2006

Tech Culture -- a real world practice of effectuation theories

Yi J. Cheng
Louisiana State University and Agricultural and Mechanical College

TECH CULTURE --- A REAL WORLD PRACTICE OF EFFECTUATION THEORIES

A Thesis
Submitted to the Graduate Faculty of the
Louisiana State University and
Agricultural and Mechanical College
in partial fulfillment of the
requirements for the degree of
Master of Science

in

The Department of
Information Systems & Decision Sciences

by
Yi J. Cheng
B.S., L.S.U., 2004
August 2006

Acknowledgements

I am grateful to Dr. Ye-sho Chen, my advisor, for his active supports and his intellectual engagement in my ideas. Without his support, this thesis will not be done. I would like to thank Mr. Terry Landry and Dr. Ed Watson for introducing me to the Information Systems & Decision Sciences Master program.

Table of Contents

Abstract

Tech Culture is a local startup company that has a very creative idea in the network security management area. The two founders have a very strong startup for their business but encounter some problems during the expansion of their business. This thesis is to analyze their business model using effectuation theories to make sure the business model fits the current economic development. Then, start using various business theories to correctly define and analyze their problems. At the end, some theory-driven suggestions are given to Tech Culture for its future development.

1. A Detail Description of Tech Culture

In this chapter, I am going to discuss in detail about how Tech Culture was born, the planning stages of the software, and its ups and downs in just a little bit more than a year. Describing it into detail is necessary because it will build a solid foundation for later chapters when we do a deep analysis using different entrepreneurship research theories with every action Tech Culture took in its life span that we are about to mention in this chapter. I will use the third person form throughout this whole thesis because I want to provide a objective to myself and everybody else so that we all can look at Tech Culture as outsiders.

1.1 All Started at the Christmas Eve Dinner Conversation

The birth of Tech Culture is really accidental. It happened on December 24, 2004. It was a Christmas Eve get together dinner. Jack went down to Houston, TX to meet up with his friend, Sia (his current partner). During the dinner, Sia talked about the issue of no software out there could do automatic configuration for network switches during the virus attacks. They had a conversation and that brought up the birth of Tech Culture.

Sia asked: “Do you think it is possible that you can write an application to help a university to configure their network switches?”

Jack thought about it for a little bit: “Programming is not a problem for me, but I am not comfortable about how switches work. After all, I have no chance to touch them yet. I know theoretically how they work, but as far as experience, I have zero of it.”

Sia replied right away: “That is not a problem at all. I know how it works from front to back. If you can handle the programming part, I can give you the step by step instructions. What you need to do is to convert my instructions into programming codes, and figure a good algorithm to make

the program most efficient. And if you get it done, I am sure the university would love to buy that from you. Here is a chance for some extra spending money."

Jack said: "Sure, if I can get the instructions from you, I can write the program, let's call it Network Quarantine Software. But do you think other people will want this application also?"

Sia thought for a moment, and said: "Good question, I never thought of that before. Let me call around and I will get back to you on that later."

The Christmas Eve dinner conversation ended there. But after 5 days, the New Year Eve, Jack got a phone call from Sia, he sounded very excited on the phone: "Hey, Jack, if you are not busy, can you come down to Houston right now? I talked to the Network Administrator from a university; he is very interested in it too. And I talked to my brother-in-law, he works in an IT department in CA, and he said if the production is functioning well enough, he will be able to convince his supervisor for buying the product also. I think we have a business opportunity here. Do you want to start a company for this software? Seems no one is working on it right now and we can be the first one to start."

Jack has always wanted to start a business and he felt this is the perfect opportunity for him. Jack was down at Sia's house 6 hours after the phone call. And they started planning for our business and initialized this Houston based local company named "Tech Culture" right before year 2005 started. Their entrepreneurship adventure started from there with their creature of Network Quarantine Software.

1.2 Background of Two Founders of Tech Culture Inc.

As you can realize from section 1.1, Tech Culture is founded by two people: Jack Cheng and Sia Yek.

Jack currently is a graduate student, getting his master degree in ISDS at LSU. Before he entered the ISDS master degree. Jack got his bachelor degree in Computer Science at LSU, and during his undergraduate study, Jack has worked as a computer programmer for the Department of Louisiana Center for Educational Technology. He was mostly in charge of the technical side of the project he was assigned to. He was taught to listen to feedbacks, and learn from them, then constantly plan for improving the website. He was in the project for two and a half years and it really helped him develop his project planning and programming skills.

Sia Yek, the co-founder of Tech Culture is a very experienced network administrator. He is one of the people who set up the whole university network system when internet first started. He has more than 11 years worth of network administrator experience and he is the core Network person for his university, and he is asked for help from universities and small to mid-size companies all over Mississippi. He is a pretty well known person in a few universities in Mississippi. Throughout his career, he has developed a very well-rounded social network.

1.3 Choosing the Technological Tools

The first thing the two founders did was to pick what technologies they should use to write the software. By technology, I mean the programming language, the server platform and the database choices.

Sitting around and facing the wall all day is not a good solution. The two founders decided to go to chat with the network administrators about their experience. And results are shown below:

1. Some network administrators love the Windows-based servers because of the ease to use features and some love the UNIX-based servers because of the stability and the flexibility.
2. Each network administrator has different expectation. They each have something they want and they don't want. They want the software to be as flexible as possible.

3. They want the maintenance minimum cost of the application for both the effort aspect and the financial aspect.

With the result, Tech Culture has the idea that the software should support multi-server platforms. It should run on at least both Linux/Unix base servers and Microsoft Windows Servers. It should be customizable to fit individual needs while not affecting what the application should do. The software should try not to depend on any commercial software to minimize the maintenance cost.

Based on those, Perl became the first pick for the application programming language for the following reasons: First of all, Perl is a platform independent programming language. It can be run on both Linux/Unix-based and Microsoft Windows-based servers without any changes. Second, despite the abnormal syntax structure, Perl is famous of its easy to use regular expression feature. Using Perl will speed up the development of the application because the software requires a lot of validation using regular expressions.

PostGreSQL was the first pick for the database system. MySQL is the most famous open source object oriented database. It is very easy to use and has several user friendly interfaces available on the internet to fit different users' needs. A very good support is provided in the http://www.postgresql.org website. And since it is open source, it cuts down the software cost.

1.4 Database and Implementation Design

1.4.1 Brief Explanation of the Application

Before explaining the implementation design, a little bit explanation of the application is necessary. Computer hackers are very smart these days. It doesn't matter what kind of firewall the networks have or how good of the security systems the networks are, hackers can always find a way to hack in the network and do whatever they want. Sometimes, hackers do it just for self-fulfillment,

but sometimes they do it to damage networks. This is why network attacks happen so often in all different sizes of networks everyday. It is one of the biggest problems that all network administrators face. Network attacks can easily bring networks down. If a network is down, business will not function and cost money loss. If a network is down, getting it back up in the timely manner is very important, but what is more important is to how to keep the network up after bringing it back up. It is a huge challenge for every network administrator. Network Quarantine software is a software that automatically accepts signals from intrusion detection software (i.e. Snort) about any abnormal behavior the network is experiencing, takes appropriate actions in real time according to the configurations of the network administrators, and finally the software will notify the network administrator about the network abnormal behavior and the action it takes. This software, since it detects the abnormal behavior and acts on it in real time, can effectively quarantine the affected part of the network to keep the rest of the network up and functioning. It provides a solution for the network attack challenge the network administrators are facing.

The software is separated into two parts, and they are only connected through the database system (this means the database acts as a medium in this software. One part of it, I call it the "back-end", is the part users will never touch, it runs as a service (the Windows's turn) or a daemon (the Linux's turn) which means it will automatically start running when servers start or reboot. When something suspicious happens in the network, it will trigger the back-end part of the software, and the software will do its job following the configurations, store the result in the database, and notify the network administrator.

Another part, I call it the "front-end", is the user interface part. It requires a security login in order to get to it. It is the place for setting the configurations to each individual network administrator's needs. All the configurations will be stored in the database, and the "back-end" part

of the software will follow them to do its job. Another functionality of the "front-end" is for network administrators to see the result after the "back-end" finishes its job. There might be additional actions from network administrators for follow up. For example, if a computer is affected by an attack, the computer will be quarantined from the network. In order to bring the computer back to the network, some clean up action is required. The interface is highly customizable to fit each network administrator's needs.

