

PM WORLD TODAY – PM TIPS & TECHNIQUES – NOVEMBER 2008

Applying Little's Law to Agile Project Management

*Tathagat Varma, PMP®***Abstract**

Little's Law¹ states that inventory in a process is the multiplication of throughput and the flow-time². While this seems intuitive, it helps us establish a mathematical relationship between basic factors that govern performance of a production process: arrival rate (or the flow-time or the cycle time), manufacturing lead time (or the throughput) and the inventory present in the system at any point of time (or the work in progress). The law has been found to hold good as long as these three parameters represent long-term averages of a stable system and they are measured in consistent units³.

Little's Law has its origins in manufacturing, but it also very relevant to a project manager in non-manufacturing setup. An in-depth understanding of Little's Law will help a project manager to improvise on critical ideas in scheduling project activities to minimize the risk of schedule variance, improve accuracy of tracking and provide more usable status reporting.

In this first paper of the two-part series, we will explain what is Little's Law using an example, and conclude with what it means for manufacturing and for software development. A second part of this paper will explore in details how Little's Law is not only conceptually akin to the Agile way of managing project, how well its mathematical principles could be used to improve project management using Agile philosophy.

Keywords

Queuing Theory, Little's Law, Manufacturing Lead Time (MLT), Cycle time (CT), Inventory, Work In Progress (WIP)

Little's Law

Little Law is a very important result in Queuing Theory which establishes a mathematical relationship between the average number of customers ('N') in a steady-state system, their average arrival rate (' λ ') and the average time they spend in the system ('T') as follow:

¹ http://en.wikipedia.org/wiki/Little's_law

² <http://www.factoryphysics.com/Principle/LittlesLaw.htm>

³ <http://www.factoryphysics.com/Principle/LittlesLaw.htm>

$$N = \lambda T$$

The strength or the simplicity of Little’s law lies in the fact that it makes no assumption about the probability distribution of arrival rate or service rate, or if they make a FIFO (First In First Out) queue, or a LIFO (Last In First Out) queue, or some other order in which they are served. The only pre-condition or requirement for Little’s Law to hold is that it must be applied to a stable or a steady-state system, which means a system that is in a transition state may not behave as per this law. A transition state could be the starting up or a system, or shutting down, etc.

How it works ?

To illustrate this point, let us take the example of a scotch manufacturer who wants to sell 10 year old scotch.

If his annual production is 1,000 cases, his inventory will look something like this:

Year	Cumulative Inventory	Stock available for sale
At the end of Year 1	1000	0
At the end of Year 2	2000	0
At the end of Year 3	3000	0
At the end of Year 4	4000	0
At the end of Year 5	5000	0
At the end of Year 6	6000	0
At the end of Year 7	7000	0
At the end of Year 8	8000	0
At the end of Year 9	9000	0
At the end of Year 10	10000	0

At the end of every year, the inventory goes up by the amount of annual production in that year, but since the scotch that can be sold has to be at least 10 year old, the stock available for sale continues to be zero for the first ten years.

Let us project the same production for another 5 years:

Year	Cumulative Inventory	Stock available for sale
At the end of Year 11	11000	1000
At the end of Year 12	12000	2000
At the end of Year 13	13000	3000
At the end of Year 14	14000	4000

At the end of Year 15	15000	5000
-----------------------	-------	------

We can say that from 11th year, the system has reached steady state. At this point, it takes 1 year to get 1,000 cases of the produce. Suppose from the eleventh year onwards, the stock available each year is sold to customers. If we project the next 5 years, they will look like:

Year	Cumulative Inventory	Stock available for sale	Stock Sold	Cumulative Stock Sold	Sellable Stock Left
At the end of Year 16	16000	6000	1000	1000	5000
At the end of Year 17	17000	6000	1000	2000	5000
At the end of Year 18	18000	6000	1000	3000	5000
At the end of Year 19	19000	6000	1000	4000	5000
At the end of Year 20	20000	6000	1000	5000	5000

From this table, we find that whenever we start selling the produce, as long as we don't sell more than what we produce in a year, it still takes one year to get 1,000 cases of the produce.

Now let us focus on year 18. We want to find what the inventory is at the end of year 18. The cumulative inventory is 18,000 out of which 3,000 has been sold so far and another 5,000 is the stock that has already been manufactured and is waiting to be sold. We can't consider this as the 'inventory' which is actually any state of raw material before the final production.

So, the real inventory in year 18 would be $18,000 - 3,000 - 5,000 = 10,000$. If we do similar computation for all years from 16 until the 20th year, here is what we get:

Year	Cumulative Inventory	Stock available for sale	Stock Sold	Cumulative Stock Sold	Sellable Stock Left	Current Inventory
At the end of Year 16	16000	6000	1000	1000	5000	$16,000 - 1,000 - 5,000 = 10,000$
At the end of Year 17	17000	6000	1000	2000	5000	$17,000 - 2,000 - 5,000 = 10,000$
At the end of Year 18	18000	6000	1000	3000	5000	$18,000 - 3,000 - 5,000 = 10,000$
At the end of Year 19	19000	6000	1000	4000	5000	$19,000 - 4,000 - 5,000 = 10,000$
At the end of Year 20	20000	6000	1000	5000	5000	$20,000 - 5,000 - 5,000 = 10,000$

In all these cases, the inventory in any year remains stable at 10,000. At a very high-level, we can apply the logic in many different ways as:

- at any time, the inventory is 10,000 cases. It takes 10 years to make the 10-year old scotch, hence the maximum produce in a year can only be 1,000.
- We know that the company produces 10-year old scotch and can make only 1,000 cases in a year. So, at any time, they must have $10 * 1,000 = 10,000$ cases in stock in various stages (1,000 cases that are 1 year old, another 1,000 cases that are 2 years old, and so on).
- We could also depict the relationship in another way. At any time, we have 10,000 cases and the maximum produce we can have is 1,000 in one year. If I must only sell 10-year old scotch, then the lead time for any of the cases, irrespective of when they started their journey, is 10 years.

