*Presented to **A**ssociation of **I**nformation **T**echnology **P**rofessionals (A**ITP**)∞ Kansas City ∞ March 2005*

Current Issues

In Software Development & Economics

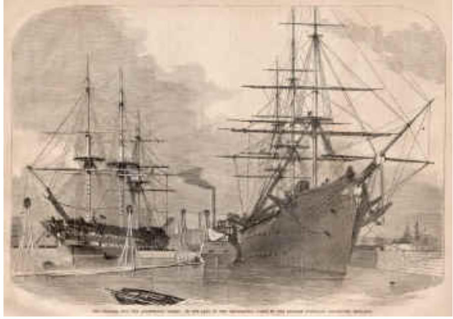
www.softwaremetrics.com

Worst Practices in Information Technology

David Longstreet

We can learn much from studying the mistakes of others. A few years ago I was asked to present a paper on best practices, but I thought it more informative and useful to share some of the worst practices I have witnessed over the past decade of consulting.

The early explorers would travel the globe looking for treasures. While they searched, they would stumble across products such as like tea, pasta, and cocoa, and bring them back to Europe (the old world). I have worked with organizations around the world and in my work I too have stumbled upon ideas and practices which I want to share.



I have consulted and presented papers on every continent on Earth (with the exception of Antarctica). I have advised organizations with IT budgets in excess of a billion dollars and those with only a few employees. I have worked for government agencies, companies that make dog food, organizations that launch rockets and every company in between. The bottom line, I have worked with every type of organization that relies on software.

There is a Johnny Cash song I relate to where he sings, "I've been everywhere, man" and he lists a series of towns, countries and places he has been. I have over 2 million frequent flyer miles and this equates to 80 times around the earth. There have been a few times I think I passed myself in an airport.

Introduction

There is something to be said about any organization which has the word professional in their name. The idea of "professionalism" lends itself to certain meaning. I think the most common accepted definition would be an organization (or person) characterized by or conforming to a technical or ethical standard of a profession. In this case the profession is information technology. Here we are in a group of IT professionals, but could we actually agree on what those technical and ethical standards are for IT?

This is this old joke and a bad joke, but it makes my point. I think it is a Jack Benny joke. It goes like this, "this guy walks into his doctor's office and says doctor it hurts when I raise my arm and the doctors says, don’t raise your arm.” Too often I am called by clients who are in some sort of pain.

The following is a list of some of the worst practices I have seen. While these are not unique to the Information Technology industry they do seem to be more prevalent in IT than in other industries.

1. Making decisions with no historical data
2. Failure to monitor and report status
3. Creating analysis documentation after coding
4. Excessive and irrational schedule pressures
5. Failure to establish clear acceptance criteria
6. Reduce testing time to make a schedule
7. Treating your clients like they were ignorant/stupid.
8. Outsourcing without clear and concise measurements and acceptance criteria.

So do you recognize any of the worst practices? Items number 7 & 8 are relatively new. I guess I am working my way up to 10 items. This could be like one of those surveys in a teen magazine. If you answered yes to more than 3 of the items then we can safely conclude you are not world class. If you answered yes to all 8 items, then we can conclude your organization is worst in class.

A few weeks ago I received a call from project manager in Canada. The firm she was working for had bid on a project and things were not going very well. They were missing a lot of deadlines, quality was poor and morale low. I was on my way to an assignment in Europe and getting ready to get on an airplane, but since she was in pain I would help her out. I asked her some typical questions like the following:

- How many staff support the existing application?
- How many transactions?
- How many function points (which they did not know)?
- How many Lines of Code (just to give me some orders of magnitude)?
- What is the functional purpose of the application?
- How many years was the previous application in production?

I had some experience with a similar application and I did some quick calculations, some orders of magnitude. I asked her what they bid on the project and she said about 12 million. I responded, “per year?” “No,” she said and added, “total.” Then she asked me what I thought, I told her, “I believe it is in the range of 50 to 80 million dollars.” I inquired how did they come up their estimate and she said, “it was a swag and a 12 million account looked really attractive.” Then she said, “What should I do?” Which I responded, “look for another job.”

This scenario is not unique at all and it is all too common.

Background

Over the past 20 years or so, I have collected both quantitative and qualitative data for IT organizations. I have worked with some of the best organizations in the world and some of the worst organizations.

I have learned the most by working with Venture Capitalists who are investing or considering investing in an IT organization or where one firm is in the process of acquiring another firm. Often a manager's intuition is correct, but investors want much more than intuition. They like, they demand, facts. Perhaps even more important they want correct assessments. As one investor told me, I am not paying you to be wrong.

