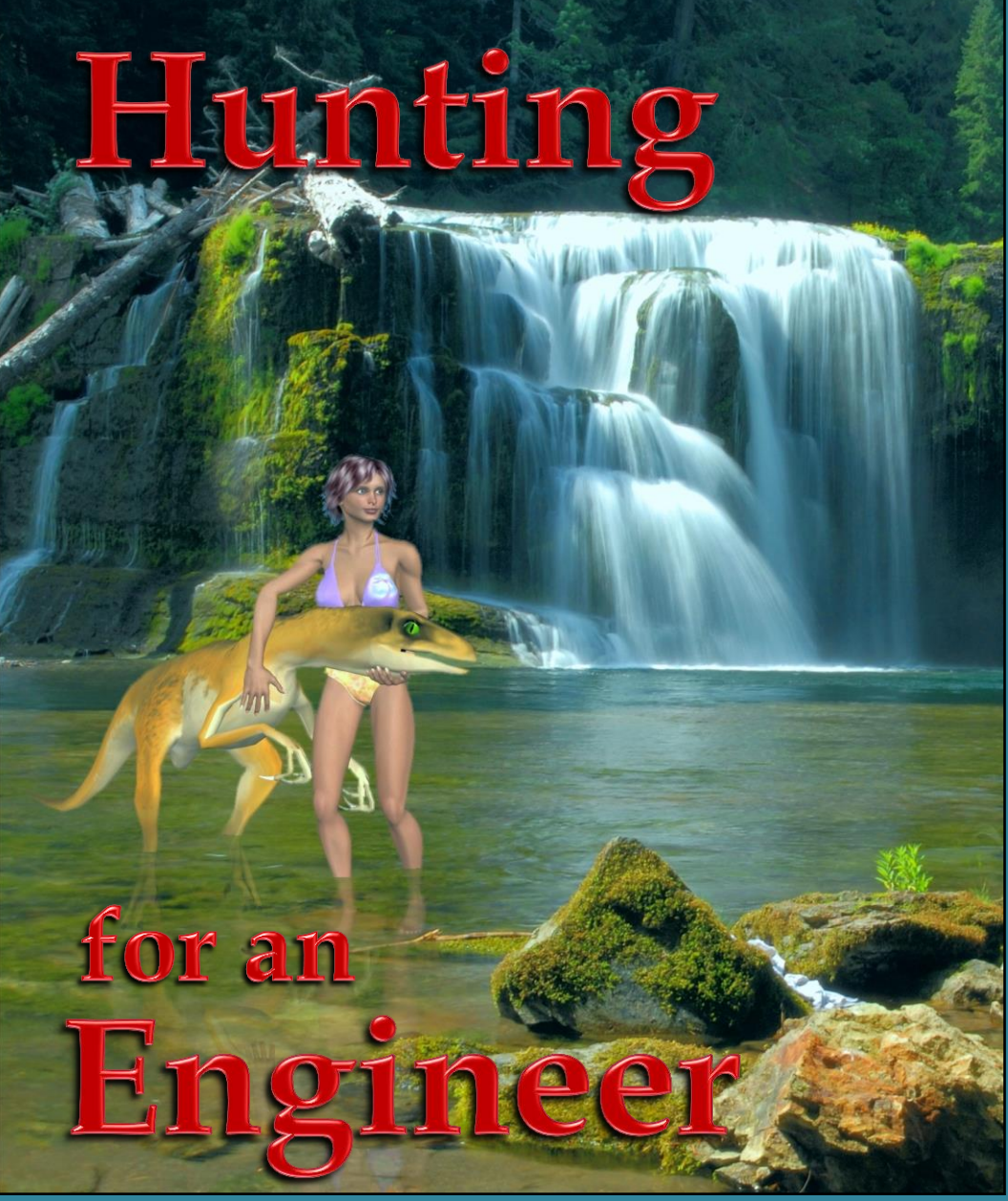

Hunting for an Engineer

A light-hearted look at business problems from the everyday life of a process engineer.

M. B. Henry

Title: Hunting for an Engineer

Subtitle: A light-hearted look at business problems
from the everyday life of a Process Engineer.

Author: Mike Henry
Ghost Town Software
Vancouver Washington

Artwork: Mike Henry

Genre: Process Improvement

Dewey Number & Library of Congress Number
have not been assigned yet

Web Site: www.GhostTownSoftware.com

On the Cover:
Jessie hunts for big game with her trained raptor.
Cover design, photo & artwork by the author.

Detailed Table of Contents:

Preface

There is nothing groundbreaking in this book. It is simply my observations and stories about businesses passing or failing on the three rules of execution that are listed below. Plenty of other books talk about these same ideas, but life in the real world is never quite as rosy as most authors want you to believe.

And that is the difference here. These ideas are described with real-life anecdotes and examples that hopefully explain why these ideas don't always work the way people think they should.

The stories and solutions in this book mostly take place on the various production lines that I've worked on over the years, but the lessons don't just apply to the production line, most of these ideas will work just as well in the office and at home.

All of the stories in this book are true but none of the businesses are identified and all of the people's names have been changed.

