SAS® GLOBAL FORUM 2020 #SASGF

Paper 5185-2020

Finding Your Seat at the Table

William Coar, Axio Research – A Cytel Company

ABSTRACT

Have you ever found yourself wondering how to take that next step, wishing you had more of an impact, participating in discussions about how to move your organization's ideas forward? Do you think you warrant a seat at the strategic table? The answer is likely yes, but the opportunity and respect will need to be earned. The programming profession is dominated by technical expertise, but this alone will only advance you so far. Leadership skills must be learned and applied to make a broader impact within your organization.

There are many definitions of leadership, but most are variations of a common theme. We view leadership as the ability to influence others to take specific action when they have the freedom to choose to do otherwise. This fundamentally differs from management. Many leadership principles are established and are not industry specific. Our technical training rarely emphasized or even taught these principles. Further, many of us are introverts and may not feel that the publicly visible skills required for leadership come easily. These are not prohibitive factors. Leadership principles can be learned and practiced every day, but a strategy must be developed. Your efforts must be intentional.

In this paper we apply leadership and acumen principles to our programming profession to help you transition from an individual contributor to an organizational contributor, and find your seat at the table.

INTRODUCTION

To be an integral part of a decision-making process, you need to obtain the trust of key decision makers as well as demonstrate your sound judgement. Just to a few, words such as impact, influence, trust, and respect are elements of leadership. Many leadership principles are well established and are not specific to any industry or profession.

As a statistical programmer you may be closer to the technical and practical challenges that can influence policy and procedure. But more is needed than technical skills. In this paper, I offer insights into how you can leverage your technical skills with management and leadership skills to further your career. My hope is to do this through the point of view of a student rather than a teacher. I am fortunate to have leadership mentors for the past 20 years, who have encouraged me to continually grow in this area. There is no pinnacle or finish line, in my opinion.

Suppose you were running code and noticed a message in your SAS log that your license is about to expire. Have you ever wondered what efforts are required to renew the license? Are there any possible upgrades (aka maintenance releases)? Does the department need to coordinate with IT? How much effort is required for validation of the maintenance release? Are there internal tools such as macro libraries that have already been through a rigorous testing process? Do you need to repeat the process? Perhaps most critical, are there any impacts on business continuity?

Such questions go beyond just thinking about day-to-day work and stretch into the operations and strategic planning of an organization. These are the types of questions that programmers who are looking for a seat at the strategic table should be asking themselves. As a consequence, they should bring solutions to the team and contribute meaningfully to the discussion. This is a fundamental characteristic of leadership.

When I stared my career in 1995, I was a SAS programmer that rarely talked or interacted with my colleagues unless I had to. With the help of a mentor, I started to grow and ask more questions. Even so, I defaulted to my introverted tendencies as an excuse to avoid venturing outside of my comfort zone. Besides, I was quite satisfied with my career trajectory. It wasn't until my return to the workforce after leaving for graduate school did I realize the limitations of the trajectory I was on.

I have been a student of leadership for the past 10 years. I remain engaged and continue to grow in this area. In 2013, Bob Rodriquez, ASA president at the time, presented on leadership principles at professional conference (DIA\FDA Statistics Workshop). This opened my eyes to their application in my career. Since then I have collaborated with others to present on these topics at SAS Global Forum, Western Users of SAS Software annual conferences, the Denver SAS Use group annual conference, and at the Conference for Statistical Practices.

Much of following discussion transpired from the American Statistical Association course on Preparing Statisticians for Leadership, presented by Gary Sullivan and Fang Chen in 2019. At the conclusion of the course, we discussed applying these principles to the programming profession, which included sharing these ideas at SAS Global Forum. Throughout my own hand-written notes are remarks about the application of these concepts to the programming field, many of which are discussed below.

LEADERSHIP VS MANAGEMENT

In an interview with Success Magazine, Seth Godin describes management as the quest to get people to do what they did yesterday, but a little faster and a little cheaper. This relates more to management with respect to production line manufacturing as was the case in the industrial revolution. Management, however, is changing. There are many other factors to consider such as quality, people, reputation, and client relations. In fact, this was the talking point of the interview: there is more to leadership than many of the traditional views of management.

There are many definitions of leadership. I have found that moar follow the common theme of the ability to influence others. In fact, this is the second law of leadership mentioned in [1]. I have grown fond of the definition presented by Sullivan and Chen in [2];

> ***Leadership is the ability to consistently deliver value to an organization or cause by inspiring people to take specific action when they truly have the freedom to do otherwise*.**

A remarkable distinction of leadership is the drive to influence others to advance a cause when they truly have the freedom to do otherwise. People are rarely required to do as they are asked. Rather, a good leader, regardless of title or position, has the skill to build the belief and trust that motivates people to follow.

