Check the Simple Things First

Alan Nicol, Executive Member, AlanNicolSolutions

Whether it’s equipment maintenance and repair, troubleshooting a process, or fixing business cultural problems, before you gear up for a complete overhaul, check on the simple fundamentals first.Often times the easily overlooked issue is the root cause.

Whether it is for my own web site or the posts I write for others, the lessons and observations that I prize the most have not, for the most part, come from executive leaders or highly paid consultants, nor have they often come from books written by expert celebrities.Most often, they come from taking to heart what my peers and teammates have taught me and learning from their examples and lessons.

The advice I have to share today comes from my father.He has given it to me before, but like most sons, I’m afraid that I often easily forget my father’s wisdom and must be reminded.Most recently, his advice concerned automotive repair, but like most good advice, it is universal.

He and I were chatting on the phone and I was explaining that I dreaded an automotive repair chore that was necessary.My pickup had adopted a disturbing engine noise that under certain conditions became severe and accompanied a momentary degradation in performance.I had eliminated all of the obvious and relatively easy things to fix, and, with advice from some experts, determined that the timing chain was rattling.Not good.

To have a professional service fix it would cost about $750.It’s amazing how when budgets are tight our own time becomes cheap.I had decided that I would make the repair myself, even though it meant removing the top and front workings of the engine, a substantial chore.

My father said, "Before you do that, look for the simple and easily overlooked problem first." He told me a story about how he and a mechanic did a complete valve replacement on his 1964 Chevy and when they put it all back together, it didn’t run correctly.They tore it all apart again and looked for trouble, and couldn’t find it.They put everything back together and it still didn’t run correctly.

Then, they pulled in a second mechanic who, not caught up in the paradigm of a valve replacement, immediately started with the basics of fuel and spark and found that one of the new spark plugs that went in with the first valve replacement was defective.One new spark plug later, the engine ran perfectly.

Taking my father’s story to heart, I decided to check the oil in my truck.The tension guides for the timing chain work on oil pressure.My oil level was very low.I changed the oil and filled it to the proper level and the problem ceased.Thanks, dad!

I still need to figure out the mystery of where my oil went, which I think I have done, but I hope you can see from the example, the value of my father’s lesson. I could have spent $750 or several days and busted knuckles tearing apart the front end of my pick-up, but by following his advice and checking for the easily overlooked problem, I solved it with a $40, 1-hour, flush and refill service.

The same phenomenon and lesson applies to product development and process improvement.Our methods and tools for process improvement are often taught to us in the context of designing or rebuilding a process entirely.It’s a good context to teach a wide range of tools and perspective.But, it’s easy to forget that sometimes a small thing can cause big problems.

Similarly, in product development, we find ourselves with a prototype or some test results that tell us we haven’t got the solution we need.The temptation is to go back to a different concept or to redesign from scratch, but the real problem could be something really simple.

Here are some more brief examples that I can recall to provide some insight into how the simplest things can be the cause of huge wastes of time and money.Perhaps they will help to demonstrate the universal application of my father’s advice.

While developing a small mechanical device product, our team encountered a complete test failure.Every sample that went through the reliability and performance test failed.Our directive was to go back and redesign the greater portion of the mechanism.The spatial constraints were difficult and altering the design to increase strength and tolerances seemed impossible.It was a frustrating problem.

Before getting carried away with a redesign, we started asking questions.The first thing to cast doubt was that our models and analysis showed the design should have had a margin of safety and shouldn’t have failed.We re-ran the models and couldn’t find a problem.Even expanding worst-case assumptions didn’t predict such failure.

Then someone asked what the quality report on the components from the supplier said.We couldn’t find one.It wasn’t done.Somehow, the parts were rushed into the test prototype before being inspected.A post-mortem on the mangled components showed that they were built out of specification.

We explained the situation to the supplier, ordered some more parts, verified them this time, built some more prototypes, and re-ran the test.All the samples passed.We had to run several samples well past the performance limits just to induce some failures and have a prediction of true reliability.The design was good after all.