Introduction

A few years ago, during one of our quarterly all-hands meetings, the VP asked us: " We had a hundred thousand dollars in process improvements last year, why don't I see a hundred thousand dollars more profit this year? "

This is an embarrassing question that gets asked by managers everywhere. Documented improvements don't turn into measured improvements. He wasn't asking anyone in particular, but as a manufacturing engineer he could have been asking me. I thought about that over the years because I had been suspicious too. Even though the math added up on these improvements, I didn't really believe they would have the impact that the documents claimed.

It was just a gut feel and I couldn't really put my finger on it, but I knew something was fishy about their process improvement techniques and calculations of cost savings. Over the years I saw others making the same mistakes and listened to their horror stories, and I slowly realized what was going on.

Two things were happening. First, the labor hours on paper are not the labor hours on the shop floor. That might sound obvious but the reasons for the discrepancies are not obvious. Second, everywhere I've worked and from stories I've heard from others, the people on the shop floor are not being used effectively, negating a lot of the expected improvements.

There are dozens of books that describe process improvement and team building techniques; some are very useful, and others not so much. Most of these books describe a lot of the same techniques but with their own personal twist. However, most are misleading about the effectiveness of these techniques.

For example, techniques such as 5-S and 5-Y are tools you can use to improve your workplace. They are not goals. The goal is to reduce cost and increase quality, and 5-S is a good tool to help you get there. Yet, the last few places where I've worked, they had put more effort into charting and enforcing 5-S than in verifying process improvements.

That's the point of this book. Businesses everywhere are doing a poor job of running their production lines and don't understand why they're struggling. The stories in this book describe a lot of those struggles as I've seen them over the years. Hopefully these are the same stories that people are seeing elsewhere, and will lead others to similar conclusions on their own production lines.

Books everywhere espouse their own particular theories on how to best run a business, but the bottom line in every case is simply: Execution. Somehow people will have to get the job done with efficiency and quality. Finding the best people to put in charge is a good idea, but to run a business profitably, your people will have to deliver regardless of who is in charge.

In the course of writing this book, I came up with three rules for running a business.

Rule #1:	Open up the communication channels.
Rule #2:	Find your best people and put them where they will do the most good.
Rule #3:	Use process improvement tools effectively, according to cost vrs benefit.

These three rules are the theme through out this book, nearly all examples are based on an event passing or failing one of these rules.

Chapter 3
Maximize the Managers

The first rule of being an entrepreneur is to surround yourself with people who will make you rich. Whether you outsource talent or hire people directly, very few people strike it rich these days without a whole lot of help. It's the same when running an office or production line. Every manager is only as good as the staff, and developing that staff is Job-One for every manager.

Most supervisors and managers are pretty good at this. It doesn't take a test or database to figure out who the best people on the team are, the astute leaders can spot the next leaders as soon as they walk in the door and they're quickly bumped up to important positions that need someone with advanced talents.

These are the people eager to learn and deliver quality quickly. They get the big picture and don't wait to be told what to do every step of the way. They ask a lot of questions that tell the boss they aren't afraid to do things the right way. It doesn't take long before they are the ones answering questions and solving problems for the others.

When we transferred my production line to Los Angeles a

few years back, we had to evaluate a lot of people for a variety of positions, quickly. We did this by bringing in new recruits two at a time each day for training. This schedule allowed a small number of trainers to evaluate a large number of recruits by ensuring the majority were past the basics. This also allowed us to evaluate the majority on most of the stations. By the end of the month, the youngest person of the group was named assistant-lead and one other was bumped up to technician (as mentioned in the previous chapter).

Note that the young lady who was bumped up to lead was not the best worker. Others were faster and had higher quality, though she was still very good. She was bumped up because she could do every step of the process and understood how the paperwork drives the whole process. This is a key concept of evaluating talent: if you are looking for your next leader, evaluate leadership skills.

This is harder than it sounds. People working on a production line or in an office generally aren't there to lead, they're just there to build a product. When the training database lists only the tasks of their job description, how will you document the advanced tasks of potential promotions. The reality of the situation is, the training database is a poor place to find your next leaders. Evaluations for annual or merit raises are best experienced in person. It seems like a five minute conversation is all it takes to spot a winner, or a loser.

But that is where it gets tricky. Some of the best workers, even those who are great at everything, are possibly terrible leaders. For example, it's not unusual for a manufacturing genius to be too reclusive or abrasive to be supervising others.

A classic example of that was my early career. I was the top ranked technician everywhere I worked. But I was one of those quiet kids who didn't like interacting with others. I was a good team player who got along with everyone, but I was just as content to ignore everyone and made no attempt at being a leader. I eventually out grew those problems, but early on no-

body would have made me a supervisor.

Someone will always stand out, in good times or bad, by instinctively taking charge when problems arise. This happened several times in my career when the boss had left suddenly, leaving a leadership vacuum in the organization. Each time, I felt like a load was lifted off my shoulders and I took on a bunch more responsibility to fill the gap, and ended up getting a rank increase or promotion each time. I had a reputation for getting things done, but I was still a lousy prospect for a supervisor.

Unfortunately, it is also common for some to make their way up the ladder just by visibility and grandstanding. This is a tough call because sometimes it's tough to tell when someone is showing remarkable initiative or in fact causing more harm than good.

This is what happened recently at a place where a friend of mine works. A new guy had some database experience and created a log for production data. The supervisor loved it and pulled him out of production and into a cubicle. However, when I.T. saw how poorly designed the database was they pulled the plug and the guy was sent back to the production line.