But from a big picture, increasing capacity of existing resources is a part of business. This is very relevant to SAS programmers. Throughout my entire career in the pharmaceutical industry, I've always observed efforts to increase our production output with fewer programming resources, primarily driven by a lack of SAS programmers with industry experience. Processes and systems are implemented and refined to increase quality and efficiency, and are often enhanced and fortified with advances in IT. While managing this drive for quality and efficiency is relevant in today's programming environments, this effort differs from leadership. However, a good manager can use leadership skills to ensure quality and efficiency, which are important to organizational impact.

The topic of this paper, "finding your seat at the table" means leveraging your knowledge and experience as a statistical programmer to generate influence and impact within your company. It's about transitioning from contributing at the individual level to contributing at the organizational level. To do this, you need to have the critical thinking ability to link that experience to business process in a way that brings positive change. You need to possess the vision of the bigger picture. In [1], John Maxwell ties this to leadership with the Law of the Lid – your ability to have an impact is bounded by your leadership skills. As you seek to be involved, your leadership skills need to be developed.

How can we, as programmers, increase impact in our organization?

THE FOUNDATION: LEADERSHIP PRINCIPLES

When someone asks me about leadership, words like influence, service, authenticity, and belief come to mind. I usually ponder about the books and articles I've read when trying to answer this question, and am reminded that there are many more. The discussion below will discuss influence, motivation, trust, empathy and connection, and how they related to a statistical programmer.

INFLUENCE

"The true measure of leadership is influence. Nothing more, nothing less." This is the second law of 21 Irrefutable Laws of Leadership" by John Maxwell[1]. This book was my first formal introduction to leadership. Consequently, I adopted this concept about influence early in my leadership studies. However, over time I have learned that there is much more to it. Influence is the ability to affect the character, development, or behavior of someone or something. Influence is often subtle. How and why certain people can influence others is likely the synergy of many individual attributes.

I routinely receive guidance from with one my earliest mentors. With his counsel, I've grown from an entry level SAS programmer to a director level statistician. My only experience with SAS in undergraduate studies was to do some simple summary statistics for a consulting project. Needless to say I had a lot to learn. I first connected with my mentor at this key point in my early career. There was something unique about his ability to drive me to be and do more. He wasn't my boss, yet I never balked at working extended hours or weekends if he asked. Why did he have so much influence?

He regularly took the time to not only help me learn the technicalities of SAS programming, but further "why" we were doing what we were doing. He helped me understand how data management, programming, statistics, and writing relied on and impacted each other. He helped me believe that my work mattered to the greater goal. He took the time to explain the statistics resulting from my programming and output so that I could understand how it was intended to be used. His encouragement with statistics led to my graduate studies at Villanova and Colorado State University. He believed in me then, and he continues to believe in my growth today.

His deeper interest in my development inspired me to help when he asked, not because I had to, but because I wanted to.

INSPIRATION & MOTIVATION

Inspiring and motivating others are one way to have influence. But what is the difference between the two concepts? The discussion in [5] has the following:

> **Quote:** *"Motivation is when you get hold of an idea and carry it through to its conclusion, and inspiration is when an idea gets hold of you and carries you where you are intended to go."* ***Dr Wayne Dyer***

When leaders inspire and motivate others, they can do so in a direction to advance the goals and objectives of each individual, the team, and the company. Each team member may have the freedom to do otherwise, but there is an inherent desire to see it through.

In [6], Pink suggests we encourage intrinsic motivation to maintain progress rather than simply providing carrots and sticks (money or punishment). Individuals more likely to contribute when something matters to them personally. This is intrinsic motivation. We inherently have a desire to be part of something bigger than ourselves. Pink suggests using autonomy, mastery, and purpose to do this. See Pink's TED talk titled "The Puzzle of Motivation" if you are motivated to dig deeper.

Autonomy

Have you ever stayed up late coding because there was some inner drive for the accomplishment? Was it something new and innovative that you were excited to finish? Was it a creative new way to simplify a complicated programming task? Personally, I have spent many nights in hotels while traveling, and even some early mornings, connected to servers writing SAS code simply because I was excited about something new. Having the freedom to do what I want, and when, are examples of autonomy. For me, this autonomy has resulted in ideas to improve upon quality and increase efficiency.

