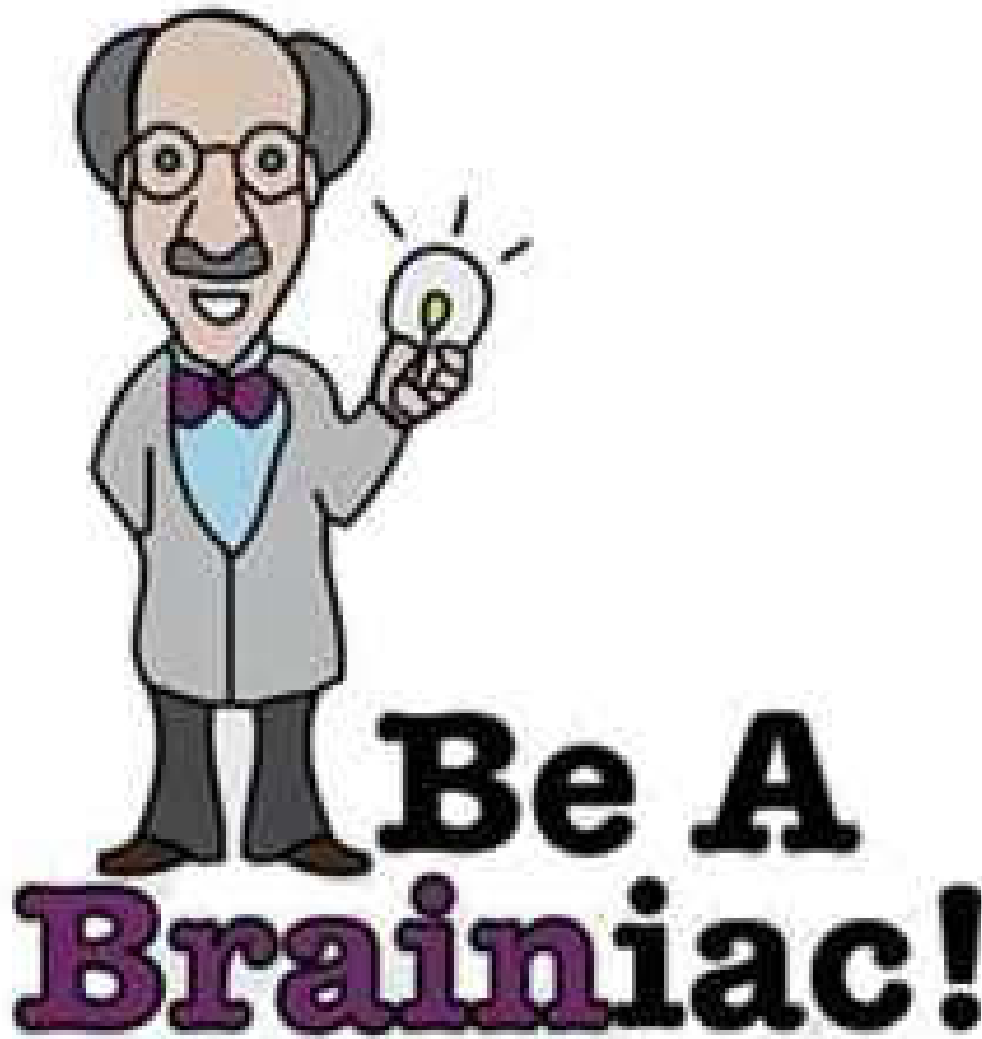—RAKO STUDIOS—

Rako Studios » Media » Management » Scientists and Engineering

Scientists and Engineering

Scientists only have to make it work once.
Engineers have to make it over and over, cheaply.

Many years ago, after a stint contracting at a Silicon Valley Semiconductor equipment manufacturer, I started wondering about the impact of scientists doing engineering work.

My experience, as well as many other consultants, is that non-engineers can do a lot of damage to a project but that scientists can absolutely destroy one.

Scientists are smart

The problem seems to stem from the fact that scientists are usually very smart. This is not a sarcastic comment but an observation of the real effect smart people can have on a project.

The problem with scientists is that because they know they're smart they think they can do anything, including engineering. They can't. When they try the results are often dismal. At that equipment manufacturer the results were observed in a photo-multiplier tube microscope.

The phrase that comes to mind is "half assed". The thing was cobbled together from odds and ends lying around the lab. The programming language for the x-y stage and the microscope was DOS running batch files (I am not making this up). As usual, after some shoddy cargo-cult/Stepford-Wives functionality was imparted to this Frankenstein, *the Management* promptly said: "Ship it!"

The scientists were shuffled of to another project and it became obvious (in their minds anyway) how much smarter they were then anybody else. The scientists knew they were much more smart then those stupid morons in manufacturing.

It seemed that manufacturing just could not build a second instrument. The purchasing agents kept asking where to buy the various parts that were used. The scientists petulantly told them that they found the parts lying around the lab and that the purchasing people should go look at the prototype to see what the parts were.