I work with a lot of IT firms trying to implement a measurement program to improve estimating, productivity or quality. To be perfectly honest, most IT organizations abandon their measurement programs after only a short period of time. Most people abandon self-improvement programs also because change is difficult.

Approach (Quantitative v. Qualitative Research)

Ultimately, I try to combine both quantitative and qualitative research to formulate a holistic picture of an organization. I do not want to digress into a statistics discussion, but it is important to define the difference between quantitative and qualitative research and data. A quantitative approach is gathering hard data, "facts." Gathering data such as hours, number of staff, defects, so on and so forth. Quantitative research or data gathering is the most common type of research. On the other hand qualitative research deals with the inquiry and exploration of social interaction and human behaviors. Qualitative data is gathered by surveys, interviews and observations.

I have learned seldom is it technical issues causing projects to fail. Most of the time projects fail because of people type issues.

No Historical Data

It was George Santayana who said, "Those who do not learn from history are doomed to repeat it." I think the key word here is **doomed** to repeat it. By the way it was Carlos Santana who sang, "You got to change your evil ways…baby" that phrase is appropriate for my presentation also.

There are three good reasons for having historical data. The first is to use it for estimating. The second is for trending purposes. The third is for monitoring and controlling projects.

Learning from History

There are a variety of ways one can learn from history. It can be as simple as making journal entries or as complex as creating statistical models.

The field of psychology tells us our memories are not very good and we tend to remember things the way we want to. There is a provincial motto, which says; *"I remember only that which suits my case at this particular moment."* Journals, diaries and letters are good ways to put together a historical perspective. Historians use these types of artifacts when creating a biography.

You have to be a really smart person to remember specific data, events and ratios in your head. At the very least I would encourage you to keep a journal of your projects.   Of course it is best if you are collecting quantitative information, but qualitative information is useful also.

I worked with an executive who had no IT experience, but was placed in charge of the IT group.   Of course, all the IT projects were late and had many problems. This executive would call in the top IT managers and ask them why. The first thing the IT management would say is, “you really don’t understand IT and the nature of IT is….”. So Mike would say humor me and help me understand the reasons. As the reasons were explained to him he would take out a single piece of paper and would write down the reasons.   Of course, the next project was late and had serious problems, so again he called in the IT managers. This time he pulled the single sheet of paper out of his desk.   As they provided reasons he would check off items. The list was almost exactly the same.   Things did not get any better with the subsequent projects, so Mike called in the IT management and this time he said, “don’t make me get out my piece of paper.”

This same executive asked a profound question which I have repeated over and over again. The question is, *“Why do you work a lot of overtime at the end of a project instead of a little overtime throughout the entire project?”*

The bottom line is you need to be looking for trends or patterns that repeat over and over again. Now, I do this at an industry level and there are many similar patterns between companies.   One interesting trend I have seen is there are a lot of musically trained executives who are successful IT managers. There are two primary reasons for this. The first is musicans are familiar with standards. The second reason musicans understand how to work with prima donna’s.   We all know the world of IT is full of prima donna’s.

When I see unrelated companies committing the same worst practice then I can conclude this is an industry wide practice.

Trends

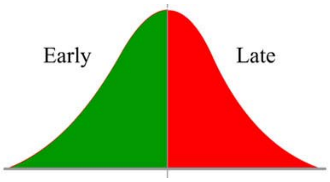
Another reason you want to have historical data is for trending purposes. In other words how do you know if you are getting better or worse? How many IT professionals and organizations can pull out a chart and prove they are getting better?

Estimates Based upon Historical Data

Estimates need to be non-negotiable. I mean it. An estimate should be created using a quantified method. That means there is some method to creating your estimates. You put some inputs into a formula and derive a result. The only thing you should be willing to budge on is the inputs. There are several inputs into an estimate including size (or scope) of the project, the deadlines, the staff, so on and so forth. Hence if the estimate is too high one of the inputs needs to be changed.

Unfortunately, what traditionally happens is an estimate is nothing more than a guess. The estimate has no substance at all. In other words, it is not based upon historical performance or statistical modeling. Often I am working on a contract and I ask the question, "How did you come up with your estimate?" more often than not the person actually admits it was a guess. Another answer is "based upon my *vast experience* as a software professional." In other words, questioning the estimate is the same as questioning their integrity.

One of my favorite movie scenes is in Star Trek III: *The Search for Spock.* Of course the Enterprise has some mechanical problems. Kirk asks Engineer Scott, "Scotty, how long will it take to make the repairs." Scotty replies,"8 weeks Captain, but you don't have 8 weeks, so I will do it in 2 weeks." Kirk says, "Have you always multiplied your repair estimates by a factor of 4." Scotty's response was, "Of course. How do you think I keep my reputation as a miracle worker?"