I don't know if that was a happy ending or not. I have a programming background and I'm fairly critical of ugly databases, particularly by the I.T. department. I.T. majors seem universally clueless about proper user interface design. But their gripe in this case was data integrity, a common problem with amateurs. However, even though this guy's skills were misguided, he took the initiative to create a useful product, and maybe training and mentoring would be a useful way to increase his responsibility a step at a time.

A second interesting problem arises when someone gets in over his head like this. Others in the organization were also taking on extra responsibilities but without the fanfare, and

were getting no recognition. How did this new guy get singled out by the boss for creating a marginal product? Simple: he created a product that was useful to the boss.

At a previous place where I worked, employees were ranked against each other and some of the very quiet workers complained about their low ranks. So I pointed out to them, to get a high rank you have to be (1) good, and (2) visible. And there is no better way to be visible than to create a labor saving device for the boss. This is what happened in the previous paragraph, much to the chagrin of the other hard workers who had been turned down yet again for a pay raise.

This story just reaffirms how important it is to get involved with the employees. It's not only useful to identify who is the best, but also who is the best at what. That is how you determine where each person belongs on the line, and also who belongs somewhere more advanced.

The Engaged Employee

A few years back, I helped setup two production lines in our China plant. This is a real challenge but surprisingly enjoyable. One of the challenges is communication within the plant, not so much English vrs Chinese, but supervisor vrs worker. This is vertical communication, and in the China plant it only went down, not up.

The first chapter described the situation where the manager insisted on shipping bad product. The workers didn't relay test results back to the manager, resulting in a short quota, and that's what happens when communication is only going down.

But a different situation unfolded a few months later in that same plant. Our circuit boards had to be baked in an oven

then electrically tested, which took about ten minutes. After about a month of this, one of the girls on the line came up with the idea of testing before baking, then the boards that pass don't need to be baked. She relayed this suggestion to the Chinese engineer, who proposed it to me.

Unfortunately, I had to say no. I showed the engineer the note on the customer drawing that required the boards to be baked regardless of test results. But I was thinking this was a surprisingly bold action on her part - production workers in China don't usually initiate conversations like this. And this was good. This is how we want workers to behave - come up with process improvements, even non-starters, then bring them to the lead or engineer.

This type of behavior is worth noting for future assignments, here's a person who is taking personal responsibility for the process. Drop her into a train wreck and see if she can unravel it. But unfortunately, this behavior isn't as clear as we would like to believe. Half of the people who speak up like this aren't doing so because they're the ones coming up with the brilliant ideas, it's because they're the ones not afraid to talk.

This was the situation with the lady in the previous chapter who was almost let go because of her poor English skills. She never made process suggestions because of her language barrier. But after working with her for a while I realized she wasn't just a robot, but had a very good understanding of how the mechanics of our operation work. She knew that certain process adjustment would or would not work, and I began to ask her opinion more frequently.

Does it really matter whether the person suggesting process improvements does so because she is the one who sees solutions, or rather, because she talks a lot? What if she's simply a chatterbox who submits an idea every day for a month, but 9 of 10 are duds? Should the boss throttle her back at that point? A better idea would be to engage her. Find out why she's missing the boat on 9 of 10, rather than trying to fig-

ure out how to quiet her down.

These big ideas fail for one of two reasons: either the person doesn't understand the process, or he doesn't see the big picture beyond the process. In the case of the Chinese lady above, she had no way of knowing that that note was on the customer drawing. But she clearly understood the process - she knew that we were baking the moisture out of the boards so they could pass an electrical test. Nobody told her that, she deduced it on her own and then took the additional step to think up a better way. Contrast that to several others in that same room, who were leaving the oven door open so they could load and unload faster. They clearly do not understand the process.

I like to visualize 3 levels of involvement in the process:
Level 1, a person sees a problem but does nothing.
Level 2, a person sees a problem and tells boss.
Level 3, a person sees a problem and suggests a solution.

The young lady above scored a 3. With a little coaching maybe she'll come up with a real breakthrough next time.

There is a wide variety of personalities on the line and some people like getting involved and some don't. In the China example, just above, that particular engineer was fairly tough on the workers and I was surprised the young lady had the nerve to present a suggestion to him. But regardless of how easy going an authority figure is, some people won't bring up problems or solutions.

The story in the previous chapter, where the entire production line ignored an obvious problem, is far more common than managers would like to admit. Probably half of test failures were caused by a problem that someone should have seen, yet they either didn't see it or they didn't want to say anything. Both problems need to be corrected, but the wrong way to correct it is to demand that the people on the line speak up.

Some just won't speak up, and some would have until the

boss yelled at them to speak up. This is in fact what was happening in the episode above. The engineer in that area was very nice but was also very stern when her instructions weren't followed. This was a very difficult product that resulted in frequent admonishments. This is how you generate cycles of people afraid to speak up and then more admonishments and then they're even more afraid.

At one point, at that same plant, I convinced one of the women on the line to try prototyping. We didn't have prototype technicians and I wanted to ramp up a few, and she was eager to give it a try. But the engineer in the example above snagged her first.

After about two weeks they seemed to be floundering, so I asked her how she liked doing prototypes, which to my surprise she said: "I don't want to do prototypes anymore! Angie is mean to us, I thought we were going to work with you!"

