

Timothy Mahan

11/14/17

CIS 410-01

Connor Metal

Connor Metal came under some radical change over the last few years. The company moved from a more functional structure to a divisional structure, with

each division manager responsible for every aspect of their division. The company also became more aggressive in its strategy, which led to an increase in shipments from just 8 million in 1982 to over 17 million in 1988. The problem is, that increase in shipments is not translating to the bottom line. Looking at the financial summary of the company, there have been years in the past where profitability even surpassed that of 1988 and 1990. Two systems have been developed and implemented in house, one is Job Boss, the other Connor Software. Connor Software was developed by Michael Quarrey in the Los Angeles plant, which was struggling, and turned the plant around. Within a few months, runs speeds on a number of jobs increased by 20%. Defective jobs were reduced by 10%, and credits issued to customers dropped to .5% of all sales during that time. As time went on, sales rose 28% even though the head count dropped from 100 to 85 employees. The Los Angeles plant was eventually hitting sales of around 10 million annually, which the Job Boss system could not boast in its utility. It was a good system, the San Jose plant that used the Job Boss system was the leader in profitability until the Connor Software came along. At this point, the issue is not whether or not to implement Connor Software across all divisions, it is a matter of how to get each division to accept and use the software from top to bottom.

Analyzing Connor Metal with Porter's five forces would serve well, it will help determine how willing the divisions will likely be to accept radical change like the Connor Software would bring by analyzing the environment. Porter's five forces include intra industry competition, threat of substitutes, threat of new entrants, bargaining power of customers, and the bargaining power of suppliers. Looking at the environment, there are a lot of job shops to compete with, somewhere around 600 to 700. The market functions like that of a commodities market, where

customers generally opt for the cheapest, as quality is notoriously low in the job shops, and parts like springs seem to just be springs, it does not seem to really matter where they come from. The intra industry competition is high in this market, customers generally go for the cheapest, so if Connor is not producing cheap enough, Connor is going to lose business to someone else. This is a market traditionally for the more functional, rather than divisional organizations, which makes the decision to switch initially strange. The threat of substitutes low to moderate. There are a few alternatives customers might be able to use, like some sort of rubber band depending on how the products are put together, but springs seem to be more durable. The threat of new entrants is pretty high. The market seems to be saturated with manufacturers, but seeing as how it's a commodity market, pretty much whatever can be produced is going to be sold at some point. I would not be very difficult to get the funding to open a job shop to compete in the market. Customers have pretty high bargaining power in this market, with all of the intra industry competition, there are plenty of alternative places for customers to get pretty much the same good, with the same quality as any other. All customers seem to worry about in the beginning is where to get the commodity items like springs for the least cost. Bargaining power of suppliers is also decently high. With a plethora of job shops to supply, suppliers will not be concerned with losing a few customers here and there is likely plenty of business as-is. While not everything Connor did was commodity-like, really only about 20% of their business fit this description, the commodity-style in which Connor would be expected to operate holds, as even though the custom products required considerable engineering expertise, customers still typically chose their suppliers based on who was cheapest. This lends to Connor choosing the functional structure it had for ages, as functional

strategies are suited for environments where demand is stable, and the strategy is cost-focused. The break from the functional structure into a divisional structure is important, as will be reviewed later in the alternatives.

In this case there are four stakeholders worth considering. First are the employees, as their acceptance of the Connor Software system is crucial to its success. Without employee acceptance, the system will fail or violate company values and structure. Next are division managers, they are each charged with managing their own division in just about every regard. They determine the direction the division takes, almost like a CEO of the division. How these managers are affected should be considered separately from employees, as acceptance of the new system will be heavily relied upon in convincing other employees the system is good to use. Next are customers, as any radical change like implementing a new system like Connor System will either hurt customer relations or improve them, how the system affects customers is crucial in consideration for the obvious fact that without customers, Connor Metal will not make any money. Finally, Sloss will be considered; there is a lot in this for him, he is the CEO and it is a family business that was started over 40 years ago at the time of this case.

The first alternative, as always, is to do nothing. Let the Connor Software run at the Los Angeles center and let the division managers make the call as to what system they use. The people put in charge make the decision that best fits the needs for their plant. This sounds pretty lazy, but it is not the worst option to consider. Chuttur discusses the Theory of Reasoned Action model, and how behavior can be predicted based on a few elements of a person. By looking at intentions and prior beliefs a person would hold for a given behavior, one can determine whether or not a person will take the action and perform the actual behavior. Knowing how people are going

to respond is crucial to knowing whether or not the people a system is being developed for are actually going to use it. As we went over in class, the easiest way for employees to kill a system they do not like is to simply not use it. When developing and implementing a system, the idea is to get people to use it once it is finished. The way behavior can be determined is to look at beliefs. From beliefs attitudes are formed, generally, this is either a positive or negative attitude, from the attitude comes the intention, and from intention comes behavior. Take for instance, the manager of the Los Angeles plant, Spanos, wouldn't initiate a technology oriented approach to solving the plants problems, this was because he believed that information systems were too inflexible for the custom work Connor Metal did. Because he believed this, he had a negative attitude, because of this negative attitude, he had no intention of implementing a technology centered solution, so he simple did not. It was not until Quarrey came along that an IS system was created and implemented. In order to understand how this is relevant to doing nothing in this situation, we next need to look at the four components of a belief, at least from an IT standpoint.