Often I ask an estimator, what is the probability of being 20% over budget and late? Normally they have an answer. Then I ask what is the probability of being 20% under budget and early? This generally causes confusion. Don't we want an estimate which has the highest possibility or probability of being correct? If an estimate has an equal amount of possibility to be early as late, then this is the highest probability. If on the other hand an estimate has no probability of being early, then it has just about zero percent probability of actually happening and about 100% probability of being late. A good estimate should have an equal probability of being early or late.

I don't want to do it here, but when you derive estimates based upon past historical performance (historical data) you also should derive a margin of error. The margin of error allows you to calculate your confidence intervals (upper and lower boundaries). By the way, the wider the confidence interval the higher the risk of the project.

Let me tell you two stories. I was working with a manager and I watched him interact with his client (it was an internal client). The manager estimated it would take 600 hours to complete a project. His client pushed and challenged him, so the manager reduced the project by 100 hours. This is a 20% reduction in time. Then his client pushes a bit more and gets the estimate reduced another 50 hours to 450 hours. This is a 25% reduction! Whoa! What behavior did the manager exhibit and reward?

I was working at Sprint for Bob who was a retired Marine Corps Colonel. I completed an estimate and when I presented it to Bob, he looked at it told me and said, "I don't believe this and go back and rework the numbers." I went back to my desk and re-checked the numbers and I considered reducing the estimate. I went back to his office and said, "I am pretty sure these number are right." Bob barked back at me, "pretty sure." Feeling a bit insecure I left his office again with my tail between my legs. I sat for a few minutes at my desk thinking. I returned to his office. Again, he said, "Longstreet I told you not to come back until you re-worked those numbers." I stopped him and said, "with all due respect, sir, I have spent a lot of time on this, I have checked and I have doubled checked and I am positive this is right." He looked up, smiled and said, "Good" He paused and said, "If you are not willing to stand up to your boss with your numbers, then why should I be willing to stand up to my boss with your numbers."

When you come up with an estimate based upon some quantitative process, then you are more likely to stand behind that estimate.

By the way, one of the benefits of good estimating is making money and utilizing resources the most efficient way possible. There is an inherent relationship between making money, utilizing resources and waste. There is not much future in losing money. I guess the largest benefit to estimating well is having less waste.

Industry Averages

If you do not have your own internal data it is really pointless to try and do a comparison with industry averages. Let me tell you right here and now, there is no such thing as industry wide IT data.   Your time and money is much better served by determining internal benchmarks instead.

Productivity

Productivity is defined as output divided by resources (or inputs). This formula applies to everything regardless of what you make.   I define productivity for software development as function points / $.   Often I use hours instead of dollars, but the concept is exactly the same.

Productivity = FP / $

Mathematically you can increase productivity by reducing hours or increasing output (function points). In practice, it is not possible to increase productivity by reducing costs. In fact there is empirical evidence to support, for every $1 reduced in cost, you should expect a $1.18 reduction in productivity. Henceforth, there is a negative impact to reducing costs especially by having layoffs. Anyone who has worked for an organization and survived a layoff will tell you a tremendous amount of time was spent worrying, talking and fretting about the layoffs.

Failure to Report and Monitor Status

I am astonished at the number of companies that do not have formal and rigorous project reporting in place. Any project of any significant size needs to have a lot of rigor. There needs to be formal methods of communicating status and monitoring status.

A good project plan needs to be based upon past historical performance. If there are significant variations from past historical performance then there needs to be some explanation associated with it. Variation from historical performance is not necessarily bad it just needs to be explained

.

I do not want to turn this into a statistics class, but you need to be calculating statistics well beyond averages. You need to be calculating things like standard deviation, margins of error and confidence intervals. When I work with a client we spend a lot of time calculating these items. Often the data support people are just making up numbers. You know I can tell with an incredible amount of confidence if you are just making stuff up.

When I work with organizations I examine past project plans and calculate a lot of ratios like the ones mentioned above. Often the results appear to be totally random. The reason for this is every project follows a different process. Basically the data tells me no process is being followed at all. Since the data is all over the place and no process is followed then future performance cannot be predicted. In other words, all estimates in this environment are a waste of time. The very first thing you need to do is stabilize your process.

Creating Analysis Documentation After Coding

If you are doing this then just stop it. Is that the correct way to say that or should I write you need to start doing analysis documentation before you start coding. There are some clear things within the IT world. It is a myth that doing requirements and analysis somehow slow projects down. Those organizations that create solid measurable requirements improve productivity by orders of magnitude. Often they improve productivity by 8 fold.

Management in organizations where analysis documentation is created after coding is derelict in their duties. Management or someone in the organization needs to enforce the practice of doing things in a sequential order.