It was hard not to laugh at that, but the prototype team was a disaster. Prototyping and engineering need to work together very closely, and that just wasn't happening. That particular engineer tended to deliver instructions in a lengthy whirlwind that left people scratching their heads. Two months later the team was disbanded and the prototype functions were reverted back to the engineers.

This was a common reframe during those years - they wanted to work with me because I'm calm and bring stability to the group around me. They were leery of working with some of the others because they didn't know what they would do wrong next.

I always made a point of fixing people's problems without the theatrics. I was more interested in getting them trained correctly so they'd do things right the next time, and less worried about pointing fingers. This was the situation at an even earlier place I worked. I was training a new doc coordinator and I let her use my computer to upload the new process doc-

uments to the server. This was a lot more difficult than it sounds, and I told her not to send them until I looked it over.

But she sent them and they were wrong. I got an angry voice mail from the documentation manager because I should have known better and I let her listen in. But I never told the manager that she sent it. That wasn't important to me - I needed her to know that we're a team, for good or bad. Either way, she knew better next time.

Situations like those above bring up a related question: Is it more productive for a supervisor to be a micro manager or a hands-off manager? The obvious answer is that extreme behavior in either direction is a bad thing. But that doesn't mean that the correct answer is right down the middle, it means every supervisor, employee and situation is different. And every answer to that question will be different.

Individual personalities will want to manage with different techniques. But even more important is the fact that different employees will require a different style from the manager, and it is very important for the manager to know which employee is which.

Job One for any manager is to identify and maximize his best talent. A micro-manager is not doing that and neither is a hands-off manager. New people on the team, or a new manager, will require closer supervision, but as the people demonstrate proficiency they can be slowly turned loose. The goal is to create a team that requires nearly no supervision, but it takes thoughtful evaluation of individuals to get there.

The Caustic Supervisor

One situation that was hinted at several times is the effect of a caustic supervisor. There are a variety of personality types

that can be caustic to an organization, but the most common is the drill sergeant manager who feels that tyrannical behavior is what drives the organization. Deviations from his plan need to be crushed with a high volume display of authority.

This isn't as common as it used to be but I've still run into it a few times recently. Most organizations these days have recognized the destructive nature of this situation and actively weed it out. But it's worth going over the details because a lot of organizations don't realize what's going on, and there seems to be a resurgence of this behavior lately.

This isn't referring to predatory or psychopathic personalities, but merely the overly aggressive types who think their belligerent style is how you get results. A lot of these people are conditioned to behave in this way - they see the technique is effective because they see results when they yell at people. But these results are short-lived. In the longer term, results disappear because this atmosphere generates a slew of problems that are swept under the rug in fear, and process improvements disappear.

Drill sergeant managers wreck team dynamics in an organization several ways. First, and most obvious, this sort of atmosphere wrecks the communication channels. All businesses live or die through effective communications. Efficient communications will head off trends before they become problems, and when problems eventually happen those in charge will know about it as quickly as possible, allowing the problem to be corrected before becoming a catastrophe. Catching a mistake the minute it happens is a huge benefit compared to catching it the next day, particularly if yesterday's mistake had shipped to a customer, such as described in the last chapter.

People at all levels have a feel for the business disruptions that happen endlessly in an organization. A glitch in the database or a gripe from the customer are common occurrences that people handle every day without a panic. But in a caustic environment people are afraid to speak up, and won't if a mi-

nor event can be safely swept under a rug somewhere. No one is going to bring bad news to a caustic manager and risk an explosion. People will hide problems, then either bring it up to a trusted manager or just hope the problem is in someone else's hand when sarge finds out.

As mentioned earlier, this situation was happening repeatedly in a plant where I worked a few years ago. No one was caustic or belligerent, but the people on the line were intimidated enough that they would hid problems until I got there because they knew I would calmly fix their problems without laying blame or throwing a temper.

Sometimes, laying blame is all it takes to disrupt these communication channels. The engineer I refer to above wasn't caustic nor belligerent, he was merely brisk. He was my mentor when I started working there, and almost immediately I was a hero on the line for getting things done without any theatrics.

This clam-effect happens at all levels of the organization. Just because someone has a degree or is on the management team doesn't mean they'll tolerate verbal assaults or profanity. When a worker is intimidated, they'll stick to their job description and forget about minor problems or process improvements. The organization seems to be running smoothly, but in fact problems are being hidden and improvements don't happen. Then when this system starts to fall apart, someone will do a study to find out what is going wrong and the results will point at the people not talking and hiding problems, rather than identifying why they aren't talking.

Annette Simmons talks a lot about the effects of a wide range of problems like this in her book Territorial Games. (see Note 4) She also describes possible solutions to these personalities, something I'm not qualified to do in this book.

Later in this book I'll talk about doing a root cause analysis. It's a great way to dig deeper into a problem to find the un-

derlying cause, and it also applies to finding out why communication doesn't work. As a rough guideline, when people aren't communicating, the root cause isn't that they are bad, but rather there is something deeper wrong in the organization.

This same effect happens everywhere, not just at the workplace. Talk shows on TV seem to encourage people to argue to bring problems out in the open. But arguing is not communicating. It causes people to avoid communicating the next time a problem arises, resulting in an even bigger explosion with each cycle. Communication and teamwork is the ability to work through problems together without laying blame or intimidating. Everyone involved needs to close the situation feeling confident in the solution.