1.4.2 Database Design

Since the software is separated into two main parts and it is only connected through the database, database design is very critical in this software. It has to be done in the way that can be kept 100% consistent. Plus, this is networking software, if the database is not consistent, it can cost all kinds of network problems. The database designed of the software is fully proved to be in the 3rd Normal Form (It means tables in the database are constructed in a certain way to make it consistent). This insures the consistency. And the next step is the code design.

1.4.3 Code Design

The software implementation is designed with future portable upgrades in mind. Since Perl is a procedure language, it doesn't support OOP method. In order to make the code reusable, the code will need to be written in procedures (functions), and try to break the dependency degree among procedures (it means to try to make each procedure as independent as possible). Perl is a scripting language, in order to protect the source code from being stolen, the source code is converted into Byte Code, which means the source code is compiled into the machine code.

1.4.4 Difficulties in Implementing the Project

Planning took a long time for this software (about 1 month). After lots of diagrams and formal documentations were written, the implementation fun started. The Network Quarantine Software

was the first project for Tech Culture. It was also the first Perl project for the two founders. The syntax of Perl could be very tricky in times. The major difficulties during the implementation were to get used to the "Perl regular expression" and passing array parameters to sub-procedures. A lot of searches were done on how Perl works in a formal way. One perfect example was: if there is a Perl file only contains procedures, the file has to be ended with "1;" by itself on the last line of the program. Or by default, the return value of the file will be "0", which means "false". When it is time to include that file in another file, it will generate an error saying "include not successful". The reason for that was the default return value for including "sub-procedure-only" file is "0" and once the program gets this "false" value, the program would think the file being included is not there, and this causes the error to show up. Only Perl would have this abnormal syntax, and it was a one of the biggest challenge to Tech Culture during the implementation phase.

1.5 Test the Software

Tech Culture can write the prettiest code in the world, but without testing the code, no one will know how the software will work. A manageable switch/router costs tens of thousands dollars. There are at least 3 major brands most of the networks administrators are using for their networks: Alcatel, North Bay and Cisco. Tech Culture will not be able to afford to buy one switch/router from each brand, not to mention different models work slightly different. So, how did Tech Culture solve this problem?

As we mentioned in section 1.1, before Tech Culture was created, there were already people who were interested in the software. People who were interested in this software certainly had troubles in their networks. There was one company that was having a big problem with their network. The network was brought down almost 4 times a week, even the president of the company started giving pressure the network administrator to fix the problem. If the problem could not be

fixed in a timely manner, the network administrator's job was in the cold ice. What Tech Culture did to solve their financial problem was to make an agreement with the company that Tech Culture would give them a huge discount if they let Tech Culture use their network for testing, along with the agreement, Tech Culture would pay for all the damage and a penalty fee if the network went down by Tech Culture's mistake during the testing period. The company took Tech Culture's offer. The next day, the two founders went down to that company to configure the switches and routers correctly, and at the same time, constructed a small network just for initial testing purpose.

After all the network setup was completed, the two founders used two ready-to-be-surplus computers in that company for testing servers. One installed Microsoft Windows Advanced Server 2003 and another one installed Fedora Core 3 Linux. The two founders also installed remote desktop connection server on the Windows server and VNC on the Linux machine. With this setup, Tech Culture could manage the testing network virtually everywhere.

1.6 Network Quarantine Software -- Version 1

The two founders used the test network as the initial testing stage during the implementation. When one part was finished, they tested that part on the testing network to make sure that that part worked correctly, then they notified the network administrator in that company, telling him that they were going to push the working part out to the whole network. The network administrator could start using that part to manage his company network.

After the implementation was completed, they went down to the company again. They checked where the network administrator put the parts of their software to and gave suggestions where the software should be located for better performance. The two founders called that location the "bottleneck" of the network. They then spent two days with the network administrator in that

company server room to closely monitor how the software interact with the network. The software was efficiently quarantined all the affected computers in the company network, and the network was up. Best of all, the network's traffic was very smooth and no one complained about the network being slow at all. With the acceptance of the company network administrator, Tech Culture decided to announce the first version Network Quarantine software was completed. Tech Culture started facing the next challenge: finding customers.

1.7 The Expansion of Tech Culture

Tech Culture's first sale business is, for course, from the test center company. In this sale, Tech Culture gave them a 70% discount. Yes, Tech Culture didn't make profit on this first sale at all. Instead of making profit on this sale, they have another agreement with the company: If the company continues to be Tech Culture's test center for their Network Quarantine Software development, Tech Culture will give them free upgrades plus a rebate of their purchase of the first version of the software. Instead of making profit out of this sale, Tech Culture converted this company to their business partner. With this agreement, Tech Culture didn't need to spend close to hundreds of thousands of dollars on switches and routers; they can use the company's inventories. On the other hand, the company would have network experts to consult if their network has any problem. Tech Culture made this agreement into a win-win situation.

The second sale happened just a week after the first sale. It was a company from Mississippi. It closely monitored the software development while Tech Culture was developing the software. The two founders were told that this company constantly asked for the information about the software but they were too scared to become the test center for Tech Culture for the first hand information. After observing the result of the company from the first sale, they decided to give the software a try.

Since Network Quarantine Software was a new concept, the founders decided to go down to the company to demo how the software would work. After they got there, an issue popped up: the network was not configured correctly at all. Most switches were in its default settings with the default session ID with no password. This was one of the major reasons why their network was always having trouble. It took the founders almost a day to configure all the switches correctly, and put the software in the "bottleneck" of the network.

From the second sale, Tech Culture's two founders got the idea of giving some follow up support to future customers. So, Tech Culture announced they would give one year of free support for all the customers. It was a very brave move and Tech Culture hoped to use this free support feature to attract more customers.

This strategy worked very well for Tech Culture. By the end of June 2005, Tech Culture already had 31 customers, and they certainly kept the two founders crazily busy. Tech Culture would love to keep the growth this way, but two people would not be able to handle this rapid growth rate, so they decided to expand the company by inviting one more person into the company; at the same time, they had an idea of expanding the Network Quarantine Software development to develop a package of software with the Network Quarantine Software in the center of the package. So, they started paying attention to their customers' complaints.

Lots of customers have complained that there was no help desk software out in the market that could connect with the Network Quarantine software. Consider this situation: If a computer is quarantined, the internet feature for that computer is disabled. The user of course will not know why. So they will call the help desk department to seek help, help desk assistants will use their help desk software to try to find out what the problem is, but nothing will work because the computer is quarantined. Help desk assistants can try everything, even physically going down to the place where

the computer is for the problem, nothing will work with the port being enabled. And all those efforts take time. The help desk assistants will then pass the problem to their manager, and the manager will contact the network administrator, and then the network administrator will answer the manager with the problem. The software helps the network administrator to manage the network efficiently, but it gives headaches to the help desk department.

Within a month, in July 2005, Tech Culture gave response to their customers. Tech Culture would push out its own version of help desk software which would interact with the Network Quarantine Software. This would become the second big project for Tech Culture. The project was planned to finish in October 2005, and it would only cost a fraction of price of their competitors for the Network Quarantine Software customers to get the help desk software. Best of all, there is no restriction for licensing. Once the customers purchase the help desk software, they can have an unlimited number of users to access the software.

This new project is the beginning of the expansion of Tech Culture. The third Tech Culture person was found at that time. Everyone in Tech Culture were very satisfied with this growth and Tech Culture's customers were totally happy with the quality of the software and the follow up support Tech Culture provided. Most of them showed their interest for purchasing the help desk software. No one in the world would know in just three months, Tech Culture would fall from the top of the mountain to the lowest ditch.

1.8 Flipping Down Hill

Tech Culture treated planning as the most critical part of the whole software development cycle. It took all 3 people in Tech Culture 3 whole days to interview all their customers. After that, they had a brain storm meeting to pick out good functionalities from the competitors, then adapted and enhanced them in their help desk software; also, from the customer interviews, they had a few new

ideas for help desk software that others didn't have. After all the features of the software have been decided, the three founders of Tech Culture started drawing the diagrams and writing documentations. It took them a total of more than 3 weeks to finish planning. At the end of August, they started getting into the implementation.

