The e-newsletter of Lean Manufacturing
LEAN DIRECTIONS

The Essence of Jidoka

The Toyota Production System is frequently modeled as a house with two pillars. One pillar represents just-in-time (JIT), and the other pillar the concept of jidoka. The house will not stand without both pillars. Yet many of us focus on the mechanisms of implementation--one piece flow, pull production, takt time, standard work, kanban--without linking those mechanisms back to the pillars that hold up the entire system. JIT is fairly well understood, but I believe jidoka is key to making the entire system stick. A lot of failed implementations can be traced back to not building this second pillar.

What does jidoka mean? A common answer to this question is "autonomation" or "automation with a human touch." This is usually illustrated by example of a machine that will detect a problem and stop production automatically rather than continue to run and produce bad output.

The principle's origin goes back to 1902 when Sakichi Toyoda invented a simple but ingenious mechanism that detected a broken thread and shut off an automatic loom. That invention allowed one operator to oversee the operation of up to a dozen looms while maintaining perfect quality. But the system goes much further.

The jidoka pillar is often labeled "stop and respond to every abnormality." This is obviously much more than having a machine shut down. Toyota refers to every process, whether human or automatic, being enabled or empowered to autonomously detect abnormal conditions and stop. The team member pulling an andon cord on the assembly line is jidoka as much as an automatic machine.

At my company, we define jidoka as a four-step process that engages when abnormalities occur.

1. Detect the abnormality.
2. Stop.
3. Fix or correct the immediate condition.
4. Investigate the root cause and install a countermeasure.

The first two steps can be mechanized or automated. Poka-yoke devices are one method to allow a process to detect a problem and stop. But if you take a broader view, every one of the mechanisms of the Toyota production system is really designed to do these two things.

Time itself can be a powerful detection mechanism if work cycles are paced to the takt time. What should be complete 25% into the takt time? How about 50%? Is the work progressing the way you expect it to? There is a huge opportunity to detect a problem when there is enough time to respond--instead of discovering at the end of the day (or week) that things are way behind. If the team member is given the means to immediately signal that she has encountered a problem (via andon, for example), then the response can be immediate.

Kanban also serves as a system to detect abnormalities. If there is inventory without a kanban attached, I know that either overproduction has occurred or somebody didn't follow the rules. If the system has been running smoothly and there is a shortage (or overage), I know something has changed.

All of the mechanisms of lean manufacturing follow the same pattern. They are designed to operate with

the bare minimum (just enough, just in time) in order to detect abnormal conditions or system changes that might otherwise go unnoticed.

Detecting an abnormal condition does no good, though, unless there is follow-up. Visual controls are just decoration unless they trigger action. The second step is to stop. A lot of people have a hard time with this because they think it means bringing all production to a grinding halt until the problem is resolved. Don't get me wrong. Depending on the nature of the problem, sometimes it does. But stop is frequently simply a mental shift. It means "stop doing what you were doing because you need to do something different." It is an acknowledgement that some kind of intervention is required. That might mean shutting down a process or machine, or it might mean signaling for assistance.

What I would not want to happen is to expect the team member who discovered the problem to try to fix it without telling anyone. But many times we expect them to do just that, and the rework becomes so deeply embedded into the routine that we can't even tell it is happening. It seems normal because it has never been flagged as abnormal.

The third and fourth steps cannot be automated. They are entirely the domain of people because they require diagnosis, analysis and problem solving. How well an organization can get through the steps involving fixing the problem and installing a countermeasure ultimately decides whether they move forward into continuous improvement or slip back when JIT reveals a problem to them.

Step three is to fix or correct the abnormal condition--to get the train back on the tracks so production can resume. It might entail adding a temporary countermeasure to avoid recurrence of the problem. It might require some fire fighting. It might require running an exception process such as putting in some temporary kanbans or putting a unit into a rework stall. It might mean shutting down production until a broken tool is fixed.

These decisions need to be made at the lowest possible level of the organization, but no lower. As the scope of the issue and its potential customer and economic impact increase, so must the level of action required. If the solution requires violating the principles of the production system, it should not be taken lightly. There are plenty of ways to solve problems while maintaining system integrity. Every time you bust the system to get something done, you erode confidence just a little bit.

The last of the four steps is to investigate the root cause of the problem and install a permanent countermeasure. This is an opportunity to expand your knowledge of your processes and production system. Depending on the nature of the problem, it could be a straightforward solution. Of course it might also require enlisting help of a six-sigma black belt. Either way your system is improved. You will certainly have to prioritize your efforts, but do not just stick chronic problems into the too-hard box.

JIT is powerful because it drives out unnecessary cost and helps detect problems that cause waste. Jidoka is the response to problems. JIT is relatively easy to implement, but without the mechanisms of jidoka in place to support it, JIT quickly erodes, and the waste finds its way back in. When JIT and jidoka work together, they form the engine of kaizen that drives your system to get better every day.