The second way this wrecks team dynamics is by driving away the best workers. In any organization, the best workers know they can get rehired quickly elsewhere. This is a particular problem for the salaried staff. No one with a degree is going to tolerate being treated like a farm animal, and job opportunities are everywhere for those with a degree. In addition, these are the people who are the most difficult to replace, those with deeper knowledge and a special skill set.

This was the situation in an organization where I worked a long time ago. Our sister plant had a very caustic manager, and in my first two years they had a 100% turnover of their maintenance staff. The cause of the turnover wasn't even a mystery. Even the management team was afraid of this supervisor and let him run roughshod all over half of the department.

More recently, a friend of mine was in a situation where a caustic manager triggered a sudden exodus of middle managers. He started replacing the outflow with his caustic friends, and began demanding that the meek become caustic too.

It was an absurd situation. Some of his business plans were good and some were bad, but nobody was willing to tell him

which were which. Profits took a nose dive and naturally it was everyone's fault but his.

Actually, a few people did tell him his plans wouldn't work, but he refused to listen. He was eventually demoted but not before nearly wrecking the place. When analyzing a situation like this, we need to be careful when attributing cause.

Was this person a disaster because:
(a) His caustic demeanor was wrecking communications?
(b) His reign of terror drove away important staff members?
(c) He didn't listen to expert advice?
(d) He was simply a bad businessman?

The answer is all of the above. Fix any one of those and the mess probably could have been salvaged. But in particular, regardless of a person's demeanor or his business incompetence, if he would simply listen to the advice of those who obviously know what they are doing, he could have been successful.

When salvaging this situation, what would be the fix? Putting the management team back in order would be a top priority, but how would top management stop this from happening again in the future? Getting rid of the problem person doesn't prevent the next guy from going haywire.

This is an important question in my mind. When I look at this from the point of view of an engineer, I have to ask, what was the root cause of the problem? This guy had disastrous business skills, but that was not the root cause. The deeper cause was that the plant had no policies in place that would allow people to stop obviously bad decisions.

This is the question businesses everywhere need to ask themselves: If the new plant manager demands that people implement plans that are known to be a failure, what can the others do about it? They won't go over his head if they assume he has blessings from the corporate office, they'll simply leave before they get blamed for the mess.

Numerous studies have shown that people tend to leave an

organization because of personnel problems as often as for money. Not only is this a detrimental brain drain, but it leaves behind a lot of new hires, an inexperienced group groping their way on a crippled team. This turn-over costs a company big money in productivity and delayed process improvements until the new guy can figure out the basics.

All of the above is what happened at the plant where my friend worked.
* People did nothing because they assumed the problem had blessings from corporate.
* Those who could, left, those who couldn't clammed up.
* A new VP was brought in for three months to clean up the immediate problems.
* He did not fix root cause and when he left everyone reverted back to self-preservation mode.

In fact, tyrants through out history tend to have very short dynasties because their top people revolt. When Qin Shi Huang united all of China in 220 BC, he was the most powerful man in Asia, yet his dynasty was over in less than 20 years. He was followed by the Han dynasty which lasted about 400 years. This same story was repeated with Alexander, Attila, Genghis, Napoleon and plenty of others. These were some of history's most powerful people who led with an iron fist, yet led their dynasties to ruin. In each case it was because they were generating enemies faster than they could eliminate them.

The third way the all-powerful leader wrecks team dynamics is when the people in these organizations have to make big decisions. It's difficult to propose a bold out-of-the-box strategy even when surrounded by a supportive group, but no one is going to stick their neck out if they know they will be ridiculed. They tend to base their decisions on trying to read the boss's mind and not getting yelled at, rather than delivering a robust solution.

People everywhere think the bulldozer method will drive the team to push themselves harder, and it does, but it also

causes people to stick to the canned solutions that are known to work. Nobody is going to suggest a revolutionary paradigm change if they are afraid to even ask for a potty break in one of Attila's meetings.

This often creates a subtle problem by generating excessive paperwork. As communication drops off, these managers tend to spew forth reams of logsheets and databases in an effort to find out what's going on. This adds an extra layer to the communication path, which slows down problem resolution even further, prompting yet more hysterics.

It's been my opinion for a long time that excess data gathering could be a sign of lost management - they know something is wrong but have no idea what it is or why. So they try to document everything they can think of in the hope that something will jump off the paper.

This situation is described in the next chapter, where the plant installed an expensive data logging system that was intended to tell them very basic process information. This was information they should have known if they would just get out of their cubicle once in a while.

Frequently, the real solution to a lot of the vague problems on the production floor is to simply improve communication, but communicating through paperwork isn't the answer. People need to get out on the floor and get to know their people and their process, and encourage open channels of communication.

There have been plenty of dictatorial managers over the years who were very effective, but most are not. Because they stifle teamwork and communication, these managers are effective only if they are organizational geniuses. And when they are effective, their success will generate plenty of followers who find out quickly enough that they are not the same genius.

These days, this is called The Steve Jobs Syndrome:

"Jobs was a dictator who did great things, so I need to be a dictator to become great too."

But in fact, being a dictator had nothing to do with his success, he did those things because he was a visionary genius. If he had calmly worked through the details with his team, he would have gotten the very same results.

When assigning a cause to situations with a lot of variables, people tend to pick and choose the variables that fit their own personal ideology. They develop their ideology first then find facts to support it later, rather than using facts to draw a conclusion. The people who did this during our annual football pool were just throwing away their money. The people who win the pool were always the ones who based their selection on statistics, rather than finding statistics to support their favorite team.