About 10 years ago, I started a new job as a remote worker in the service industry. I oversaw projects, but not the actual programmers. As deadlines for my first deliverable approached, I had about 50 summary tables in individual documents. I needed to put them into a single file, ideally with a hyperlinked table of contents and bookmarks. I had managed to avoid this clerical task for 15 years of my career as it was likely done by medical writers or their assistants. There were no existing internal standards for how to complete this task. Some people did it manually, some used visual basic code. The challenge I faced in that limited resource setting was that all visual basic code was on individual computers and not a central server. Not knowing visual basic, I decided to try the best tool I had available at the time and use SAS. The idea was to call Dynamic Data Exchange to send commands to MS Word. This process was easily automated, especially since we had a pre-defined list of outputs in a program tracker. I was excited, and set off on a very late night. The outcome of my efforts was SAS code to generate a single file deliverable that was automated and independent of the user. My autonomy resulted in a technical improvement for the whole team.

Mastery

Many programmers have an inner drive to take what they have done and make it better. I have a desire to constantly improve my coding. Mastery is the result of an individual becoming skilled at something that matters to him or her personally. Have you ever had difficulty focusing on programming solutions that made your assigned tasks easier and more efficient?

While DDE saved many hours of manual labor and provided a higher degree of certainty that everything was included and in order, DDE itself has limited support. I knew we needed to find an alternative to address future limitations. There were many people trying to post-process actual output files (either using SAS or scripts), but I was intrigued by an idea resulting from a conversation with a friend from SAS. Rather than post-process individual files, "replay" the output into a single file using ODS Document, a newer capability in SAS. This turned out to be much simpler to automate, and far more stable. Having the ability to learn something new, something that few (if any) others were doing, something innovative that had direct application to my work was exciting.

Purpose

In terms of purpose, Pink argues that we all have a natural desire to contribute to a greater cause. There is a sense of accomplishment that we feel when our work contributes to a something important, something larger than ourselves. We become inspired, and this drives us to do more. I liken this to the "why" discussed by Sinek in his TED talk titled "How Great

Leaders Inspire Action”. Your “why” is the reason you get up excited to go to work each day, rather than just looking for the paycheck. Remember our definition of leadership suggests people have the freedom to do otherwise. From a leadership perspective, when people are inspired by your “why”, they want to volunteer to help; being told to help is not required.

The recent trends with Open Source coding are a great example of purpose. So many programmers are now going to extreme lengths to make code freely available for others. Collaborating with other programmers seems to only drive us to contribute more to the cause. Given the rise of R and Python, and the integration with SAS, implementing new innovative programming ideas is becoming easier, and are more easily shared with others.

See [7] for a more detailed discussion on “why”.

TRUST

Have you ever been in a situation where a programming task needed to be completed, and even though inconvenient, you said “it will be easiest if I did it myself” or “it would take too long to explain it to someone else”. Could these have actually been issues with trusting a colleague to do the work? I have had times where there was certainly an absence of trust – trust that someone will mess up my code, trust that they will not get it done in time, and even trust that they fully understood the problem at hand. This was far more common for me earlier in my career. Later in my career, I had to learn and rely on trust. I contantly work to improve in this area.

Trust is something that is not freely given. It is earned. It is also something difficult to quantify. I have read that others suggest trust is a complicated emotion but leave the detailed discussion to the psychologists.

In [3], Sinek suggests that to build trust, we must take a risk and be vulnerable. In fact, he says that trusting relationships require both parties to take risks. If a relationship is to develop, someone must be the first to start, and the other must reciprocate. While this is may seem superficially logical, I found this to be a different way of thinking about trust.

As programmers, we can be the person to take the first risk. We can also be the one to reciprocate when the opportunity presents itself. Either way, you will likely be outside of your comfort zone. This is a part of being vulnerable.

Trust is also a requirement in building trusting teams. Most of us do not work alone. Having a trusting team allows us to take advantage of each individual’s strengths and allows us to let others contribute more. In turn, we can also delegate tasks to others who may be better suited and more motivated to perform that task than we are.

We should first work on trust with those around us. A perceived lack of trust with our peers will be noticeable by other leaders.

EMPATHY

During a leadership workshop at PHUSE Connect 2019, the instructors presented on the concept of Design Thinking. Design thinking can be viewed as an iterative process where we seek to understand the end user, challenge assumptions, and identify alternative strategies to solve problems. A large component of the workshop focused on empathy – the ability to understand and share the feelings of another. This is the ability to put yourself in someone else’s shoes to better understand their point of view. This reminds me of the first seat at the table that I ever was invited to join.