The lessons of this story are several.First, don’t go into testing with bad parts.Verify them first and save the very expensive iteration.Second, if we had not second-guessed the cause of the failure, and had just gone back to the drawing board we would have spent months redesigning something that was designed well already.

In another example, a colleague of mine facilitated a lean re-design of a manufacturing process.The new process enabled a SMED inspired quick change of tooling, so the production line could be used for several different parts and changed from one to another in just a few minutes.

The early test runs showed production levels substantially lower than expected, and lower than the original processes could accomplish.The immediate reaction was to revert the production line to its original layout and remove the changes.He stayed the course and systematically went through the process, observing and measuring each step.

He found that some of the assembly steps were taking too long.Nothing his team changed in the process should have altered the assembly.The problem was that the parts were not fitting together correctly.He traced it back to one of the new quick-change manufacturing tools.It was built out of specification.They changed the tool to a backup that proved to be correctly made and the process worked very nearly as well as his models predicted.

Again, a simple thing that is easily overlooked or assumed to be correct turned out to be the true culprit of a process performance problem.Because my colleague took a couple of hours to hunt it out, he proved the new process to be a genuine improvement, and he prevented two days of re-arranging equipment to put things back they way they were.

In my last example, a team was tasked with the difficult and disenchanting task of devising a single Engineering Change Order (ECO) process that would be used by nine different engineering design centers inside of a corporation.The corporation had adopted an Enterprise Resource Planning (ERP) system and to take advantage of its power, needed to standardize many business processes.Heretofore, the nine different design centers each did ECOs their own way.

Without going into the process of consolidating everyone, we’ll jump to the

deployment.The new process worked well enough once everyone got the hang of it, and seemed to be working in every location, until the collaboration started.Suddenly, the benefit of the ERP system wasn't working; the ECOs from one design center were causing trouble in another.

Immediately, everyone who had already "villainized" the ERP system and everyone associated with it for all of its forced changes cried fowl.Even champions of the system were disappointed and doubted the utility.Fortunately, the leaders of the team that designed the common process were able to quickly deduce the problem.

Neither design center was utilizing the process in the system exactly as it was intended.Their distinct adaptations of the process map and instructions caused a significant mutual conflict in which ECOs were simply getting stuck and failing to move on.The problem was corrected with some clarity in the instructions and some adjustments to how each center understood the process.

Fundamentally, in this example, a simple communication error or misunderstanding drove a meltdown of process that also fed off of enmity for change and rivalry between business centers.The team was able to correct things just in time, before cultural challenges combined with a process malfunction nearly sabotaged the ERP initiative or otherwise tore a long-lasting gulf in cooperation among design centers and corporate functions.

I'm hoping that at least one of these examples fits a context close to home for you, the reader.I tried to recall one where the team did not find the simple fix and went ahead with the nightmarish plan, but I couldn't think of one.I suspect that the reason is that in some cases where the fix to a problem was a monstrous endeavor, we never did identify a simple fix.There might have been one we overlooked, but since we didn't find it we never knew.

These opportunities for a simple thing to inspire enormous effort happen frequently.They often occur when we make a major change.We forget to warn or alter an input to keep up with the new process.We forget a part, or a part isn't correct, or isn't installed correctly.Something is out of calibration.We change the rules and forget to tell someone.

Every major change is an opportunity for many small errors.Check for the small errors first; the ones we assumed couldn't or wouldn't go wrong, or that someone already checked before jumping to conclusions that the new system is broken.

Whether the context is indeed a recent change or something that has been humming along just fine until now, before you contemplate a major overhaul, look for the simple and overlooked thing that could be part of or the root of the problem.Finding and fixing the simple things can save you enormous amounts of time, money, and frustration.

Stay wise, friends.

If you like what you just read, find more of Alan's thoughts at www.bizwizwithin.com

[1]

Source URL (retrieved on *11/21/2014 - 7:18pm*):
http://www.foodmanufacturing.com/articles/2013/02/check-simple-things-first

Links:
[1] http://www.bizwizwithin.com/