not need advice. Vijay persisted and convinced Brett to re-write his code. Vijay left the meeting frustrated and did not return again for nearly a month. Nor would he again participate in a code review. Vijay holds me responsible for the results, will not allow me to replace Brett and will not allow me to remove Brett from the project. I am the one to blame for this mess.

I, in turn, blame Vijay for not allowing me to fix the situation and I blame Brett for the results.

Luke has also grown tired of the situation and has asked me to remove Brett from the project, but I cannot. So Luke blames me for the arguments and slow rate of progress. Mahesh barely speaks to Brett and Srinivas has begun to complain to me as well. None of this is helpful. Brett is here for the duration of the project, whether we like it or not. We are all either going to get along or we will utterly and completely fail.

I am trying to mentor Mahesh. I reminded Srinivas and Sanjay that I am the sole point of contact for their work assignments. I keep sending emails to Vijay and I leave messages for him. And then I do my very best to be a real and present person while working with Brett. This is hard work. If I pretend to want to work with Brett while deeply hoping he will fail then my true intent will be transparent. Instead I must put myself into a mental state of desiring to help. Here I borrow from what I have learned in working with the homeless. I do not confront the discontinuity between their explanation of reality and my perception of the same. I do my best to instead see reality from their point of view. I am trying my best to see reality from Brett's point of view.

Meanwhile, however, the deadline is nearing and Luke is beginning to worry. The Arbinger Institute has an interesting concept that explains how our actions trigger our sense of frustration with others. (Arbinger Institute) Basically, Luke now fears that his decision to give us half a resource instead of the two resources we asked for was wrong. He fears that his action is now going to cause us to fail and thus there is a sense of guilt. Rather than absorb that sense of guilt, Luke looks around to find the person that causes him to feel this way. Then he projects that feeling onto the person who causes him to feel guilty. Thus, Luke has now adopted the point of view that it is my fault that I did not hire enough people.

This finally came out one day when he could no longer contain his hostility toward me. He asked me to stay behind after one of my status meetings. Then he explained to me that his company has lots of money and that we should have asked for more money so that we could have hired more people. I reminded him that we had asked for more people but he denied that I had done so. He then told me that he is disappointed that Brett is not the expert we had promised. I sympathized with him but reminded him that with the updates I have made to the schedule we are still projected to finish on time and on budget. He then told me that it is my fault that this project is failing. I assured him that it is not failing. It degenerated from there and again he reminded me that project managers do not do anything of value. I attempted to salvage the situation but I am truly frustrated. I sense that Luke is afraid he will lose his job if this project fails. I sense that Luke is afraid to tell his boss that he withheld funding when we asked for it. And I sense that I am now to be the scapegoat for what Luke believes is going to be a catastrophe.

The Arbinger concept of self-deception says that I first choose to act in a way that I know is short of what I could have done. This makes me feel guilty. Then I project that guilt onto those around me. They sense my attitude toward them and they treat me accordingly. This then spirals into chaos. My project is on the edge of collapse into destructive conflict. It is important that I not reciprocate Luke's projections. Instead, I feel sorry that he lives in fear. Personally, I completely sympathize with him. I also feel like my career with this company will end if this project fails. The difference is that I, like Manny, only consider each job to be a brief stop on my journey. We are all going to die anyway. What matters is what we do with the brief span of time we are given.

As for Brett, the reason he behaves like he does is three fold. First, he has gotten away with it all his life and is going to get away with it here as well. Second, he makes a lot more money

than any of the rest of us even though he is the youngest person on the team. He does that because he projects himself as being massively superior to the rest of us. Deep down I think he knows that his behavior is a product of those two tributaries. What I do not think he realizes, however, is that he may well have been an expert in relation to other people on other projects, but here he is dealing with people who in a few months have surpassed him.

I took some time and went over to see Greg. I gave him a candid update and told him just how fragile the team is and how great the risk is that we might not finish on time. We then talked about Luke's fears and his use of me as a scapegoat. Greg told me he is running into exactly the same issue now on another project where that customer has decided that all the past successes were due to their talent and the current difficulties are entirely Greg's fault. It does not matter that all of the coding on all of those efforts were performed by our programmers, not theirs. It does not matter that we have had the same project manager in place for almost two years and he is the person responsible for all the good results so far. Instead, all of the good is now credited to the wisdom of the customer and all of the bad is attributed to Greg. Scapegoating is a very popular practice among people who live in fear. Ultimately the deaths of Jesus, Gandhi and King are all the result of people who were afraid of what might be. Poor souls projected their internal fears onto those three saints and killed them rather than confront their own internal demons. This is human nature. We cannot tolerate internal dissonance and so we display external antagonism.

The next several weeks became intense. As a team we worked a lot of twelve hour days. Mahesh found solutions to numerous problems. Srinivas took over as the subject matter expert and found solutions for the problems that Mahesh did not have time to deal with. Vijay started responding to some of my emails and offered technical advice on a few key issues. And Brett was progressing. I had given about a quarter of Brett’s work to Mahesh and another quarter to Srinivas. That left the load balanced so that our expert, Brett, was finally able to keep up with the new hires. We never got out of the "storming" stage of team formation. There were more arguments including one spectacular one on a day when I was off at another site.

And then one day we finished.

Lessons Learned

We missed the go-live deadline by four hours on a five month project. We came in $300 below on a $300,000 budget. I felt like this was a fantastic triumph. We missed the budget by 0.1% on the plus side and the time by 0.5% on the negative side. Considering how close we came to total failure, finishing at all was remarkable. And finishing that close to the baseline was amazing.