In the late 90’s, my employer sent me to the corporate offices with other management for a workshop on rapid applications development. It was an attempt to bring “users” of in-house computer applications together with the “developers”. When users and developers don’t communicate well, the users are stuck with whatever the developers decide to give, and it likely deviates from what the users actually need. While this isn’t exactly using empathy, empathy should have been a component of the workshop.

In your programming experience, has anyone asked you to generate some output that they (not knowing the details) assumed would be easy, but you (knowing the data) felt otherwise? Did you leave the conversation frustrated because you assumed they didn’t understand what they were talking about or the difficulties that existed for you? An example of this was very apparent in an experience I once had with some clinical scientists and medical writers.

I was the primary programmer and assisting my colleagues with data review. They were requesting a listing of raw data on prior medications. Going through the proper channels, I provided them with a (lengthy) document that listed the patient number, medication (verbatim term), start/stop dates, and various other fields collected. This was a simple request that had very limited utility, as it turns out. First, having to page through hundreds of pages of data was time consuming, frustrating, probably caused migraines, and ultimately was not useful for the clinical scientists. Second, it was a repetitive task on cumulative data. Thus, they found themselves having to re-review data without knowing if anything changed which did not provide the information relevant to their task. Having done numerous types of validation of SAS output that required me to perform similar tasks, I was eventually able to relate to their experience, but not at first.

I clearly remember sitting in an office with one of the clinical scientists when I finally comprehended what they were trying to do. The listings were used to perform a review of the patient population to make sure they actually met a key inclusion parameter. It was critical to have a specific patient population for regulatory approval. This required that they review the verbatim terms of each medication a patient received prior to study entry and to verify that the patient did in fact meet the criteria. This suddenly changed how I viewed the request. It also helped for me to listen about their frustrations with the way the data were presented. Simply changing the sort order could have helped substantially. The frustrations we were all experiencing were simply due to a lack of my understanding of how they wanted to use the listing that I was producing. We subsequently worked together to provide a useful solution – using a unique record ID in the database to compare a current snapshot of data with a previous snapshot to identify what was new, deleted, unchanged, or changed. Furthermore, if a data point changed, we were able to show what it changed from, and what it changed to. Having a better understanding from the clinical scientist point of view together with a programmer’s understanding of data capture and database structure proved to be a successful combination.

CONNECTION

In this day of apps and smartphones and social media everybody communicates, but few connect. Connection is the focus of one of John Maxwell’s books. Building connections with others is part of being human, and it is part of our day-to-day work. According to Brene Brown in [8], connection gives us purpose and meaning in life. Since many of us spend much of our day online, perhaps real face-to-face connections can help at work.

As a student, my eye catches many articles that say little things matter. I see this over and over. In fact, it was even the topic of several articles posted on LinkedIn by the Harvard Business Review. Keeping the little things in mind is one way to build connections with our coworkers, but to start, we first need to talk.

Have you ever sat at your desk and put some headphones on, and stared at your computer to avoid making eye-contact with someone as they walked by? Sometimes this is needed for focus on detail, but other times, this is due to social laziness. We need to learn to recognize when this is the case to better understand why we do it. Is it because of our introverted tendencies? Is it because we want to avoid intra-office gossip? Or do we feel intimated by others, particularly management? All of these are valid reasons to want to hide, but they also make building meaningful connections more challenging.

Building connections takes time, trust and empathy. It also requires listening. Many of say we are listening, yet at the same time we are formulating a response before the other person is done describing the problem. I suspect I'm not the only programmer that sometimes develops an algorithm to solve a problem while the other person is still talking, before the problem is fully specified. True listening involves letting the other person fully describe their problem and their point of view and actually hearing what they are trying to say. One way to achieve this is to develop a habit of repeating what you heard in your own words. This not only helps with clear communication; it also tells the other person that you are actually listening to them.

Building connections also requires us to go beyond the walls of our cube or office. There is more to life than work. When we feel a connection to others, we have a better understanding of who they are as a person, why they come to work every day, and want motivates them in life. We don't have to be "besties" with the people we work with, but to build a connection, we have to recognize that our colleagues are human beings. Computer algorithms and logic do not apply to building connections. It requires being vulnerable, and sometimes even emotional.

BUILDING THE FOUNDATION

The first law in [1] is the law of the lid – leadership ability determines a person's level of effectiveness. If we want to increase the impact we can have in our departments, if we want a seat at the decision-making table, if we want to transition to contribute to the organization's vision, we need to develop our leadership skills. But in addition to leadership, we need sound decision making skills. We now turn to the concept of acumen to increase the overall impact we can have at the table.