In a study done in 2009 of South Korean university students, Park describes the structural model the study revolved around, these were self-efficacy, subjective norm, system accessibility, perceived usefulness, perceived ease of use, attitude, and behavioral intention. Of these, usefulness, ease of use, subjective norms, and system accessibility are components of a belief. In class, we called these performance expectancy, effort expectancy, social influences, and facilitating conditions respectfully. Performance expectancy is what a person can expect to get out of a system, it is where the person assumes what the technology will do for them, the higher the performance expectancy, the more likely the technology is to

be adopted. Effort expectancy is how difficult a person expects learning the new technology will be. The lower the effort expectancy, the easier a system will appear to learn, if it is easy to learn, it more likely to be adopted. Social influences, or subjective norms, rates how socially acceptable the technology is. To put it simply, the cooler technology makes someone look, the more likely they are to pick it up and use it. The iPad was the cool thing to have when they launched in 2010, having one elevated social status, and that gave people incentive to buy them. Finally, the facilitating conditions weigh in. If the support for a new piece of technology is already there, it makes the technology easy to adopt. If it's easy to adopt, it's more likely to happen.

The Connor software as described by users in the Los Angeles plant as easy to use, easy enough that users could ask each other for assistance rather than going to Quarrey for help. Even employees who had never before used computers were taking to the system with ease. This system a proven low effort expectancy for any new prospective users. Next the Connor Software proved its worth when it increased sales, even as head count was reduced, the plant still reached annual sales of \$10 million. The system also allowed for better communication between the shop employees and engineering, which resulted in higher quality products. Customers started buying Connor parts even though they were more expensive than the competition. Now that quality has reached such high levels, Connor can compete with a differentiation strategy in a cost focused environment. Being able to charge more per part means more revenue generated per unit. Provided the selling price outpaces the costs of production, this makes the company more profitable too. In terms of social influences, it comes down to whether making use of the system is seen in a positive light by employees collectively. In Los Angeles, though Spanos

was anti-technology, the employees had a say in how the system was designed so it met their needs. By contributing in the creation of the system, they had a stake in the use of the system. It only seems natural that people will support an initiative they had an impact in. Because of this, social influences will be different from plant to plant. The San Jose plant, for instance, will likely resist the Connor Software since they have Job Boss, and likely had a hand in its creation and implementation as well. The facilitating conditions are not entirely there at the other plants just yet, they'll need the terminals that Connor Software relies on, but that shouldn't be difficult to implement.

The how and the why of the do nothing alternative are pretty straightforward. Doing nothing is a good option because it will allow the plants to adopt Conner Software without coercion. Without coercion, the divisions will take ownership in the choice they make. The Portland plant is already chomping at the bit to get the Connor Software, let them contact the Los Angeles plant and work something out. They do not directly compete, so there should not be a problem there. If needed, facilitating the implementation by corporate should happen, but with this alternative, the corporate office would not be proactive in getting Connor Software rolled out unless contacted by a plant looking to improve. First Portland would adopt it, and in the interest of preserving the division by finally turning a profit, Dallas would likely seek it out next. It would be a waiting game for San Jose, eventually, provided the leader of the division is rational, the superior system would win and the San Jose division would adopt Connor Software. This alternative would take the longest, but it would produce the least friction between the plants and corporate.

The next alternative is one management would typically be more drawn to as it stems from the idea of benefits deferred are benefits lost. To the reap the most

benefit, whatever is beneficial should be implemented as soon as possible. By this logic, in order to reap the most benefit from the Connor Software, it should be implemented in all plants yesterday. This alternative, is to force the immediate rollout of Connor Software to all remaining plants, including San Jose, which already has Job Boss. This would ideally allow each plant that uses the system to reach the levels of performance seen in Los Angeles. The problem with this goes back in part to the Hackman and Oldham model. The model describes 5 core characteristics and 3 moderators for employee satisfaction, the relevant part of this model to analyze in this alternative is that of autonomy. Giving employees more autonomy makes them happy, taking it away has the inverse effect. Sloss has also pushed the company into a new structure which is centered around the autonomy of divisions to handle their affairs. Forcing the Connor Software onto the divisions will violate the divisional autonomy Sloss has spent the last few years building, and it is going to generate push-back from the San Jose division. San Jose already has Job Boss, it has served the division well, and is popular among the employees. For these reasons, forcing the Job Boss software onto the divisions is a bad idea. It undermines everything Sloss did previously, and is only going to cause issues in a plant that is hitting record sales. It makes no sense to go and mess with a good thing. If the San Jose plant later proves to be less profitable because it is using an inferior infrastructure, then the issue should be revisited. Adams highlights in chapter 15 that employees do not like change, as change often does not work in the favor of employees. In this instance, that is probably true. For divisions like Dallas and Portland, getting the Connor Software would probably be a great thing, the problem is, this alternative involves not only changing how San Jose functions with respect to its IS operations, it's forcing change that employee typically fear anyway on a group that is already