All these empirical observations lead us to a mathematical relationship between these three numbers: 10 years, 1,000 cases/year and 10,000 cases of inventory at any point in time.

- 10 years represent the time it takes for a raw material to complete its journey and become a finished product. Let us call it Manufacturing Lead Time (MLT)
- 10,000 cases of inventory at any point in time represents the raw material that must be always maintained in the system to achieve production targets and production goals. Let us call it Work in Progress (WIP).
- It takes 1 year to 'catch' 1,000 cases. Suppose I have the 75th case in my hand. To 'catch' the 76th case, I will have to wait 1/1,000 of a year. Let us call this Cycle Time (CT)⁴.

We can now re-state the equation as:

$$MLT = WIP * CT$$

In this case, the equation will be :

$$10 \text{ years} = 10,000 \text{ cases} * 1/1,000 \text{ year per case}$$

In this example, we took a simple M/m/1 queue. However, Little's law has been tested and found correct on multi-product, multi-queue production systems. The only constraint is that we must apply it when the system is in steady-state. In this example, if we were to apply the Little's law in the years 1 to 9, we would get erroneous results because the system is still under transition (the production process has not completed its first cycle yet).

Implications of Little's Law

Based on the simple illustration above, we can intuitively conclude that larger the manufacturing lead time, the more will be inventory in-process at any time. It is possible that manufacturing lead time comes across as a non-negotiable constraint in certain industries,

⁴ In some literature, the term 'throughput rate' is also used in place of the inverse of cycle time. Note the difference – one is a 'rate' and the other is a 'time'.

but in general, if it could be reduced, we could reduce the work in process. A lesser work in-progress requires lesser capital expenses and other operational costs. In our example, if we decide to lower the ageing criteria to only 3 years, we would be able reduce the inventory from 10,000 to only 3,000. This will result in a lower amount of capital being locked.

Traditional manufacturing relies heavily on 'capacity utilization'. This forces pile-up of inventory so that there is no downtime for any machinery due to lack of raw material. However, modern approaches such as Lean Manufacturing advocate total elimination of inventory. Inventory can be thought of as the efficiency of the production process.

In the context of software development, manufacturing lead time could be construed as the time when team starts working on the requirements to the team when product is delivered. In a traditional model (like the waterfall model), a customer might not get to see the end-product until after the team has implemented the entire project scope and has completed all aspects of design, development and qualification. This creates two major challenges:

1. For the entire duration of project, the work is always in-progress, or lying as the inventory. As the software is not yet shippable during the duration of the project, it leads to locked capital without any way of an early cost realization.
2. The team can't deliver often. A larger manufacturing lead time might lead to not more than one or two releases per year, while the customer(s) might expect more frequent releases.

Modern approaches to software development such as Agile (specifically Scrum) solve the same problem by breaking both the inventory and the manufacturing lead-time into smaller and more manageable pieces.

About the Author:***Tathagat Varma****Author*

Tathagat Varma, PMP®, 6σ Green Belt, Certified Scrum Master, Lean Certified, Kaizen Certified, is General Manager with NetScout's Bangalore Engineering operations. [NetScout Systems](#), Inc. (NASDAQ-NTCT) is the leading provider of integrated network and application performance management solutions for organizations worldwide. Prior to NetScout, he has led large-scale product development efforts with DRDO, Siemens, Philips and Huawei in hi-tech areas such as Digital Video Broadcast, Core Routers, Softswitch and Routing Platform. Apart from large-scale development efforts, he has been part of SEPG, CMM Assessment Teams, Internal Audits, and Process Trainings at various organizations where he has worked. Tathagat has been involved with software product development for the last seventeen years and has been part of the CMM, ISO TickIT and TQM initiatives at his previous employers. He is currently working on understanding and cross-pollinating the principles of Supply-chain Management, Lean and Six Sigma in modern and evolving paradigms in software development like Scrum. He has been formally trained and certified in CMM, PMP, Lean, Scrum, Six Sigma Green Belt and Kaizen. He holds a Masters Degree in Computer Science from JK Institute of Applied Physics and Technology, Allahabad and an advanced post-graduate certificate in HR from XLRI, Jamshedpur and is currently pursuing Certificate in Business Leadership from Cornell University, USA. He is also a reviewer with IEEE Software since 1998. Since Jun 2008, he has started a forum for [Software Architecture](#) and hosts a blog, [Manage Well](#), where writes about his views on how to simplify management in everyday job. He can be contacted at Tathagat.varma@gmail.com.