PILLARS OF ACUMEN

Beyond leadership, we need to also work on our ability to make good decisions. There of many books on the concepts of critical thinking and decision making. At a very basic level, we make decisions based on information, or data. The pillars of acumen can help us think about the big-picture, and how many factors can go into any decision making process.

The pillars of acumen is a concept presented by Dr. William Wang in [2]. By definition, acumen is the ability to leverage good judgments to make quick decisions. For success, one must possess insight into both their organization's purpose and how it functions. The three pillars that support acumen are technical, business, and social. Developing and understanding each pillar will aid in our ability to make good judgements and quick decisions.

TECHNICAL

Technical skills are the strength of most programmers that I know. We all know that technology changes quickly. It takes focused and deliberate efforts to keep up. I suspect many programmers look back at some of their code and say that more current technology can often simplify. For example, today we have ODS in SAS to create RTF files. In 1998, creating tables for MS Word often required the use of DDE and other complicated process just to have rows and columns that could be easily manipulated by writers. Just because something worked 15 or 20 years ago does not mean they are the best option now.

To develop technical acumen, we need to be aware of the technology landscape. We need to recognize and prepare for eventual changes. There are many ways for programmers to stay on the leading edge. The first one that comes to mind is to be active in professional organizations and be involved in professional conferences. Note I did not say "join" professional organizations and “attend” conferences. Many of us do that already but fewer are actively participating in the initiatives of professional organizations. This not only keeps us informed about the direction technology is going, it helps a great deal in the development of other leadership skills. Volunteering and presenting are ways to be involved at conferences. In doing so, we are presented with many opportunities to grow, personally and professionally. We are also presented with opportunities to improve our leadership skills and network with others that can help us on our journey.

Technical acumen extends beyond software and operating systems to areas such as data standards and even data privacy. In the pharmaceutical industry, data standards are an extremely important trend. They can also be challenging. Even though they are standards, they can change over time. Currently, standards are required for submission to FDA. Rejected submissions due to non-compliance are not viewed kindly. They cause delays and disrupt the submission process. By far, programmers have the deepest understanding of the data as they transform the original database to submission ready data, including analysis datasets.

While I have no experience with artificial intelligence or machine learning, they are certainly a hot topic data science. Their emergence comes from the ability collect massive amounts of data and perform complex algorithms, neither of which were available 20 years ago. While this may not apply to my current day-to-day work, I do find myself wondering when it will be.

BUSINESS

Every business has a purpose. For many, it is to make money. For some, it is to advance a cause, but finances are always needed to do so. Have you ever had discussions with your supervisor to better understand “the business”? This may be the mission and vision of your own company, or industry trends. Both are important when it comes to business acumen.

Many companies claim to have a vision, though Sinek argues in [3] that many are not actually a vision. Vision is not “meet a number” or “be the best”. True vision may not even be a finite measure, or realistically achievable. Rather, it is more related to “just cause”, or the “purpose”. Given my profession, I follow a number of pharmaceutical companies on LinkedIn. I often read about quarterly earnings and addressing unmet medical needs. Personally, this does not inspire me to turn on the computer every day. Making a difference in the lives of people when they have no other options for treatment, or giving someone another year of time spent with their family is much more inspiring. In terms of vision, we may not know the impact of our work at the onset, but we continually strive to make a difference.

Beyond the “why”, we also need to understand the business finances, market, and strategy. This may not be easy. These are not typically offered any a math or statistics program, but they are essential to the operating a successful business. Obviously, we need to understand how our departments are contributing to the business, but we also need an understanding of other business units as well.

Have you ever found yourself annoyed that you needed to fill out a timesheet? I suspect I am not alone here. While some of us may simply wonder why we are burdened with this requirement, our finance group will feel otherwise. There may be cases where this is a legal requirement such as projects with government contracts. Finance may also be under pressure to bill clients as soon as possible rather than waiting until the next billing cycle.

Other cases may involve hourly work where billing needs to accurately reflect the hours spent and services performed. When we put ourselves in the position of someone in finance (ie, empathy), maybe we can better understand the frustration and potential liability they may have by delays or inaccurate time keeping. In a service organization, timesheets may also be used to better understand how time is spent to perform certain services. This may help with monitoring out-of-scope work of current contracts and plan more accurately in future contracts. To be fully transparent, I never was good at completing a daily timesheet. This continues to be an area of constant struggle for me. Having said that, we all have areas where we can improve. The key is to understand what they are and develop a personal plan for improvement.