Unfortunately Marketing had shipped the prototype the week prior. The assemblers kept coming back and asking how to build things that were more like a puzzle then a design. I saw a tall, swarthy, arrogant scientist looking down on an earnest young Vietnamese assembler who was begging for help in how to assemble the product.

The scientist listened to the first few words and then interrupted: "If you're having trouble assembling the product then that's a manufacturing issue." Then he turned on his heel and walked away. It's a good thing you still can't carry guns in California or I would be in San Quentin this very minute.

Never left the carpet

Another engineer who I told this story to said: "Oh yeah, I know the type, he's never been off the carpet and he never will get off it." This was in reference to the fact that in Silicon Valley the R&D cubicles are always nicely carpeted and the manufacturing areas are usually tile.

This clown had no experience in reality and he wasn't going to let any glimpse of reality disrupt his precious overblown self-absorbed ego. He never did get off the carpet. He would design sheet metal parts with a few thousandths tolerance specification over a bend. He would captivate my electrical assemblies by putting a small hole for a wire in a panel as opposed to a slot as I asked.

It takes 3 or 4 extra mouse clicks to do a slot and he was way too important to be wasting mouse clicks just to accommodate a dummy like me.

When the VP asked why manufacturing had to thread a wire through a hole in a panel and then crimp on a connector which captivated the solenoid valve to the panel I replied: "Because James thinks a panel-solenoid sub-assembly is a sensible assembly and service partitioning". (I got my slot.)

Back in the old days, software people were also Scientists. The professors had inculcated in them that algorithms and math were what real scientists did. The further from the hardware and user interface you were the more you're prestige. This is why I had to use a Pal programmable logic chip to do what a 20-cent logic gate could do. Programmable was better even if there were no other inputs or outputs that could conceivable appear to justify the programmable chip.

Scientists like things shoddy and fancy and *100% buzz-word compliant*. Thank god for the young hackers and game designers that are actually able to engineer a software product as opposed to those pompous computer scientists of old.

Fighting the scientists

I had designed an ultraviolet erasing chamber for one of the machines. There was a lot of fun doing the project. They had taken orders a year ago and still didn't have a product. I worked over Christmas and got a prototype running.

This got me labeled as a top-notch technician because everybody knows real scientists never get their hands dirty. The UV lamp was dangerous and needed a key-switch lock out.

The first fun thing was trying to convince the scientists that we needed a switch with gold plated contacts since we were using it as a TTL input to the system logic.

Silver contacts can't switch low-level voltages. They tarnish and need substantial current through them to break through the film and conduct. The scientists knew their periodic tables dammit, and silver is an excellent conductor.

After I used the key-switches laying around the lab (hurry, this is late!) I finally was able to hook a voltmeter to the output and show them that flipping the key-switch did not pass the voltage to the logic circuit. Of course the questioned my setup, the voltmeter and my competence, but after wasting a day or two I was allowed to order the gold-plated contact key-switches I should have ordered in the first place.

This sums up the main foible of most scientists. They think engineering is an unpleasant little stint in the lab. They hate it because they wanted to be theoreticians anyway, not lowly experimentalists.

Engineering hard on the ego

It's devastating on ones ego to do real engineering because things always go wrong. Scientists know this and avoid it. Engineers learn to stiffen their upper lip and slog on. The best engineers realize that making mistakes is the hallmark of a competent engineer. Anyone doing something new makes mistakes. Lots of mistakes. It's part of learning. If the problem is so trivial it can be solved without mistakes then there is probably no value in the solution or somebody else has solved it a long time ago.

Scientists think design is a computer simulation or, at worst, a lab experiment. The job is finished when the slapped-together kludge-ass prototype works (sort of). Then they rush off to other projects to avoid the chaos of trying to replicate their little "experiment". Best of all it shows how stupid other people are because they "just don't get it". "Gosh those manufacturing types are thick," exclaim the scientists.

An engineer is an explorer.

She first surveys the problem and thinks about the potential solutions. The literature is canvassed. The web is searched. Friends are called. Vendors are brought in to make proposals and recommendations. The problem is thought about globally.

Scientists (and bad engineers) dive into some detail, usually in their comfort zone (e.g. batch files) and then complain when marketing changes things. The good engineer is what we are concerned with here. As I said, first comes the exploration, including the customers' attitudes and needs and the areas the product might go in the future. Then the design is intelligently partitioned to allow for modular design, manufacturing and service.

The greatest practitioners of the craft (art) create open systems. Examples include the pulse telephone, the old Harley Davidson, the IBM PC, and the old VW Beetle.

Documentation

An engineer knows that the real goal is to make a pile of paper that instructs others how to make the product and how to change the product and most neglected of all, what products are already out there and how those products were built or changed after shipping. The existence of a prototype is just a verification of the accuracy and completeness of the documentation.

This comes to another profound lacking of scientists as engineers. The complete lack of appreciation of configuration management. They are like the Italian driver in the movie "Cannonball Run", the one who ripped the rear view mirror out of the car an exclaimed "Whatsa behind you does not matter!" That may be true in outlaw road races but in engineering this is one of the most important and neglected areas of design.