Here is a simple exercise: List ten things that made Steve Jobs great. Now have ten people pick the trait that they think was most important. Did everyone pick something different? Did they each pick something that matches their own personality? The one I picked is the opening line of this chapter: he surrounded himself with people who made him great.

The High Performance Team

In his book Leadership Gold (Note 5), John Maxwell talks about working with eagles and ducks. Eagles are the people who get things done without being told and fix problems before anyone even knows there is a problem. These are the people you want running your organization. Ducks are people who just show up and do their job, without any drive beyond that. These are frequently good people, but they aren't going anywhere in the organization. It's an interesting concept, but Maxwell, and nearly every other author, needs to take that idea

a step further.

The main theme of this book is communication, and one of the biggest barriers to effective communication is a stonewall or caustic manager. A caustic manager can turn an entire team into a room full of ducks. People who were highly effective yesterday suddenly clam up and rush through their assignment to get it out of their hair as quickly as possible.

When a problem appears on the road, it is swept under the rug. Nobody is going to bring up a problem to a caustic manager, and risk his wrath or extra work. They feel no commitment to helping this guy out. If they are smart, the team will do just enough to keep the manager happy, and he'll go on thinking he's doing a good job. But more likely, the project is a dud and the obtuse manager blames the ducks for the mess.

This is a shortcoming in most management books. They talk a lot about elevating individual performances but forget about elevating team dynamics. Some books actually encourage behavior that could wreck a fragile team. Rule One in almost all management books is to weed out the low performers. The idea is misguided. Someone needs to evaluate why a top performer elsewhere is not working on this new team.

Google recently researched this exact problem. Google can afford to hire nothing but the best, but still, they were finding that some teams got good results and some did not. They quickly found out that individual high performers had no bearing on the team outcome. A top performer leading a group was no more likely to achieve a breakthrough than a group of surfing buddies. (see Google, Note 6)

Google brought in a few researchers who studied these dynamics for two years and came to the startling conclusions:
A. High performing individuals do not equate to a high performing team.
B. Teamwork and inclusion do equate to a high performing team.

I could have saved Google a million dollars if they would have just asked me first. This effect has been know for years - it's called Buy-In. When everyone feels they are an important part of the team, they put their best effort forward. When they feel that they are nothing more than carrying the banner for someone else's glory, they do enough to not get fired.

When the team becomes the family, each person wants to ensure that the family becomes successful and takes personal responsibility in the final result. The family thrives on wild ideas and late evening pizza, and is secure knowing the rest of the family will help them out of a jam if needed. They're all committed to the same big picture.

Cogs don't do that. Cogs have a real life elsewhere, and their life as a cog on someone else's wheel never goes beyond the wheel. These are smart people as often as not, and smart people don't want to be a cog, they want to be an important part of a team that does important things.

This idea brings up an interesting corollary: What is the difference between Plan A and Plan B? There are three general answers:

* Plan A is the plan that will generate most profit or least risk.
* Plan A is the plan that we can implement right now.
* Plan A is the boss's plan. Plan B is anyone else's plan.

These are all perfectly valid and all good plans have a little of each. But I'd like to throw a wrench in this idea, based on the Google research above.

Rule One of strategic planning is, "All strategic plans will live or die based on their tactical implementation."
This means, the most successful plan will be the one pushed the hardest by the people implementing it, which is the one with the most thorough buy-in.

This means, the plan that reduces cost by 50% might not out perform the plan that reduces cost by 30%, if no one takes it seriously. Note that the people who have to buy in to the

plan are both those who develop the plan and those who will have to use the plan. If either group doesn't buy in, then it's a good idea to have Plan B ready in the wings.

This was the case at an organization where I worked recently. They had several process improvement databases that seemed to be getting nothing done. The main one was web based with the idea that anyone can add to it from anywhere. This was a beautiful idea, but it wasn't nearly that simple. Entering data was far too complex and participation in the weekly meetings was dismal, resulting in conclusions taking absurdly long. For example, as I was browsing the listing I came across one entry that was a year past due but listed as "on schedule."

This is a prime example of near zero buy-in. The database needed one more process improvement entry - to fix its self. The system needed to be redesigned around simplicity and zero effort, and the team needed to be made up of people who wanted to be there. About the time I left, this system was scrapped, but I don't know what replaced it.

Note that buy-in isn't simply a matter of taking a vote in the project meeting. It's common for skeptics to vote for the inevitable, then complain about "that stupid idea", as soon as they are out of the meeting.

A friend related that exact story a few years back, second hand from another friend. At one of the brainstorm meetings, the boss asked how he can make the job easier. Joe had a suggestion that was controversial but effective, and the boss asked if anyone had a problem with that idea. No one complained and they voted it in. As soon as the meeting was over two of the guys told Joe, "Why did you tell him such a stupid idea? That costs us money!" Joe's obvious reply was, "Then you should have said so in the meeting, that's why we were there."

That's how meetings go. As described in the first chapter, there is always a small number of people who won't voice a suggestion among strangers or superiors. This is obviously not

how buy-in works. I wasn't there, but I wonder if the team leader should have been able to tell by the look on people's faces that this idea wasn't going to fly.