Because the three founders of Tech Culture lived in separate places, once they were assigned to their parts of work. They went back to their homes and each of them expected other people to work on their own parts until the time to meet up again.

The first two founders, Jack and Sia totally trusted the third person because he was Sia's ex-coworker. Sia knew for the fact that the third person has a very solid background of computer theories and programming skills. Plus, since Jack and Sia had a very great partnership for the first half year of the Tech Culture adventure, they naturally thought that this partnership would expand to the third person.

In early October, Jack and Sia have done their parts. Jack has finished his part of the program and Sia has contacted all the interested customers, and made promises to those customers that Tech Culture will deliver the first version of their help desk software to them in late October. And it was the time for three founders to meet again to combine everything together and test the software. The meeting was held in Natchez, Mississippi, but only Jack and Sia showed up. The third person didn't show up. The third person sent an email to Sia telling him how sorry he was to miss the meeting. And he said to give him another week to work on his part before meeting again.

Sia and Jack went down to Natchez, Mississippi again the next week, hoping he would be ready. This time, he showed up in the meeting, but with only a few pieces of code. He apologized and said that he would not be able to continue working in Tech Culture because he had gotten a great offer from a major corporation and he wanted to advance his career with that company.

This was the turning point for Tech Culture because the deadline was approaching in less than three weeks, but the software is only 70% done, on top of that, the software has not been tested yet!

Jack and Sia were very angry of the irresponsibility of the third person. They let him out of Tech Culture, and they gave him back the code he has written for the project because Jack and Sia both understood that if they accept his code, they might have some legal issue with the third person later on.

Jack started working on the unfinished parts of the software while Sia started calling all the customers who were interested again to try to have the release date delay to Mid-November 2005. The customers were certainly unhappy because they were promised to have the product by late October. Business is business. When they promise something, they'd better keep their words. Breaking promises in business is one of the worst things that would happen to companies. Tech Culture failed to keep their promise to their customers. Though this was the first time, for a start up new company, this first time was enough to kill the trust they built in the past half a year. After the Network Quarantine Software, customers started trusting Tech Culture, but the foundation of trust was not solid at all. It could be broken very easily, and it was broken in this help desk software delay.

The help desk software was done in November. Tech Culture did sell some copies to their Network Quarantine Software customers. To show apologies to their customers, they gave each of them a 50% off discount.

At the same time, some of their customers' networks had some problems. Though they were not related to the Network Quarantine Software, the customers still called Tech Culture for support. The two founders knew it was not their responsibilities to fix those problems, but in order to gain

the customers' trust again they had to fix those problems for them. Worst of all, they had to do it for free because they don't want to make the customer relationship any worse.

Just after finishing solving the network problems, a major corporation in the networking field announced that they would have a new model of switch out. The new model of switch would have network quarantine features embedded in the hardware. Worst out of the worst, they claimed it will support multiple brands.

If this new model switches really got out to the market, it would easily drive Tech Culture out of business. And there was nothing Tech Culture could do about preventing this from happening. The two founders knew that they could not compete with big corporations, not even a chance.

1.9 Questions for Tech Culture's Future

With all the problems Tech Culture is facing, there are 5 major questions the two founders should themselves:

1. Should Tech Culture continue? If yes, what do they need to do to bring Tech Culture back?
2. How can they keep their existing customers?
3. Should Tech Culture continue on the Network Quarantine Software project?
4. Should they start looking for new projects? If yes, what project should they start?
5. What should Tech Culture improve to make their business model more efficient?

The purpose for this thesis is to find answers for these 5 questions by applying business theories which will show in later chapters.

2. Effectuation --- The Business Theory behind Tech Culture

With the explanation from the previous chapter, we now have a clear understanding of what kind of company Tech Culture is and what has happened to it during its life span. At the end of last chapter, we left several questions that need to be answered. Before we can answer those questions, we have to analyze Tech Culture in depth to understand why Tech Culture is acting the way it has been acting. In order to shape Tech Culture's future, we have to understand why it is doing what it is doing right now.

The two founders of Tech Culture are definitely entrepreneurs. And the most fundamental business theoretical approaches for entrepreneurship research are causation and effectuation. In this chapter, I am going to briefly explain each of the two separately, and then I will apply them to Tech Culture to see which one is really driving Tech Culture forward. Since Tech Culture is having problems now, I want to do a deeper analysis to see if the theory that is driving Tech Culture forward fits Tech Culture. At the end, I want to analyze the theory to see if it fits today's macroeconomic development.

2.1 Causation

Traditionally, entrepreneurs will fix themselves a set of goal when they first start their business and they will choose their own ways to reach their ultimate goal. The ways they choose will closely related to their personal experience. Let's have an example here: John is a local entrepreneur who wants to start a fast food restaurant and eventually expands his business using franchise model. He starts doing research on other fast food restaurants to learn how they do business their ways, such as how much a cup of orange juice should cost, what the right size for the restaurant is, how many

tables the restaurant should have inside; what kind of fast food most of people like, how much time the food should take to be ready, etc. After getting all the data, John will analysis those data to see what fits him the most, and use those as a guideline to start his restaurant. He starts building his own: he builds a kitchen that most fast food restaurant has, builds a drive way for drive through service, and buys tables and desks to put inside the fast food restaurant. After spending $120,000, John finally got his first fast food restaurant ready.

In the above example, John has a fixed goal since day one he starts setting up his business --- He wants a fast food restaurant and his goal is to open a fast food restaurant just like other fast food restaurants and join the fast food industrial competition. The fast food industry has given John a very good model to follow. He takes the model as given and analyzes the model to make it fit for his restaurant. Every thing is planned for only opening a fast food restaurant, and in the future, he wants to franchise his restaurant to other franchisees to expend his fast food restaurant business. He has his goal and he is trying his best to achieve that goal and doing what he thinks is the best of his business.

This is called the Causation Process. By definition, causation processes take a particular effect as given and focus on selecting between means to create that effect.

This doesn't fit in the model of Tech Culture. As we showed last chapter, the birth of Tech Culture was so incidental. The two founders got together for a dinner. They discussed a problem at the dinner table, saw an opportunity, got really excited and decided to start their own business. Their short term goal is to write software to solve a network security problem. They looked at what they could do, and knew that they had the ability to write the Network Quarantine Software, so they decided to do it. That is all. No fixed goal at all. The two founders of Tech Culture focused on what they were able to do, not what they can achieve. This process is called the effectuation process.

2.2 Effectuation

Effectuation is a popular topic for entrepreneurship research in recent years. The most sounded people who were involved in this field is the 1978 Economic Nobel Price winner Dr. Herb Simon and his student Dr. Saras D. Sarasvathy.

By definition, effectuation process is the process that takes a set of means as given and focus on selecting between possible effects that can be created with that set of means.

Let's use John, our local entrepreneur, as an example again. He wants to start a business, but he only has $2,000. Using causation process, he is convinced that he doesn't have enough money unless he goes to the bank and get a loan. Instead of going to get a loan, he spends his $2,000 to rent a counter in the food court in a mall. It is a small counter that only one person can stand inside the counter. He has no kitchen which means he can not cook, but there is a fridge that comes with the counter. He has an idea of making sandwiches which doesn't require cooking. Since the counter is very small, he can't make the sandwiches in the back, so, he decides to make it in front of the customers. After the first month, he finds out that his business is not giving him a return that he is happy with. Also, he notices that most people go by his counter ask for smoothies, which he doesn't have. After finding out he doesn't have smoothie, the majority of the customers will walk away, only a few will stay to get a sandwich with some drinks. With this observation, he decides to put his sandwiches aside and starts on a smoothie adventure. This change brings him a big surprise. He has a great return at the end of the second month. He then partners with his neighbor to give a combo deal option to shoppers---John makes the smoothie while his neighbor makes the food which brings him more return. After two month of making smoothies, in order to provide better services to the mall shoppers, he decides to create smoothies with a healthy nutritional value in it. He succeeds in

the research by finding a new formula for making smoothies. He no longer wants to be stuck in the small counter. He decides to start his own consulting business for the nutritional smoothie industry.