Luke, however, was furious about both us taking too long and not spending as much as was budgeted. I called Vijay and Greg to warn them that Luke was going ballistic. I also told them, however, to just give it a week and he would calm down.

The following week Mahesh and I were invited to the quarterly meeting for the division. The division executive talked about some of the big efforts underway and then he introduced Luke. Luke gave a presentation describing the success of his project and then he asked Mahesh, Kathy and me to stand up and be recognized. Out of our band of six scapegoats Mahesh and I were the only two present so I sent a nice thank you email to the rest of the team and copied Greg and Greg's boss. I included a good description of the praise that the division executive had for our team and our company. Both Greg and his boss responded with praise for the team.

The contract was extended for twelve more months. And with that I had finished my work. I had demonstrated that we could deliver using a mixture of on-site and offshore programmers.

Soon Greg sent me off to another customer. Brett learned to ask Mahesh for advice, though he seldom used it, and then he too was transferred to another customer. Mahesh took over from there and will run the project going forward. The key lessons that I learned from this project can be summarized in three points:

--

We must consciously choose to either inflame the chaos or absorb the insults and dampen the antagonism.

If we can hold the chaos in check we can create dissonance that will change people.

If we find a way to help people change we just might finish the project.

--

Bibliography

Arbinger Institute; Leadership and Self-Deception; BBC Audio; ISBN 9-781572704442.

Brooks Adams; Theory of Social Revolutions; MacMillan; 1913.

Carl R. Rogers; On Becoming a Person; Houghton Mifflin; 1961; ISBN 0-395-08409-1.

Charles Darwin; On the Origin of Species by Means of Natural Selection; Librivox; MP3 recording; http://librivox.org/the-origin-of-species-by-charles-darwin/.

David A. Lax and James K. Sebenius; The Manager as Negotiator; Free Press; 1986; ISBN 0-02-918770-2.

David Bohm; On Dialogue; Routledge; 2004 ISBN 0415336414.

Division of Christian Education of the National Council of the Churches of Christ in the United States of America; The Holy Bible, New Revised Standard Edition (NRSV); Holman Bible Publishers; 1989.

Goldratt, Eliyahu M. and Cox, Jeff; 2004; The Goal; The North River Press; ISBN 0-88427-178-1.

Goldratt, Eliyahu M.; 1997; Critical Chain; North River Press; ISBN 0-88427-153-6.

Goldratt, Eliyahu M.; 1994; It's Not Luck; The North River Press; ISBN 0-88427-115-3.

Frederick P. Brooks, Jr.; The Mythical Man-Month; Addison-Wesley Publishing; 1982; ISBN 0-201-00650-2; page 25.

Jim Collins; Good to Great; Harper Audio; 2001; ISBN 0-06-079441-0.

Kenneth Paul Kramer with Mechthild Gawlick; Martin Buber's I and Thou; Paulist Press; 2004; ISBN 0809141582; page 131.

Martin Buber (author) and Ronald Smith (translator); I and Thou; Scribner; 2000; ISBN: 0743201337; page 109.

Peter M. Senge; The Fifth Discipline; Doubleday; 1990; ISBN 0-385-51725-4.

Project Management Institute (PMI); A Guide to the Project Management Body of Knowledge (PMBOK) 2004 edition; PMI; ISBN 193069945-X; pages 172-176.

Robert Fritz; Corporate Tides: The Inescapable Laws of Organizational Structure; Berrett-Koehler Publishers; 1996; ISBN 1-881052-88-5.

Spencer Johnson; Who Moved My Cheese?; G. P. Putnam's Sons; 1998; ISBN 0-399-14446-3; page 32.

William Isaacs; Dialogue: The Art Of Thinking Together; Currency; 1999; ISBN 0385479999.

#####

If you enjoyed this book, please contact Robert E. Perrine at http://www.robertperrine.biz. Also, please logon to Smashwords.com and check for other titles by Robert E. Perrine.

Summary of Best Practices for Information Technology

The theme in this book is the need to align all the efforts in an Information Technology shop. The chapters describe the need for vision, best practices for projects, operations, governance and how to measure the results. The book is filled with short stories about how things really work today and how they will be better tomorrow.

Coping Styles: Dealing with Life on Life's Terms

Coping Styles makes concepts from organizational psychology relevant to peoples' daily lives. This book describes nine ways we cope with life's difficulties and the nine ways we learn to help one another.

Growth Rings: How We Get Connected

Growth Rings makes concepts from developmental psychology relevant to peoples' daily lives. This book describes the eight stages of development and shows how those stages influenced key historical figures. It then links developmental psychology with current legal topics.

Survey of HR for Projects

The theme in this book is the human relationship aspects of project management. The chapters survey the best known models of organizational psychology, leadership and organizational management. The concluding chapter describes an integrated model of organizational psychology and explains how this is relevant to project management and team formation.

Presentations on the Critical Path Method

This book contains a series of presentations on the Critical Path Method. It includes a bonus presentation on Earned Value. These lessons are for people learning more about the mathematical aspects of project management.

Communication Deficiencies A Case Study in Project Management

In this fictionalized case study, Robert Perrine describes what a typical project looks like within information technology. The project is already behind schedule when Robert arrives. There is no agreement on scope and the project budget is being siphoned away. Somehow Robert needs to bring the team together and get the work accomplished before they run out of time, money or sanity.

How Big Is Your Fishbowl?

How Big is Your Fishbowl is a short story about the adventures of Marlene and Bob. Bob leads Marlene to tranquility. Marlene finds the path with many steps. And Linda stops eating long enough to find out what has Marlene so excited.