I've been in project meetings and there are a few things that help get better participation from everyone involved, based on the info above. First, the highest rank person can't micro-manage the project. Even though mentally she does, everyone needs to feel like they are driving their own portion. Second, the meeting leader needs to actively push dialog. People need to talk among each other to become part of the family. When someone wants to say nothing more than "I'm on schedule", it might be a good idea to prod for details on their particular problem. These details might make or break others in the meeting.

The Myth of Passion

One idea hinted at in the last chapter is the concept of employee commitment. The idea is that someone committed to his job will work harder, but someone who is not committed won't work as hard. This is a lot of baloney in my estimation, for the simple reason that most people in any organization would rather be somewhere else but still make wonderful employees.

I could probably dig up plenty of evidence to support both sides of that statement, but I'm more concerned about people's behavior than the truth in this case. Here's why: A few years back, a panel of business managers were on a TV Q&A show. Someone called in the question, "One of our employees wants to leave for a better job, what should I do?"

One of the panelists suggested that the caller just can the guy right away, before his toxic attitude spreads. This horrific advice is embodiment of the committed employee myth. A far

better course of action would be to find out why he wants to leave. Statistically, the most common reason would be to get away from a dud supervisor. So that guy's advice would likely cause the caller to can the good guy and keep the bad guy.

The real issue here is whether any employee who lacks commitment is worth keeping. And the answer is: An employee is hired or fired for their performance; their commitment is meaningless. It would be great if all our employees were committed and passionate about their jobs, but most are not and they do great jobs.

Here is a good example of that. A looong time ago, I got hired on a production line. When the stockroom guy went on vacation they moved me into the cable room where spools of multi-conductor cables were hanging everywhere. The place was a mess with spools out of order, unmarked stuff lying on the floor, and what have you. In two weeks I got all of the spools on the racks in numeric order, all the unmarked pieces identified and hung, the floor was swept and the file cabinet completely indexed, all the while delivering cables to the production line on schedule, according to the workorders that kept arriving.

When the production manager walked in, all he said was "Wow. What happened in here?" So I told him all the stuff in the last paragraph. This sound like a story of a passionate worker who is committed to the organization, but it was nothing like that. I wasn't passionate about anything, this is just how I work - I straighten and organize everything. My house is like that and the outline to this book is like that.

As for my commitment, a week later when the stockroom kid came back I was sent back to the production line, and then a week later I found a better job. If they had put me somewhere important I would have stayed, but if they're not committed to me, I'm not committed to them.

This same story plays out everywhere. It's called The Myth

of Passion. There is no connection between passion and performance. It's great when a person has both, but if I had to choose, I'd pick the person with performance every time. In organizations everywhere, some of the best workers are the quiet organizational wizards.

Passion its self isn't always a good thing. Who here hasn't worked with a passionate buffoon? These are the people gung ho on the job even when they don't know what they're doing. They'll move furniture and disassemble equipment before realizing the image in their head won't work. Then the boss calls in the quiet deep thinker to bail them out.

In reality, passion is a good thing, but it isn't something someone brings to the job. Constructive passion almost always comes from working on a high performance team, whether formal or informal. The team becomes a family and each person becomes an important part of a team that does important things. Staying late with the family becomes more interesting than going home.

Passion also works in reverse. A passionate individual walks in with big ideas and boundless energy, but is quickly doused with the cold slog of a rudderless boss and a dozen oarsmen all paddling in different directions. Nobody wants to hear the big ideas, and as long as the place makes its budget why stick one's neck out.

A close cousin of passion is competition. Businesses will frequently create a competition between units to stir a passionate effort. Henry Ford did this a hundred years ago by hanging the name of the leading production team on the plant entrance. This created an intense competition between teams, who would kill themselves to see their names presented to the world.

The competitive spirit is an interesting cookie. It's not always clear who the competitive ones are. It's not always the big talkers, all talk and no action, but is sometimes it’s the quiet

ones and you don't even know it.

That was the case a few years back, where the quietest person on my line was the highest performer. She was unbeatable, and it took me a while to realize that it was all a competition to her. She would watch everything going on around her out of the corner of her eye, to ensure that no one was out producing her.

As we neared the end of life on an injection mold process, we did a huge run of all available material. Everybody paired up and took turns running the machine every day until all of the material was gone. Each day the numbers went up and up, in a friendly competition. 300 units, 400, 500, and up. This young lady had the machine on the final day, and the next morning I saw on the log sheet that they had done 950 units.

So I asked her how she did, and got the heroic reply: "Nine hundred and fifty units - there's no way Joey can beat those numbers!"

What a laugh. It was all a big competition to her, and she won. I need to put that attitude in a bottle and hand it out to the rest of the team.

The Paradigm Change

One of the more innocent ways to disrupt a facility is through a paradigm change. The new guy comes in and wants to make some kind of impact that will justify all that big talk in the interview. This happens at all levels of an organization, from top to bottom, and is probably a good thing. In fact, this is a common reason for swapping in some fresh blood in the first place.

A paradigm change is a broad new plan that changes everything in the organization in subtle ways. For example, when

the shop floor is suddenly changed from functional alignment to customer alignment, all of the productivity charts on the wall are suddenly obsolete.

Sometimes these subtleties can be wacky. A long time ago, when I was a technician in the Air Force, we had a wooden cane at each of our work sites. These were for pulling an electrocution victim off of electrical wires. One day, the new TAC safety officer declared that the varnish on the canes needed to be removed. It was holding in moisture, allowing the canes to conduct electricity.