Who can predict John will end up in the nutritional smoothie industry with his $2,000? If John follows causation process, it is impossible for him to even start a business, but effectuation process leads him to a whole different direction.

According to Dr. Sarasvathy, the distinguishing characteristic between causation and effectuation is in the set of choices: choosing between means to create a particular effect (causation process), versus choosing between many possible effects using a particular set of means.

Figure 1 by Dr. Saras D. Sarasvahy in Appendix A can clearly explain how effectuation processes lead entrepreneurs' adventure. Effectuation processes start with generally three different kinds of means: who they are, what they know and whom they know. With all these three points being identified correctly, Entrepreneurs will brainstorm themselves to find out what they can do based on those three means. With the temporary goal, they will take it to their social network community to see how their social network reacts with their idea. Then later on, stakeholders will come up to negotiate with the entrepreneurs and make commitments with them. And during the negotiation process, both the entrepreneurs and the stakeholders start redefining their temporary goals, and start finding new means and new goals. Finding new means will bring the entrepreneurs to the beginning of a new cycle. They need to redefine who they are, what they know and whom they know, and what they can do that they could not do earlier with the new goal and then go to the expanded social network to explore their new goal. By finding new goals, it will bring them to the stage of "what can I do?". And then explore this new goal to the community. With this diagram, we can see effectuation processes is very self-explanatory. The force that is driving entrepreneurs forward is the effects (means). Effects can happen everywhere at anytime, which means goals will

be changing every time an effect is caught by the entrepreneurs. Without a fixing goal, no one will be able to predict where the entrepreneurs' future adventure will be.

2.3 Use Effectuation Process on Tech Culture

I have explained how effectuation processes work on general entrepreneurs in the last section. In this section, I am going to apply the effectuation theories to Tech Culture.

2.3.1 "Who Am I; What I Know; Whom I Know"

Again, let's go back to the birth of Tech Culture section. There was a short conversation at the dinner table between Jack and Sia. Sia asked Jack if he was confident on his programming skill. Jack's reply was yes, he could do the programming part, but he was not confident on his knowledge on network structures. Sia quickly responded that he was the network administrator; he could handle the network structure algorithm and configuration parts. This is a reflection of "Who am I". Jack identified himself as a computer programmer, and Sia identified himself as network administrator. They both made their identity very clear. Then the conversation went on. Sia started giving Jack confidence that as long as he can concentrate on the programming part, Jack could leave others to Sia because Sia knew what he could do. This is "What I know". The conversation ended at the point that Sia told Jack he was going to follow up by giving his friends some calls to see what they would say. This is the expression of "Whom I know?" Isn't Tech Culture's birth and the first stage of the effectuation process a perfect match?

2.3.2 "What Can I Do" and "Call People I Know"

After the conversation, Jack and Sia found out that they could write the Network Quarantine Software. As promised, Sia called his friends to see what their friends' inputs were. It turned out that Sia's friends were very excited, and then Jack and Sia decided to start the business together. Please note that until then, they didn't set any goals. They found something they were capable of

doing, and treated this as an opportunity to start a business. They never thought where this idea would lead them to.

2.3.3 Stakeholders Commitments

Tech Culture's stakeholders could be the first company that provided them a test center or customers who were happy with their Network Quarantine Software. They wanted to keep using Tech Culture's product. That was why they started complaining the help desk issue to Tech Culture. They were committed to Tech Culture's product, and they were trying to tell Tech Culture to find a solution to make Network Quarantine Software communicate with their help desk software.

2.3.4 New Goals

With the negotiations with their customers, Tech Culture started thinking "What can I do?" again. At the end, they found that they could join the competition of the help desk software market. They redefined their goal from the Network Quarantine Software to a suite of software that will interact with Network Quarantine Software. Help desk software was just the first step of it.

2.3.5 Conclusion

The three short sections above have linked Tech Culture's actions to the effectuation theory. I don't expect my readers to be convinced that Tech Culture is a real world practice of effectuation process because there is more evidence to come in the later sections of this chapter. In the next section, I am going to explain the "ABCD" of effectuation process and see how Tech Culture's actions match its "ABCD".

2.4 ABCD in Effectuation

In Dr. Saras D. Sarasvathy's paper "Causation and Effectuation: Toward a Theoritical Shift from Economic Inevitability to Entrepreneurial Contingency", she gave a comparison and came out 4 principles to differentiate between effectuation and Causation.

The first one is “Affordable loss rather than expected returns”. In causation, the focus is the goal (expected returns). Causation entrepreneurs will go for their goal by picking the effects that are the most efficient for them. They look at returns rather than the cost. In effectuation, entrepreneurs don’t have a fixed goal. They are more focus on effect. And the effect has determined how much the loss is going to cost. Let’s see how Tech Culture’s model fits into this principal.

How much does it cost the two founders to write the software? Their time and their efforts, that is all. They picked to use open source technology as we mentioned in the previous chapter. It means it doesn’t cost them a dime but their time and efforts to find out how those open source technology can come together. And their goal at that stage was to create software to solve the existing network problems. If Tech Culture failed, the two founders would lose all their time and efforts they spent on the software. They know they can afford to lose.

Second, one is “Strategic alliances rather than competitive analyses”. Remember we gave an example about John, our local entrepreneur opening a restaurant using the causation processes? He studied his competitors --- the local fast food restaurants and analyzes the restaurants’ data to pick what he would have in his restaurant. In effectuation process example of John, when he found out that selling smoothies was better than selling sandwiches, he partnered up with his neighbor to create a combo meal option for the shoppers. He turned his neighbor from his competitor to his partner to create a win-win situation for both of them. In Effectuation, entrepreneurs don’t analyze their competitors. They follow the effects along and find business partners to join to reduce uncertainty and erect entry barriers for the field they pick. In Tech Culture’s real world practice, they didn’t really have a competitor yet; they were the innovator in their field. Even though the start up cost was low, only took their time and efforts, but testing the software could be costly. The two founders didn’t have the money at that time but they partnered up with a company that had lots of

troubles with its network. By partnering up together, Tech Culture cut their cost they needed to spend on testing, and the company could start using the software while it was in development. Their network could be partially managed by the software and reduced the trouble that they had before the software was completed. This created a win-win situation by the partnership.

Third principal is "Exploitation of contingencies rather than exploitation of preexisting knowledge". Causation is good when entrepreneurs have their specialties, but this, at certain degree, limits the imagination of the entrepreneurs because they would not think out of the box anymore. Effectuation, on the other hand, is better for exploiting contingencies. When an unexpected problem shows up, effectuation entrepreneurs will start thinking a solution for it. With no preexisting knowledge present, they can come up with a very creative solution to solve the problem. Over time, the future of the entrepreneurs" business will be reshaped. In the Tech Culture practice, after the development of the Network Quarantine Software, they started hearing some complaints from their customers' help desk software. That was totally unexpected when they were developing the Network Quarantine Software. How could anybody possibly think that a network security software somehow affect the efficiency of help desk process? But it did happen. And in order to solve this expected problem, Tech Culture found a new goal---- to create their own version of help desk software to solve this problem.

At this point, I will need to go off the topic a little to explain something out of these 4 principals for a moment. I am doing this because it is the best place I can explain this point. The point is what causation and effectuation can happen spontaneously. We have been talking about causation and effectuation separately to explain their differences, but it is very important to know that the two processes can go on at the same time for a business. And the reason I point this out is

Tech Culture's way of solution, the help desk software problem, is the prefect example to show this point.

The effectuation process brought Tech Culture to realize that their Network Quarantine Software created an unexpected problem for help desk process, and they found a new goal --- to write their own version of help desk software. Then what Tech Culture did was to analyze other help desk software out in the market, to see what their goods and bads are, and then Tech Culture would selectively adapt the goods, eliminate the bads and add new features on the software. This is causation process because the second principal stated, competitive analysis is one key principle for causation.