Programmers in a service company have a far better understanding of how time is spent than someone in finance or business development. Input is often critical in the contract development phase as well as resource planning. Personally, I have learned that collaborating with business development is critical in most contracts. While we have the technical understanding, we tend to be more protective of our time and resources. From a business development perspective, it is fair to say that initial contracts are often our “best guess” at the efforts required. Sometimes there is more interest in getting contracts signed, with the understanding that contract amendments will be required. I do not necessarily agree with this, but I acknowledge sales and marketing are not my background or training. Negotiations are real, and I need to trust that business development will act in our better interest.

Let us not forget about ethics either. They apply to programmers and statisticians as well. We look to advance the business goals without compromising work ethics. Many of us have knowledge of extremely confidential information in our day-to-day work. Most of us accept this level of confidentiality. We must be mindful of risk at all times. In the pharmaceutical industry, this could impact not only the businesses we work with, but patients participating in drug development, and future patients in need of new treatment options.

Once we better understand the business goals, we can then begin to identify areas where we as programmers can contribute on an individual level.

SOCIAL

We live in a social world. Human beings are social animals. Social dynamics are real, especially today when social media has such control over lives. During my early days of SAS programming, I would argue against this. I was more socially awkward than social and continue to be to some degree today. When we think about the social perspective of acumen, we try to better understand human dynamics within the organization. Not only the connections we make, but connections others make within the company. Remember that with acumen, we need to go beyond our selves to understand the dynamics that exist.

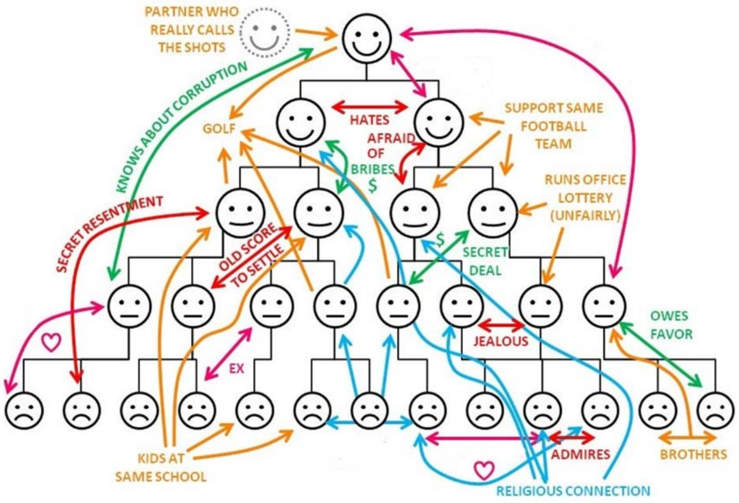

Figure 1: Social Dynamics (Source: Mark Walsh)

Figure 1 is fan fun graphic that illustrates social dynamics. While fictitious, I'm sure every one of can relate someone to something depicted. The source of Figure 1 comes from Mark Walsh from Integration Training, and can be found in [4].

Since my first job as a SAS programming, I have witnessed social dynamics at various levels. Early in my career I observed the usual intra-office romances and sports betting in the kitchen and at the watercooler. Over time, of course my interactions with colleagues changed, as has my view of social dynamics.

In your current work setting, do you see anything that might fit into Figure 1? It might not be as dramatic, but it the dynamics likely exist. Understanding some of the dynamics are important information in decision making. For a particular task, it may be ok to form a team of programmers that are sports rivals but there are several scenarios depicted in Figure 1 that should probably be avoided. We often say that we need to put personal matters aside, and that it just business, but it may make sense to avoid it if possible.

Communication

With Facebook, Instagram, Linkedin, Skype, Teams, today's technology makes for far more interesting social dynamics that back when we huddled around a watercooler to get the latest on what's happening with our colleagues. Far more communication is done electronically than in person. With that comes many challenges with perception. Have you ever sent a Skype message that you wished you hadn't sent simply because of the wording? Have you ever spent 30 minutes writing a 3 sentence email because you were concerned about how it would be interpreted?

Not all communication is verbal. We also need to understand the power of non-verbal communication. How much communication is non-verbal? I'm not sure if this can ever be accurately quantified, but I am certain it is real. Some estimates are as high as 93%, and include such things as facial expressions, body movements and posture, gestures, eye contact, and tone.

Programmers in my experience tend to be social in small groups, and often perceived as anti-social. There are even cultural differences or communication barriers. Often, many programmers continue to stick to themselves. For me, this is the area where I need to take the risk first, be vulnerable, and practice empathy. This requires going beyond our comfort zones.