doing incredibly well. This is different from being stagnant, the Job Boss is simply a different direction from the Connor Software, and in San Jose it works. With Job Boss, San Jose can continue to refine the system it is familiar with, and the profitability of the San Jose plant isn't violated in order to implement a system that makes San Jose uniform with other divisions it has nothing to do with. Because this alternative would violate the direction Sloss is pushing the company, would reduce autonomy for no perceivable benefit, and would disrupt one of the most profitable centers of the business for no discernable reason, this alternative will not be the direction Connor takes, nor should it be.

The final alternative is to go ahead and roll the Connor Software out to Portland and Dallas, but leave San Jose alone. San Jose is all set, they have Job Boss, they are very profitable, and employees and management seem to be pretty happy with the system. Job Boss isn't an old and outdated piece of technology that is holding the division back, it is a key to its success. This also does not violate the more divisional structure Sloss has moved the company to, and it doesn't reduce the autonomy of the San Jose division in exchange for no benefit. This alternative does give Portland the system they have already shown intense interest in acquiring, and it would give Dallas a platform to work from in order to finally become a profitable center for the company. In order to maintain the divisional autonomy of Dallas, and illusion of choice can be given, it would not matter so much if Dallas used Job Boss or Connor Software, but it definitely needs something to help it get to a profitable state.

Connor Software will be the way Dallas goes however. Cash lays out the four stages of learning in adopting technology. The first is initialization. In this stage, data-processing technologies are introduced for automating administrative operations. Next is the contagion stage, in which, the learning curve for a specific piece of

technology moves significantly upward as increasing numbers of people adopt the technology. This is the stage where everyone jumps on the bandwagon and begins experimenting with the technology. Next is Control, in this stage, managers pull back and trade slack that was prevalent in the contagion stage for tight controls until revenue growth passes up the IT budget. The final phase is integration, where the technology becomes a standard in the company. The Connor Software is in the contagion stage, it's new, it's popular in the Los Angeles and Portland plants, and employees are taking to it very quickly. Because of the social forces around the Connor Software, it being the cool new thing to use, Dallas will likely go with it, especially since it proved to increase the throughput of the Los Angeles division. Connor Metal isn't seeing much in the way of increased profitability right now with Connor Software, profit looks about the same as it has in the past. The profit is going to show when the Connor Software moves to out to stage 4. Here it will be standardized, finalized, and in constant use by employees without constantly revising and editing the platform by Quarrey.

The final alternative listed in this analysis is the best option by far. It does not disrupt a very profitable division, while at the same time it gets systems out to two of the least profitable divisions in the company. This is a system that, like Job Boss, has proven to assist in increasing productivity, getting it to the two least profitable divisions should be the initial focus. This is accomplished by getting Portland setup as quickly as possible, and then giving Dallas the choice between implementing Job Boss or Connor Software. Dallas will surely roll out Connor software since it's proven to be a good tool and it's the hot new thing to have at the moment. Connor Software has high performance expectancy, low effort expectancy, high social influence, and the facilitating conditions can be achieved in a fairly short period of

time. These attributes will form a positive belief in Dallas as it did Portland, which will lead Dallas having a positive attitude about the system, which form an intent that the Dallas division will eventually act upon. The second alternative isn't ideal as it creates too much friction for not enough return, and doing nothing will take too long to get the desired results. With the final option, Dallas can still take its time deciding, but the sooner Portland is successful, the sooner Dallas will adopt the software on its own.

Work Cited

- 1) Team FME. *Porter's Five Forces*. N.p.: Free-management-e.com, 2013. Print.
- 2) Chuttur, Mohammad. "Overview of the Technology Acceptance Model: Origins, Developments and Future Directions." *Overview of the Technology Acceptance Model*, Sprouts, 27 Sept. 2009, aisel.aisnet.org/cgi/viewcontent.cgi?article=1289&context=sprouts_all.
- 3) Park, Sung Y. "An Analysis of the Technology Acceptance Model in Understanding University Students' Behavioral Intention to Use e-Learning.", http://www.ifets.info/journals/12_3/14.pdf
- 4) "Hackman and Oldham's Job Characteristics." *Free Management Books*. N.p., 02 Nov. 2017. Web.
- 5) Adams, Scott. "15." *The Dilbert Principle: A Cubicle's-Eye View of Bosses, Meetings,*
- 6) Morgan, Gareth. *Images of Organization*. N.p.: SAGE, 1997. Ch 6. Print.