Getting back to the 4 principals' discussion, the last principal is "Controlling an unpredictable future rather than predicting an uncertain one". Entrepreneurs using causation process will predict what risks they will have and try to prevent them from happening. Entrepreneurs using effectuation process, on the other hand, don't predict risks. Their logic is to the extent that they can control the future, but they don't need to predict the future. In the Tech Culture practice, this is exactly what they need to do now. Their business was going down hill because they broke the promise of shipping the help desk software on time. The reason that caused this was the problem of the internal management. This incident cost a big loss to Tech Culture. However, Tech Culture is still here. They are currently facing a huge unexpected challenge---- a big corporation announced that they were going to join Tech Culture's field and be their competitor. This could be predicted easily, but Tech Culture simply ignored the prediction. Now the problem came, it could be fatal to Tech Culture. The question becomes: how can Tech Culture control this problem in order to survive through it. I will talk more about this at the end of this thesis.

Dr. Ye-sho Chen, a professor who emphasizes his research on entrepreneurship development at Louisiana State University, summarized these 4 principles into "ABCD": Affordable costs, business partnership, contingencies leverage and decrease risk.

I want to propose that there are causal reasons among Dr. Chen's "ABCD":

1. Because of business partnership, the affordable loss will be lower. If Tech Culture didn't partner up with a company for testing their software, they would have to buy or rent the routers and switches and if they failed, they would lose the money that they spent on the routers and switches on top of their time and efforts they spent. For example, if they could not find a way to solve the problem of the big corporation joining the competition, they would be eaten up very quickly. They will have to eat up the costs of the switches and routers, which lead to more loss.
2. With affordable loss being lower, it gave entrepreneurs flexibilities to control their future. With the partnership, Tech Culture didn't invest too much into the network security field except their time and efforts. If, the worst case, they could not find a solution to prevent being eaten up, they can easily switch to another field and find another new goal to continue their entrepreneurship adventure. Lowering the affordable cost can give flexibilities to entrepreneurs to jump from field to field freely. This decreases risks that entrepreneurs would be trapped in one field because they invest too much and eventually were being eaten by major corporations.
3. Because of business partnership, contingencies leverage will happen more often. Because entrepreneurs partner up with other companies, business interaction between the entrepreneurs and their partnered companies is enforced. Unexpected issues could be brought up throughout interaction and increase the frequency of creasing new means. With

more new means, entrepreneurs can have more options to identify themselves, more means mean more effects. More effects mean entrepreneurs can have more directions to choose, more new goals to choose and more accomplishment they can complete. The effectuation cycle (See Appendix A) will happen more often. The more interaction, the more contingencies leverage, the more entrepreneurs will learn. The more entrepreneurs learn, the more completed knowledge they will absorb. What does "more completed knowledge" mean? To understand this phase, I have to get into Dr. Herb Simon's research about "Locality".

According to Dr. Herb Simon, people's knowledge, not just entrepreneurs', are highly bias. Most people obtain their knowledge through the social interaction with the community they are in. The knowledge is local. It works perfectly in the local community they are involved. So, what about moving their knowledge to another community? It might not work at all. In order to make their knowledge work in this new community, they have to learn what this new community can offer, and modify the knowledge they have in order make it work for this new community. To get a complete picture of certain knowledge requires practice in many different communities. Because the world is moving forward, the knowledge will change according to the world's movement. This is a never-completed practice. This is what I meant by "more completed knowledge".

Let's give an example. For example, Joey was a recent computer science graduate. During his college study, his department highly promoted Linux-based operating system. He used Linux-based operating system through his college study because the department provided full support for any questions students had. He became an expert of Linux OS and he has not touched Windows since he got into college because all he needed to do for his college courses could be done in Linux. The last version of MS Windows he used is Windows 98. Because of his high academic

performance, he was hired by one of the biggest corporations in the world. The first day of work, he was required to use Microsoft Windows to program. He started complaining how Microsoft Windows has changed and all the changes only led to make things more complicated. His co-worker Mike heard what he said, and said he has been using Windows product since he first touched computers, and Windows has advanced a lot only to make thing simpler.

Joey and Mike had different opinions about Windows because they were from different communities. Joey was from a community that uses Linux to do everything, while Mike, just like the majority of the population, in a community that used Windows as their everyday computer use. Joey didn't like MS Windows products not because Windows products were not good. It was only because he was not used to the Windows way to program.

After working for 2 months, Joey started agreeing with Mike because he discovered that Windows products did make things simple, even simpler than Linux. But he also realized that Windows was not as stable as Linux base OS. At this point, he gained a more completed knowledge about Windows and Linux. He adjusted his knowledge to fit in his work environment. He learned that Windows was good for desktop because of those entire easy-to-use features and Linux base OS was good for server side uses because of its stability.

Entrepreneurs need knowledge that is as completed as possible in order make a correct decision. And the only way to complete the knowledge is through interaction with different business partners in different communities. The more they interact with different kinds of business, the better decision they will make and the better control they will have for their future.

2.5 Effectuation Fits the Current Macroeconomic Development

Lots of researches have been done to show that Effectuation is better for start up entrepreneurs than causation for various reasons because of the "ABCD". In this thesis, I want to propose that it is a

good theory for entrepreneurs from another angle. I want to discuss this issue from the macro-economic aspect. The inspiration is from my recent independent research class with Dr. Ye-sho Chen. He introduced me a very interesting book called "The Fortune Favors the Bold" written by Lester Thurow. And after reading most of the book, I came to a conclusion that Tech Culture is just facing one of many problems it will be facing later on. If the two founders can make a correct decision for the problems that Tech Culture is having right now, Tech Culture will go back to the right track again soon. Effectuation theories will be a fit for Tech Culture.

According to Lester Thurow, we are living in the ongoing third industrial evolution. The information knowledge evolution, the result of the evolution will lead globalization. He discussed that lots of companies are converting their business model towards the globalization form to fit the third industrial evolution. Globalization is the trend, but it is not controlled by governments at all. In fact, it will decrease the power of governments in all different levels. There are no guidelines or rules set up for how globalization should go. Globalization has lots of forms because each major corporation will have their own form of globalization development according to its own culture. No one knows where Globalization will lead us to. It might crash the human economy or it might bring the global economy to a new level. He gave three options for companies to choose: Go for globalization, not go for globalization and shape globalization.

Obviously, shaping globalization is the right way to go. It is doomed to fail if someone simply goes for globalization but let it go the way it wants to. In order for globalization to be successful, we have to control the globalization. By controlling it, rules have to be made to shape the globalization to the bright future. Now, can you start seeing the similarities between globalization, the current trend of economic development, and effectuation theories?

Effectuation for entrepreneurs has the similar approach that globalization has. Its approach is to control the unpredictable future rather than predicting and trying to prevent the future problems. If entrepreneurs want to survive in today's economic development with totally unclear future, they will have to use effectuation rather than causation. Causation is good for entrepreneurs who enter a field with a very predictable future where most of the problems have been identified and they can use their own way to prevent all those from happening. But how can they prevent problems from happening if the problems are unknown? Effectuation fits the correct economic development because it encourages entrepreneurs to get into the field they want to get into, and let the effects (means) to lead them to an unknown future. They will meet lots of unpredictable problems. But through practice, they will get a more completed knowledge in their field. Through the knowledge, they can better control the unpredicted problems, and ultimately shape their own future.

Because of this similarity, I am convinced that Tech Culture is going on the right track. The problem it is having right now is just the process of learning the more completed knowledge to gain better control for the future unexpected problems. After analyzing Tech Culture's actions with the effectuation theories and proving that effectuation is a better approach on today's macroeconomic development, I will propose that Tech Culture's two founders should not give up Tech Culture, and I will give suggestions for what Tech Culture should do to solve the problems they are facing now.

3. Solve Tech Culture's Problems

Remember there are 5 questions I listed at the end of chapter one that remain unanswered? I am going to dedicate this chapter to answer these questions with suggestions for what Tech Culture should do in the near future.

Just to refresh our mind, the 5 questions are:

1. Should Tech Culture continue? If yes, what do they need to do to bring Tech Culture back?
2. How can they keep their existing customers?
3. Should Tech Culture continue on the Network Quarantine Software project?
4. Should they start looking for new projects? If yes, what project should they start?
5. What should Tech Culture improve to make their business model more efficient?

