"Core Process Pull: Little's Law in Action" - Webcast Addendum

Lars Maaseidvaag
Senior Master Black Belt
Email: Lars@MoreSteam.com

***Q:** You mentioned transactional and manufacturing. What about applying the pull system to product development, where design of features and functions are the WIP?*

A: This may be the very best application area for Core Process Pull. Perhaps the two biggest factors in making product development processes unstable are the high task variation and status updates.

High task variation comes from the fact that the development of one feature or product may take much longer than the development of another feature. In a manufacturing environment, or a repetitive transactional environment, the amount of task variation is relatively low. In development work, this task variation can be extremely high.

The second big factor that consumes time is providing updates on the projects that are being worked on. Because of the high task variation, management typically likes to know how things are progressing, thus driving the need for frequent, detailed updates, which consumes time, which then lengthens the project lead time and often leads to more task variation. It becomes a vicious cycle.

If you have more projects (the WIP) that are in process, they will all be sharing the development capacity, and therefore getting done more slowly. Because they are being done more slowly, your management will want more status updates. The status updates consume more capacity, which makes the projects take longer, which means more updates are required… you get the picture – **it becomes a death spiral** of high variability, long lead times, and endless reports about high variability and long lead times.

Core Process Pull provides a simple method to control this downward spiral. A simple **cap on the number of projects** allowed in the development process will allow those few projects to be completed in a timely manner, without an excess number of status update meetings. When a project is completed, the next, highest priority project is begun.

A very important benefit of Core Process Pull in this situation is the **prioritization of the queue of waiting projects.** What is started next may not have been in the queue the day before, but it became a high priority for whatever reason, and therefore it will start next. No low priority projects enter the process only to languish and take up capacity.

If our development process is running well, we may never have to interrupt a single project to work on a 'burning' issue, because we can complete the high priority projects so quickly that they never become 'burning' issues.

If you think about how LSS Green Belt and Black Belt projects are assigned, you will see a Core Process Pull system in action. Projects per person are limited to two or three, and you do not start a new project until you complete one of your existing projects. When you are ready for a new project, the highest priority project is begun, not the project that has been waiting the longest.

Q: *For a make-to-stock business, how do you calculate safety stock minimum quantities that then trigger a kanban card to make more?*

A: Safety stock and make-to-stock environments use Replenishment Pull systems, rather than the Core Process Pull system. All Replenishment Pull systems use three basic inputs to calculate the various parameters of the system. These three inputs are: vendor lead time to receive a new order to replenish stock, the cycle time of your on-hand inventory (how often do you want to re-order), and the safety stock.

Of these three, the first two are simple, straightforward numbers. However, safety stock calculations can become very complex quite quickly. The basic safety stock formula given in many industrial engineer texts is:

safety stock = (service level (usually 2)) x (standard deviation of demand variation)
x (the square root of the lead time for replenishment)

While the simplicity of this formula is appealing, it relies on several statistical assumptions that do not always hold true, and it leaves out other data that is often available and can help us make a better safety stock decision.

In practice, safety stock levels should be set by a blended collection of results from several calculations and factors unique to each pull system implementation. And, in every case, conservative starting values for safety stock should set as the pull system is first implemented. As the system stabilizes, you will be able to see how often the system is forced to use the safety stock, and from those observations you will be able to fine tune the pull system to minimize the amount of inventory in the process.

***Q:** Have you applied Core Process Pull to a continuous manufacturing closed system that produces multiple SKUs over 100, where packing constraints create a need to have a buffer inventory?*

A: Without having a chance to ask for more information, it appears that this question is actually asking about the sizing of Replenishment Pull systems at the end of a continuous manufacturing line. In this case, it sounds like packaging constraints are forcing the main manufacturing line to produce in a batch mode, or the packaging line needs to produce in batch mode.

Because of this, buffers of WIP need to be maintained to disconnect or decouple to the two parts of the process – the continuous manufacturing and the packaging. **These buffers act more like Replenishment Pull systems than Core Process Pull systems.**

Other Lean tools that would be interesting to investigate in this situation include setup reduction and proper batch sizing to make sure the process is producing in the smallest, leanest possible batch sizes.