Let me put it another way. Most of the homes we live cost in the range of $80,000 to $300,000. If we were going to have a home built we would insist the building have a set of blue prints (the analysis document) prior to starting construction. Yet in IT we seem to have a different attitude. If we think about it an $80,000 to $300,000 IT project is pretty small.

Excessive and Irrational Schedules

When there is a deadline I ask what is the motivation for the deadline. How did you come up with that deadline? Now there are some real deadlines and non-negotiable dates, but too many are deadlines based upon a whim or a wish versus a solid business reason. Ultimately this is how work gets prioritized.

The bottom line most organizations can have about anything they want, but they can't have everything they want.

Failure to Establish Clear Acceptance Criteria

How do you know you are done with requirements? How do you know if you did a good job with requirements? When do most of your requirements show up? Requirements are done when you can measure them (or size them). If you can't size them, then you have no idea if requirements are growing and what rate they are growing. If you can't tell if requirements are growing then you cannot determine the impact to schedule and budget either.

There needs to be some standardizations of terminology in all documentation. There is no reason to use different terms to mean the same exact thing. What is a worse practice is using the same term to mean different things.

Reduce Testing Time to Make Schedule

A long time ago when I was managing a testing team I made the comment to my boss, "just because we are not testing a function does not mean it will not work." Not testing is risky business especially when testing was planned from the start. If a project is starting to slip and miss a deadline then this is an indication there are some veryl serious problems.

Treating your Clients Like they are Ignorant/Stupid

There is this great Saturday night live skit about Nick Burns the computer guy. There is no doubt "The Computer Guy" knows a lot of about computers and software, but he is insulting and condescending to his "clients." As soon as the "clients" can get rid of a guy like Nick Burns they do.

I hear folks in IT say things like, "I know more about the business than the business person". Now I am not suggesting it is not true, but it is an indicator of a bigger problem when your technical staff has a higher level of business knowledge than your business staff.

The bottom line is customer service in IT is very poor. It does not matter if it is technical support, requirements gathering, programming, etc. Overall IT needs much work on their people skills.

Several years ago before I learned to be more diplomatic, I was in a meeting with a very senior executive and I thought, well for a lack of better words, he was an idiot. In one of our many meetings he said to me, “Every time I talk I get the impression you think I am an idiot.” I sat quietly and did not respond. Then he said, “what, you have no response.” So I said, “your observant.” Trust me, that one little comment did not help my career.

By the way, there is a difference between ignorance and being an idiot. Once an ignorant person is informed and educated they do not let their ego get in their way. They learn and they correct the problem. On the other hand, an idiot is a person who is not persuaded with facts and figures. They let their egos prevent them from changing direction and adopting new policies. Generally they are very insecure and surround themselves with people with less competent then they are.

Outsourcing without Clear and Concise Measurements and Acceptance Criteria.

If you outsource your software development without first establishing internal benchmarks of productivity and quality, then you are doomed to failure. At the very least you will not know if your “outsourcing vendor” is doing as well, better or worse.

It has been my experience most outsourcing initiatives are not successful. Those initiatives where it has been successful are those where management remains in control and there are comparative metrics in place.

Summary

I wish I could tell you over the years that I am seeing IT organizations getting better and better, but what I am seeing is the contrary. It looks like IT organizations are getting worse. I believe this because projects are getting larger and more complex without best practices in place.

There is an old proverb saying, “seldom does Saul become Paul.” What happens is a new generation grows up using the same ideas as the old generation. It is very hard to change. Over the years I have learned it is seldom-technical issues which inhibit project success. Most of the time it is people issues and practices. There are fundamental behaviors that need to change but changing behaviors is not easy to do.

References

Buehler, R., Griffin D., & Ross M. (1994). Exploring the "planning fallacy": Why people underestimate their task completion times. *Journal of Personality and Social Psychology*, 67, 366-381.

Creswell, John. Qualitative Inquiry and Research Design Choosing Among Five Traditions. Sage Publications. London, 1997.

Gill, John & Johnson, Phil. Research Methods for Managers. Sage Publications, 2002.
Herman C. P. & Polivy J. (2002) If at First You Don't Succeed, False Hopes of Self-Change. *American Psychologist*, Vol. 57, No 9, 677-689.

Kruger, J. & Dunning, D. (1999). Unskilled and unaware of it: How difficulties in recognizing one's own incompetence lead to inflated self-assessments. *Journal of Personality and Social Psychology*, 77 1121-1134.

Lecci, L. Okun, M.A. & Karoly P. (1994). Life regrets and current goals as predictors of psychological adjustments. *Journal of Personality and Social Psychology*, 66, 731-741.