The first part of the first question is answered. Tech Culture should continue. In order to answer the rest of questions, I have to start with the "3M". 3M means, in the order of importance from high to low, market, money and management. These three key concepts are involved in all different business models in the economy. No matter how different each model is from another, the 3M is always involved somewhere in their business model. I am going to discuss each element of 3M in the coming three sections and apply the 3M to the Tech Culture business model, in the management section, I will also quote Michael Porter's five competitive forces to further explain what Tech Culture should react with all the problems it is facing.

3.1 Market

In order to start discussing market, first, I will have to ask three very fundamental questions: What is market? What created a market? How is the market created? Traditionally, market is assumed to exist when entrepreneurs open up their new business adventures. Recent research by Dr. Herb

Simon and Dr. Saras D. Sarasvanthy has a new explanation of how market is created. According to Dr. Herb Simon, market is one of the "human artifacts" human creates. It is created by interaction between people, but on the other hand, no one can control when the market is created, as Dr. Saras D. Sarasvanthy said in her paper "Constructing corridors to economic primitives: Entrepreneurial opportunities as demand side artifact": Market is created whenever it is needed to be created.

Market is created through human action and interaction. As people interact with each other, they exchange information. If something is needed by certain amount of people, a demand is created. The opportunity is discovered. It doesn't matter if anybody sees this opportunity, it is created. This is why the discovery of opportunity is objective. Then it is the entrepreneurs' job to recognize this opportunity. The recognition of the opportunity is subjective because different entrepreneurs will have their own view toward this opportunities based on their personal experiences.

Following my explanation so far? Please make sure understand the above paragraph before reading following one because it is about to get more abstract.

When an opportunity is recognized by entrepreneur, they will explore the given opportunity, which means entrepreneurs don't create the opportunity but they view opportunities as the results of their actions!

Confusing, isn't it? Let me explain them in a clear way. First, we have to understand that discovery of an opportunity is objective. It is here no matter people see it or not. The recognition of this opportunity is subjective because different entrepreneur view this opportunity differently. After they discover it, they will start exploring the opportunity using their past experiences. Effectuation theories state that when entrepreneurs start their new business, they will not predict where the business will lead to. It is only that entrepreneurs only use the means to choose many possible

effects, which means, they recognize the opportunity their own way, and they use effectuation process to construct their own version of opportunity through interaction with their social network. In their point of view, their version of opportunity is a result of their actions. This means opportunity is created when market is absent. Market, in entrepreneurs' view, is created by them exploring the opportunities, and through the exploration, their actions and their interaction with their social network created the market.

I assume that my readers understand the abstract argument in the paragraphs above. Now we understand what markets is, how market is created, who creates the market and how entrepreneurs treat opportunities.

So, now, let's see Tech Culture's version of market. The two founders recognized the opportunities of creating automated network security software. They acted on it and eventually, a market for them was created through their interaction with Sia's social network. What is the limit of this market? I will use the phase "sky is the limit" to answer the question. Networking is becoming more and more important in today's business communication. Each business, from the smallest company to the largest corporation has to have their internal network in order to have their business running efficiently. The infrastructures of the network can be from simple to complicated, but complicated network in a corporation is constructed through combining all their internal simple networks together. The Network Quarantine Software can be used in every company that has a network infrastructure, simple or complicated.

As far as the concern of the market, Tech Culture is in a very good shape. With that, Tech Culture should continue with the Network Quarantine Software, only with a new approach. The new approach will be discussed more into detail in the later sections.

3.2 Money

As we mentioned in the first chapter, the start up cost of Tech Culture was close to 0. The biggest investment was the time and efforts two founders devoted to the business. They smartly avoided the cost of purchasing switches and routers by partnering with one of their customers to use their network as a test center.

The main focus in this section is how Tech Culture should use the money efficiently to make the maximum return.

Let's go back to Chapter one for a moment. Tech Culture had a big problem about their customers not wanting to open their networks to Tech Culture founders for remote access because of security reasons. The two founders had to be physically in their customer's companies most of time if there were configuration problems. Can you see my point here? Money can be applied to solve this issue. Tech Culture should use the money they made to contract people from areas where they have customers, and train them to become network administrators with knowledge of the Network Quarantine Software. Every time customers from those areas have problems, they can simply send their local contractor to fix the problem for them. By paying the contractor to fix those problems, the two founders can be more focused on the software development and customer relationship to expand their business.

3.3 Management

In order to make correct decision for managerial strategies, Tech Culture must understand the industrial structure it is in. In this section, I am going to use Michael Porter's Five Competitive model (See Appendix B for the model diagram) to show what Tech Culture should do for their managerial strategy.

In the Tech Culture case, suppliers don't play an important role because Tech Culture is a consulting business, and they do business in the form of information and knowledge; plus, they chose open source technology for development. The cost of the technology they chose is free. The only supply Tech Culture needs is two reliable computers for development purpose.

The other 4 factors play very important roles for Tech Culture's decisions on their strategies. I am going to explain how they affect Tech Culture one by one.

3.3.1 New Entrants and Competitors

New entrants are fairly easy to enter the industry Tech Culture is in. Reasons are: low initial investments, low fixed costs, no raw material is needed. Almost everyone who has network security knowledge can enter this field easily, and start competing with Tech Culture. There is one advantage that Tech Culture has and other competitors don't have: Tech Culture has already had customers invested in their software, the switch cost is not incredibly high, but it is high enough to keep the customers with Tech Culture if Tech Culture can provide what they need. This builds the barriers for new entrants.

This barrier, on the other hand, can be broken very easily by major corporations. Just like the one which announced that they will have a hardware level network quarantine switches/routers coming up in the near future. This major corporation will join the industry, and since they have the sounded name brand, plenty of financial support and resources, they can easily become Tech Culture's major competitors and quickly drive Tech Culture out of business, what should Tech Culture do to avoid competing with this kind of major corporation and still survive in this industry they are in?

First of all, we have to understand this is an industry that is very easy to enter because of low initial investments and low fixed costs, which means companies in this industry can walk away

from this industry very easily without any major loss. Tech Culture can walk away from this industry very easily. But I have already proposed Tech Culture should stay in this industry. Why? The answer lies on the next section.

3.3.2 Substitutes

Since we have been using Tech Culture as the representative of the industry they are in, we all have a wrong impression that Tech Culture should be the center of the industry. In fact, Tech Culture is not at the center at all. Tech Culture, after all, can only be defined as the innovator in this industry. According to Dr. Scott A. Shane, every Technological industry has its own standards, and innovators are the ones who create the standards. The question here becomes: Does Tech Culture has a big enough influence to create an industrial wise standard? The answer is not yet. It is still a regional company, even though it has a lot of potential to grow. This is exactly why it is such a big fear for Tech Culture when a major corporation announced to join their competition. That corporation is big enough to influence everybody in this field. The reason they can drive Tech Culture out of business is they can easily use their established brand name to set up the standards quickly. They can set the standards to a point that Tech Culture's software will become totally useless. So, what can Tech Culture do? Well, try to think out of the box, if Tech Culture has 0% of winning chance competing with this big corporation, is there a way for Tech Culture not to compete with this big corporation?

As I proposed, Tech Culture should stay in the industry, which means the answer is yes, there is something else Tech Culture can be in this industry without being a competitor of this big corporation. The role Tech Culture can play in this industrial is to be a substitute, which makes perfect sense. According to Michael Porter, the definition of substitutes is "an alternative product with lower prices of better performance parameters for the same purpose." The big corporation's

product is at the hardware level, Tech Culture's product is on the software level. The cost of a new switch from the major corporation is predicted as at least $20,000, while Tech Culture's product is only a few thousand dollars. People who don't want to invest tens of thousands of money for a new switch can buy Tech Culture's software. The only thing Tech Culture should pay attention to is how the new switches are going to work. Instead of creating their own standards, Tech Culture should closely monitor how the network security is done by using the big corporation's way. This actually creates an advantage for Tech Culture. They can save their limited resources on researching for the standards and use the resources to modify the software to fit the standards the big corporation creates. This is more productive. If Tech Culture does the research on its own, it might be going towards a totally wrong way because of their uncompleted knowledge. Their knowledge is highly local while the big corporation has a very completed knowledge about networking and network security. The standards set up by the big corporation will contain much less bias. Tech Culture can adapt these standards by modifying their software. Since they have reasonable numbers of customers invested in their product already, if Tech Culture can keep a good relationship with their customers, Tech Culture should be able to have a fair size piece of the big pie.