This was obvious nonsense, but we sanded them down anyhow. (They don't ask our opinions in the Air Force.) Then two months later we were told to put the varnish back on because it was actually preventing the canes from conducting electricity.

I'll admit this is a lame story, but I saw it repeated nearly everywhere. The new guy wants to make an impact and presents a flawed plan, before gathering his team who already knew it wouldn't work.

One of the most common changes is to realign the org chart from functional alignment to customer. This is usually a good idea, but it's not the golden egg that people seem to think it is. Rule One in any organization is simply: Execution. People don’t really care who the boss is, they just want a boss who lets them do their job.

Changing to a customer aligned chain of command can help the team respond to events quicker, but does it actually reduce cost or increase quality? Sometimes, but when quality and deliveries collapse, it might be a good idea to re-align to a functional layout. With this type of organization a manager might be in charge of all mold machines (for example) regardless of customer or industry segments. This allows him to become an expert on that particular technology and hopefully get his team running at optimum performance. This is frequently a

good idea and some places will just leave it this way.

Most places where I've worked it was common to toggle manufacturing engineering back and forth between manufacturing and engineering. Either side of the org chart has its pluses and minuses, but the bottom line is: it never made a difference. We were the same people doing the same job.

Appendix A References

These books are referenced in this book.

Note 1:
The Principles of Scientific Management
Frederick Taylor
Published: 1911
Republished dozens of times.

Note 2:
Kip Tindell, The Container Store
standfor.containerstore.com/putting-our-employees-first

Note 3:
The Mythical Man-Month:
Essays on Software Engineering
ISBN-13: 9780201835953 (pb)
Author: Frederick P. Brooks Jr.
Edition: 2
Publisher: Addison-Wesley Professional
Published: August 1995
Original:
ISBN-13: 9780201006506 (pb)
Published: January 1975

Note 4:
Territorial Games:
Understanding and Ending Turf Wars at Work
ISBN-13: 9780814474105 (pb)
Author: Annette Simmons
Publisher: AMACOM
Published: April 2006

Note 5:
Leadership Gold:
Lessons I've Learned from a Lifetime of Leading
Author: John C. Maxwell
ISBN-13: 9780785214113
Publisher: Thomas Nelson
Published: March 2008

Note 6:
What Google Learned from Its Quest to Build the Perfect Team
www.nytimes.com/2016/02/28/magazine/what-google-learned-from-its-quest-to-build-the-perfect-team.html
(this article is an excerpt from this book)
Smarter Faster Better:
The Secrets of Being Productive in Life and Business
ISBN-13: 9780812993394 (hc)
Author: Charles Duhigg
Publisher: Random House
Published: March 2016

Note 7:
New Shop Floor Management:
Empowering People for Continuous Improvement
ISBN-13: 9781451624243 (pb)
Author: Kiyoshi Suzaki
Publisher: Free Press

Published: October 2010

Note 8:
Toyota Kata:
Managing People for Improvement, Adaptiveness and Superior Results
ISBN-13: 9780071635233 (hc)
Author: Mike Rother
Publisher: McGraw-Hill Education
Published: August 2009

These are other interesting books that I've read in recent years:

Follow This Path:
How the World's Greatest Organizations Drive Growth by Unleashing Human Potential
ISBN-13: 9780446530507 (hc)
Authors: Curt Coffman; Gabriel Gonzalez-Molina
Publisher: Business Plus
Published: October 2002
ISBN-13: 9780446690355 (pb)
Publisher: WARNER BOOKS

What America Does Right:
Learning from Companies that Put People First
Authors: Robert H Waterman jr
ISBN-13: 9780393035971 (hc)
ISBN-13: 9780393342017 (pb)
Publisher: W. W. Norton & Company
Published: January 1994

The Medici Effect:
What Elephants and Epidemics Can Teach Us About Innovation
ISBN-13: 9781422102824 (pb)
Author: Frans Johansson
Publisher: Harvard Business School Press
Published: October 2006

Contagious Success:
Spreading High Performance Throughout Your Organization
ISBN-13: 9781591840602 (hc)
Author: Susan Lucia Annunzio
Publisher: Portfolio Hardcover
Published: November 2004

Serial Innovators:
How Individuals Create and Deliver Breakthrough Innovations in Mature Firms
ISBN-13: 9780804775977 (hc)
Authors: Abbie Griffin; Raymond Price; Bruce Vojak
Publisher: Stanford Business Books
Published: May 2012

Lean Production Simplified,
Third Edition: A Plain-Language Guide to the World's Most Powerful Production System
ISBN-13: 9781498708876 (pb)
Author: Pascal Dennis
Edition: 3
Publisher: Productivity Press
Published: September 2015

Appendix B
About the Author

Mike Henry is the programmer for Ghost Town Software.
He formed GTS in Colorado 20 years ago, where he developed and sold topographic map software for hikers and bikers. This cutting edge software combined multilayered topographic maps with popup photos and trail reports into a handy reference for hikers and bikers. Finding supplies in those early days was difficult, like living in a ghost town - hence the name.

The topo software was very popular with the users, but low sales forced him to get a real job, which landed him in Vancouver Washington as a manufacturing engineer. GTS is now mostly apps that were created for personal use but are posted to the web site for any others who might find them useful.

www.GhostTownSoftware.com