A semiconductor machine story

When I worked at a different Silicon Valley company, they had been making semiconductor capital machinery for many years. One of the machine designs was 5 years old. An engineer in my group took a technician and crawled through the machine to figure out just what harnesses and wires went to what boards and operated what components. It had never been done. After all, scientists know that all you have to do is wire the thing up and give it to a tech and eventually a wire harness vendor to make more.

Until Bill took the time to diagram the system, not just the parts, none of the exalted scientists had bothered with it. Three years later Bill was still receiving calls from field service engineers thanking him for doing this because now they had some small hope of fixing the machines.

But I digress. We were talking about configuration management. See, if your just doing a lab experiment to get something working why bother to think about the incredibly difficult issues involved in making an open, sustainable and maintainable design?

I heard many stories about a customer ordering an upgrade on a machine. When the new system was installed something else stopped

working. I would hear folklore engineering like "Oh yeah, when you use the rev D chamber valve then you've got to change the comm board to a rev B or the valve won't work."

Until Bill and a tech crawled around a machine for a month no one knew that there was a signal needed by the valve that ran through the comm board rev B but not any before or after. No one knew for sure why.

We suspect it simply because the connector on the board had a few spare pins and that the scientist that designed it was sure he was really really clever to use the communications board for valve control signals even though they had nothing to do with each other.

Configuration management
Configuration management is knowing what you built, knowing why you built it that way and by implication knowing what you are building right now and how to provide for what you are going to build in the future. Too bad the scientists ran the show at that company back then.

I worked for an experienced engineer and a great one at that. Fred would shake his head at these problems. He told me once "You know, we make very expensive capital machinery in relatively low volumes. I used to work for a company that did the exact same thing only they could tell you exactly what and how they built something 20 years ago, and not only that, they could tell you every maintenance and modification performed on the machine since it was in the field."

I was amazed. "Really Fred? What company? What company did this? Was it in Silicon Valley?" Fred scoffed--no--not here--- it was in Seattle.

The name of the company was Boeing Aircraft. I smiled. "I see your point Fred." Fred said we could go to San Jose airport, pick out any Boeing aircraft, and look at the number plate in the door jam. With that number you could determine when it was built, the bill of materials and revision level of every part used.

You also could find the records of every maintenance and change to the aircraft. He said Boeing developed this stuff 40 years ago--now it was all computerized and available for use by companies wishing to use it for a fee.

The "A" and "B" teams
Unfortunately the scientists at this company did not think it was important. I suspect things have changed since. Now don't get me wrong, I like scientists. My brother has a Doctorate in non-linear optics. He worked at the old Bell Labs and then AT&T for a while and had some interesting observations.

He pointed out there was an "A" team and a "B" team.. He had experience with both. He got on the Bell Labs "A" teams due to his specialty. He got on the AT&T "B" team because his doctorate was not earned at an Ivy League school but at a land-grant college in Ohio. He said he was astonished at the difference in the teams.

The "A" team couldn't accomplish anything. Everyone was way too concerned with proving they were smarter then everyone else in the room and elevating their rank in some misty pecking order. The primary code word was "bright" (as in "He's very bright" i.e. "He's one of us"), as opposed to "She's a good scientist" (but certainly not one of us).

Exclusion was based primarily on Alma Mater but gender, ethnicity, and God know what else might get you on the "B" team. When my brother got on one of these he couldn't believe he was in the same company. Instead of the pompous grandiosity of people trying to prove they were smarter than every one else (including the boss) the "B:" team actually got things done.

It really was a team and John saw it in the way they would work. A fiber guy would say he needed more power to overcome a loss in the splices. Instead of a dismissive, "You should learn how to splice better" the amplifier guy would say, "Well I could give you more power but the die would have to grow 5 mils. That would interfere with the case." Then the case guy would say,: "I think I can give you 5 mils, I'll use a smaller diameter end mill to route the pocket." Then the fiber guy would say, "Yeah but won't that up your cost and scrap rate?" And the case guy would say, "Yeah, a little, but let me talk to the machine shop and see if they can do it. If it's important to get the power up I want to help any way I can."

Then everybody would say, "Great/ hooray, right on!" and begin talking about their kids and the weekend and baseball game. Whole complete functioning people solving problems together. What joy.

The team is the thing

The Scientists? My brother postulates that they were cooped up in the library their whole lives and unlike us rural Ohio boys never played on a sports team in High School. Without this team formation and maintenance experience the "A" team "bright" kids were domed to a life of prestige-seeking instead of enjoyable accomplishment.

Of course I see nothing wrong with prestige seeking, I hope to get a little from this article. But when they plant you 6 feet under all the prestige in the world won't bring you back. Better to live a life of enjoyable team-spirited problem solving. You'll die happier and the world will be a better place due to your accomplishments. Revision 4

FAST lab, photo credit Anna Peacock