I recognize communication can often be a challenge, sometimes on a daily basis. At professional conferences, I typically go out of my way to sit with people I don't know. Some days this is really hard. But in doing so, I am continually reminded the similarities I have with other programmers, and sometimes this results in an immediate connection. It allows me to learn from their experiences as well.

Another way to work on social and communication skills is by volunteering within our companies as well as professional conferences. It not only helps develop social and communication skills, it develops a professional network. It also helps us recognize how social dynamics can work. For example, many SAS programming groups have annual conferences. Most are run by volunteers on a rotating basis. Personal differences can often be subtle, other obvious. We must trust that professionalism will overcome personal differences.

I truly believe the volunteering has been a critical to my personal growth. I made a commitment to myself to somehow volunteer at every conference I attend. It is performing a service to others. I continue to grow both personally and professionally from being involved with the professional communities, especially SGF community. Through volunteering, I have developed mentoring relationships that undoubtedly will last throughout my career. For that, I am extremely grateful.

THE REMOTE SETTING

Many programmers are working in a remote setting. The good news is that all the concepts of leadership and acumen still apply. In the remote setting positional leadership (influence via authority) may be more common, but that not the focus of this discussion. As a remote worker, you too can get a seat at the table, though it may be through video.

In my own experiences of being remote for 10 years, I have learned that you can have similar impact as your in-office colleagues, but the approach may differ tremendously. For example, I have learned that it is much easier to first develop connections and influence with a few key in-office individuals, and they in turn had influence over the masses. This may be true for your in-office colleagues as well, but it is extremely challenging for the masses to see your own vision when you are 1000 miles away.

Since I've been a remote worker, most of my "social" interactions with my colleagues has been through involvement in periodic office visits and professional conferences such as PharmaSUG, SAS Global Forum, the Joint Statistical Meetings, and the ASA Biopharmaceutical Section Regulatory-Industry Statistics Workshop. As I reflect upon my experiences over the years, it is clear that the principles of influence and acumen apply to each of these settings as well. Leadership and acumen apply to many settings outside of the office.

This has enabled me to stay in tune with industry trends and emerging technology. As industry trends and technology change, so must we. I have read and strongly believe that improvement is impossible without change. I have also read that change is the price of progress. But to have an impact, we must get buy-in from our colleagues as well as others at the decision-making table (Law 14 in [1]). Getting buy-in can be challenging when you are a remote worker.

When a department historically relies on in-house expertise, it is easy to develop a resistance when any change is suggested by someone external, even if external is a remote worker. Change must be intentional and takes time, yet we are all too busy. Resistance to change can be magnified when it is suggested or imposed by someone from afar. This may be true even when the intent is to benefit to the team.

Based on my own experiences, I have learned to ask a few questions. What does benefit mean? Does everyone understand why? Maybe they don't fully understand the motivation, especially if it disrupts their established work-flow, or even increases it for a short while. Distance is an inherent barrier to the concept of the team, but we can overcome this by applying the concepts of influence and acumen.

In a remote setting, we need to consciously make efforts to be more present. As mentioned above, we have so many tools for communication. In my experience, video conferencing is underused. It is a personal goal to use this more in 2020. Many of us are shy in front of the camera. We need to find a way to become more comfortable. Video is a reminder that we are all people, and it increases our ability to communicate substantially through non-verbal ques. We have challenges, personal and professional, both of which need to be considered when it comes to influence. Emphasis may be on professional challenges at work, but there will be times when outside influences or events may impact our work. This often comes in the form of non-verbal communication.

Without video, it is extremely hard to gauge how a conversation is truly going. For example, there is usually discussion in department meetings. In times of disagreement, there may be a pause. When in the meeting room, it is easy to interpret opinions or reactions simply by observing general behavior/non-verbal actions. Over the phone, this is impossible.

At other times, you've undoubtedly been on a call with someone that appears to be listening yet you have a feeling that you aren't being heard. Coincidental pauses and even tapping on the keyboard are forms of non-verbal communication, even when not done intentionally. I must admit that I multi-tasked from time to time, but I also am more aware of when individualized attention is required.

REFLECTION

Now what? You may have read some papers, attended a seminar, but there seems to be a barrier in taking the next steps. Is it a formal course on leadership? Is it setting up an internal working group to share what you've learned so that others can glean and grow? As a student of leadership, I constantly ask myself these types of questions.