Here is what they should do to shift their position in the industry to become a substitute:

1. Closely monitor the development of the new switch from the corporation.
2. Slow down their development of the Network Quarantine Software.
3. Focus on technical support for their customers. After the one year free support is done, they should charge their customers by each problem they need support rather than providing support by charging a flat fee for a fixed period of time.
4. Tech Culture doesn't release a new version of Network Quarantine Software until the big corporation releases their product.

5. Once the new switch is released, Tech Culture should buy it as soon as possible. Then store it in their test center to learn how the big corporation handles the network security problems.
6. After having a clear understanding of the new switch, the two founders should have meetings to brainstorm themselves with ideas for improving their software according to what they have learned from the switch.
7. Start engaging a development for the new version of the software. The software should follow the same general logic to do the job as the new switch does.
8. Apply creative ideas on how to make the interface more convenient to use.
9. It is better that the software can interact with the new switch to improve its performance.
10. Once the software development is completed, set an affordable price to attract small local companies.

3.3.3 Customers

Now that we understand that Tech Culture can still stay in the industry it is in right now, but in order to stay in the industry, Tech Culture has to keep a good relationship with their customers. How can they do that?

Before answering a how question, a "what" question should always be asked first. The "what" question here is: what kind of customers should Tech Culture focus on? The answer will come to the surface with this logic: Tech Culture will change its role in the industry and play as a substitute to avoid competing directly with the big corporation that is about to enter the industry and the advantages for being a substitute are using Tech Culture's limited resource more productively by adapting the big corporation's industrial standards rather than creating its own and lowering price for the software that does the same thing as what the big corporation offers with their new product. Customers who would pick Tech Culture's products are the ones who can not afford the

big corporation products. This means Tech Culture's customers should be mostly local small to mid-size companies. They, just like Tech Culture, have very limited resources. If Tech Culture can provide a solution to them with a cheaper price (less than half of the cost for a new switch) and if Tech Culture's product can keep them happy, Tech Culture will win the customers.

Now, let's return to the question of how Tech Culture can keep their customers happy. Dr. Blake Ives's Customer Service Life Cycle (CSLC) provides the answer for it (Please see Appendix C for the diagram). According to Dr. Ives, CSLC "is intended to help you differentiate the various stages your customer goes through in acquiring a product/service from your firm."

Requirement stage is when customers start establishing a need for the product and determining what attributes the product should have to fit their needs. This initial stage is very critical for Tech Culture to win the customers. Tech Culture should study their customers in depth to discover what their real needs are. Here, Tech Culture would need to understand that customers, due to the lack of knowledge, might not be able to fully express their needs through oral conversation. Tech Culture needs to understand the situations their customers are in, and put themselves in their customers' situations in order to get a complete understanding.

In the acquisition stage, customers would decide where to get the product, how to pay for the product and after obtaining the product, they would decide if the product fits their needs. What Tech Culture should do in this stage is to do analysis to their customers to find out what features in the software are the core features while other features could be add-ons to fit each customer's needs. By doing this, Tech Culture would be able to provide highly customizable software to each customer to meet their individual needs.

The third stage is ownership. In this stage, customers like to control and use the product, upgrade products and repair products to fix their changed needs. What Tech Culture should do is

after selling the software, they should still keep in contact with the customers, constantly contact their customers for their opinions about the software, listen to customers' suggestions about the software and react with the valuable suggestions, such as providing patch updates to solve customers' complains and fix bugs. Keep customers' software up-to-date by providing timely upgrades when some new threats are formed. Remember we mentioned Tech Culture should have local contractors for providing timely and satisfying supports to customers? That action is perfect if it is put into this stage.

The final stage is retirement stage. In this stage, the product might be outdated, and customers start looking for new ones. They would start monitoring the expenses of the software they have also to see how much it is to maintain the product. Is there a better alternative out in the market? What Tech Culture should do is to push out a new version of the software periodically with new valuable features added on the software, on the other hand, keep the software price affordable. Maybe give a discount to customers who have an older version of the software. Local contractors should be used for supports for installing and configuring new version of software also.

3.3.4 Learn from Experience

Business theories are very important to direct Tech Culture on the right track. As we can see above, we have used various theories to give Tech Culture suggestions for their problems. On the other hand, the two founders have to learn from the past experiences, both winning or failing. One of the biggest mistakes Tech Culture made was to totally trust the third person during the help desk software development. The reason for this was because Sia and Jack had such a great business partnership. They trusted each other without a doubt from the beginning. This great partnership impression expands to the third person. They totally assumed that they will be in the same happy

partnership with the third person joining. They became over-optimistic. This caused them a big down fall.

In the future, the two founders should be very cautious on selecting a new partner or a new employee. They can look at the resume to choose who the best candidate is, but once they find the best candidate, they should not be optimistic at all. They should not give the new person an important role in the development. They have to give him non-critical pieces at the beginning to test his/her abilities. They should meet maybe once a week for each person to report what they have done in that week, and see if each of them reaches their project goal. Each person should also present their plan for at least the coming week in the meeting. In order to have a meeting every week, they can't have their new person far away from them. The new person has to either live very close to Jack or Sia.

3.4 Conclusion with One Unanswered Question Left

With Dr. Ye-sho Chen's 3M, combing Michael Porter's 5 Competitive forces theory and Dr. Blake Ives's Customer Service Life Cycle, I created a few very theory driven suggestions for Tech Culture: There is a huge market for Tech Culture's growth, but in order to survive in the industry, Tech Culture has to avoid competing with major corporation competitors who are entering or who will enter the industry. To do so, Tech Culture has to shift their software focus and make it become a substitute of the product that is created by the major corporation. Tech Culture can save their resource from creating standards for the industry by doing so. Instead of creating the standards, they should adapt the standards that are created by the major corporation. This is better for Tech Culture because the major corporation has a very completed knowledge in the industry. The standards it set up should contain very little bias. Tech Culture, by adapting those standards, will not easily be left behind in the industry. This lowers the risk of being phased out of the industry. Then Tech Culture

should focus on local small to mid size companies because they have limited financial abilities. The product from the major corporation might not be affordable for those companies. Tech Culture should follow the Customer Service Life Cycle to keep a good relationship with their customers for possible future expansion.

By now most of the question from the end of Chapter 1 should be answered except question 4: Should they start looking for new projects? If yes, what project should they start? At this moment, Tech Culture should not start a new project because they should focus on shifting the role they play in the industry and keeping a good relationship with the customers. Because Tech Culture is using effectuation process to drive their business forward, they should not have a fixed goal. They would start a new project when the means go to them. When is it going to be? No one knows. How do the means go to them? No one knows. What product is Tech Culture going to start with the new means they will see? Again, without knowing the means, no one knows. This question remains unanswered for Tech Culture's own good because without a fixed goal, the two founders can use their imaginations to create surprises for their own entrepreneurial adventure.

References

Sarasvathy, Saras D. and Read, Stuart, *Encyclopedia of Entrepreneurship*

Sarasvathy, Saras D., *Constructing corridors to economic primitives: Entrepreneurial opportunities as demand-side artifacts*

Sarasvathy, Saras D., *Causation and Effectuation: Toward A Theoretical Shift From Economic Inevitability to Entrepreneurial Contingency*, Academy of Management Reivew Vol. 26, (2001), 243-263.

Sarasvathy, Saras D. and Menon, Anil, *Failing Firms and Successful Entrepreneurs: Serial Entrepreneurship as a Simple Machine*, (2002).

Sarasvathy, Saras D. and Menon, Anil, *Failing Firms and Successful Entrepreneurs: Serial Entrepreneurship as a Temporal Portfolio*, (2002).

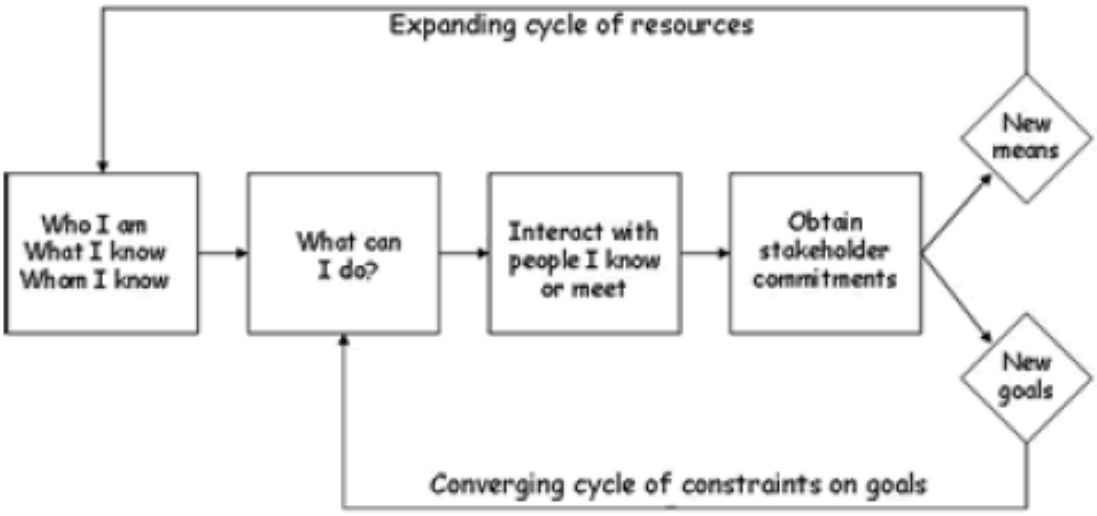
Shane, Scott A., Finding Fertile Ground Indetifying Extraordinary Opportunities for New Ventures, First Edition, Wharton School Publishing, 2005.

Thurow, Lester, Fortune Favors the Bold, First Edition, HarperCollins Publishers, Inc., (2003).

http://www.themanager.org/Models/p5f.htm

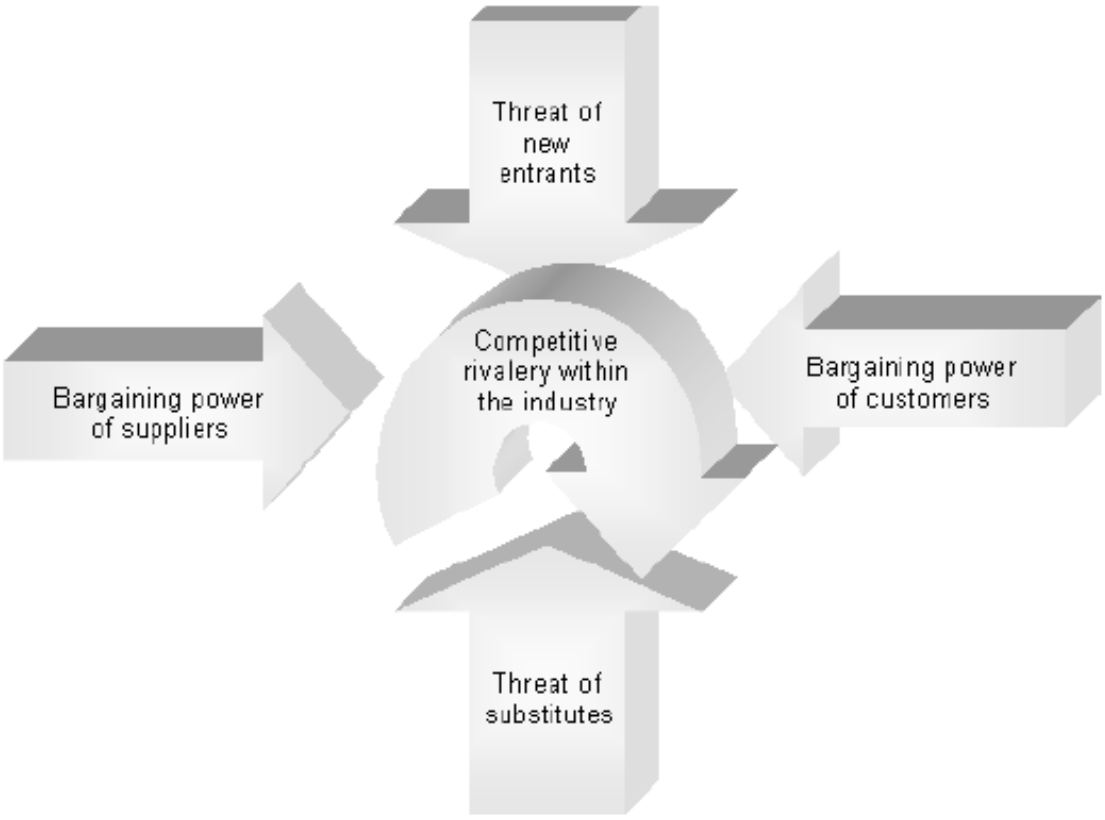
http://isds.bus.lsu.edu/cvoc/projects/cslc/html/intro.html

Appendix A Dr. Darasvathy's Effectuation Process Diagram



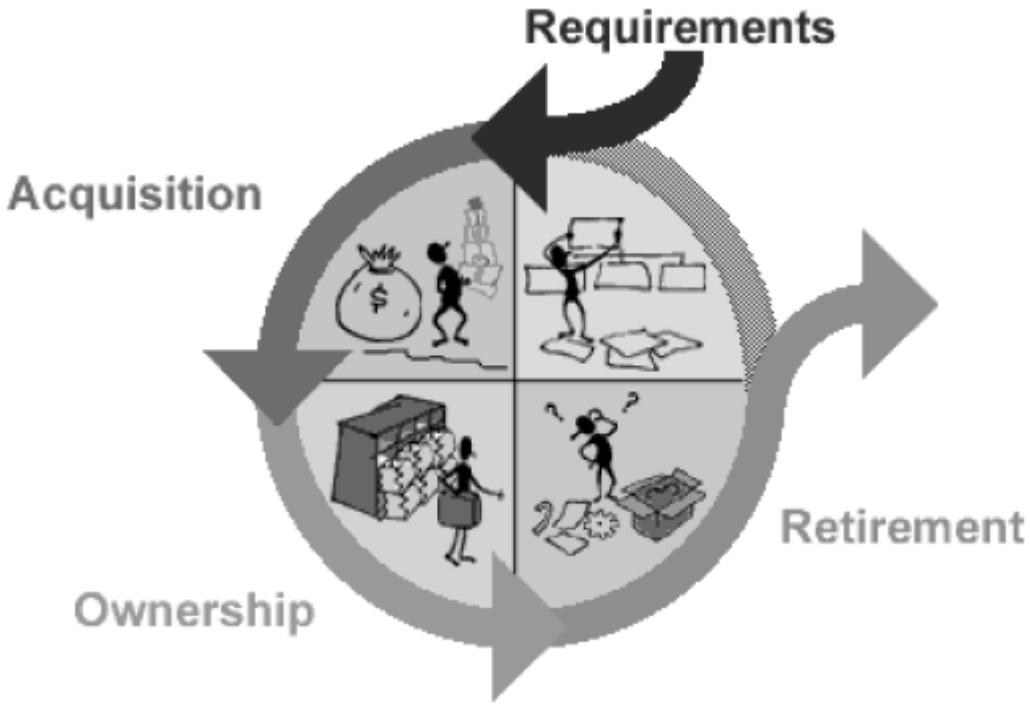
Source: Sarasvathy, Saras D., Causation and Effectuation: Toward A Theoretical Shift From Economic Inevitability to Entrepreneurial Contingency

Appendix B Dr. Michael Porter's Five Competitive Forces Diagram



Source: http://www.themanager.org/Models/p5f.htm

Appendix C Dr. Blake Ives's Customer Service Life Cycle



Source: http://isds.bus.lsu.edu/cvoc/projects/cslc/html/intro.html

Vita

Yi J. Cheng was born on April 26, 1980, in Guangzhou, China. He completed his high school education from Trinity Episcopal Day School in Natchez, Mississippi, in May, 2000. He received the degree of Bachelor of Science in computer science from Louisiana State University, Baton Rouge, Louisiana, in May 2004. Later in August 2004, he was enrolled in the Department of Information Systems & Decision Sciences at L.S.U., to attend graduate school. He is currently a candidate for the degree of Master of Science to be awarded from the Department of Information Systems & Decision Sciences.