On step that you can take immediately is learning through reflection. After any leadership experience, both good and bad, take time to reflect. Ask yourself a few questions, such as:

- What did I see, hear, and do?
- What do I think other's saw, heard, and did? (use empathy)
- What can be learned from this experience, and are there other areas that it can be applied? Did you have similar experiences in the past, and what was the outcome?
- Could I have done anything differently? (the obvious)

Let's recall the experience mentioned earlier regarding the cumulative review of prior medications listings. This serves as a great example of a case where a programmer's intuition and understanding of data helped tremendously, but it took an understanding of the clinical scientists point of view in order to provide something useful. It also serves as an experience to reflect upon in other ways. What did I see, hear, or do? I simply did what I was told, through the proper channels. What did other's see besides the fact that it's hard to put a staple through 100 pages of paper? The end user was growing even more frustrated over time. I could see it in their facial expressions as they walked through some examples. What can be learned? Communication and an understanding of each other's challenges resulted in a positive collaboration. Even today, I often see work performed to specifications, yet the specifications themselves did not sufficiently describe the end product. This is especially true when working with data standards, where "start with the end in mind" has proven to be true.

Through reflection I can also think about ways to create more influence. Could I have tried to setup internal working groups designed to develop more efficient processes? In the above example, the initial listings provided where efficient for me, but far from efficient use of time for the clinical scientists. Could I have asked my superiors to re-consider defining what "the proper channel" is? I understand that we need to manage resources, but it also seems illogical to be so rigid that people are afraid to ask questions.

From time to time, ask yourself what your definition of leadership is? This is likely to change over time, and might be driven by non work-related experiences.

CONCLUSIONS

Remember, leadership happens daily, not in a day[1]. I often remind myself to remember this, practice patience and the understanding that there is always something new to learn. Recognize that gaining leadership skills takes time, and will always continue. As Maxwell states in [9], no true leader has reached a pinnacle and declared there is nothing new to learn.

When Darrin Hardy was Editor in Chief of Success Magazine, he always ended every interview by asking for one or two take-aways that a listener can work on right way. This helps the listener pick a few things they can improve upon without the need to spend hours or days to ponder. Here are a few take-aways that programmers can implement today, tomorrow, and the next day:

1. Actively listen to your colleagues when they are talking. When in meetings, practice patience and listening by being the last to speak. This helps us understand the various points of view. It also helps us use our non-verbal communication skills to read how the discussion is really going.
2. Volunteer at the next opportunity to help your colleagues or at a professional conference. Service is a vital part of leadership. When we serve others, they will have a natural tendency to want to reciprocate.
3. Be vulnerable by admitting when you are wrong and acknowledging when you need help. This is critical in building trusting teams.
4. Practice empathy. Nothing bad will ever happen from doing this.
5. Be a student. You can start today. Podcasts and LinkedIn posts are immediately accessible. Build your library. Find a mentor. Practice daily.

Don't be discouraged or overwhelmed if you don't earn you seat at the table right away. It can take time. Recognize that you don't need to be good at everything to be a leader. You can still have an impact by making small steps every day.

REFERENCES

[1] Maxwell, J. 21 Irrefutable Laws of Leadership, Thomas Nelson Publishing, 1998

[2] ASA Preparing Statisticians for Leadership, American Statistical Association, Course materials

[3] Sinek, S. The Infinite Game, Portfolio/Penguin Publishing, 2020

[4] Shen, C, Mindset and Postures for Transition Design, https://medium.com/@cathrans/mindset-posture-for-designing-for-transitions-dfd1a2a50498

[5] Prasad, Ron, The Difference Between Motivation & Inspiration, LinkedIn Article (https://www.linkedin.com/pulse/20140512234002-23063390-motivation-inspiration/)

[6] Pink, D., Drive: The Surprising Truth About What Motivates Us, Riverhead Hardcover, 2009

[7] Sinek, S. Start With Why: How Great Leaders Inspire Everyone to Take Action, Portfolio/Penguin Publishing, 2011

[8] Brown, B, The Power of Vulnerability, TED talk

[9] Maxwell, J. 15 invaluable Laws of Growth: Live Them and Reach Your Potential, Center Street Publishing, 2014

ACKNOWLEDGMENTS

I would like to thank Gary Sullivan, Fang Chen, William Wang, and for everyone else that has contributed to the Preparing Statisticians for Leadership course offered by the American Statistical Association. I would like to thank my many mentors that have encouraged me to speak up and challenge that status quo.

To Emily Woolley and Amber Randall, for your review of this paper and willingness to always listen, thank you.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at: