Kanban Scenario One: Visualize Little's Law

Little's Law statement is: The average number of customers in the store L is the effective arrival rate λ times the average time that a customer spends in the system W. Please note that the long-term arrival rate is equal to long-term departure rate for a stable system used in Little's Law.

$L = \lambda * W$

Eight customers enter a store each hour on average. So, λ = eight customers per hour.

Each customer spends about 30 minutes on an average at the store. So, W = 0.5 hour.

How many customers will be there at the store at any point in time (on average)?

If we apply Little's Law, the number is = eight customers per hour multiplied by 0.5 hours = four customers.

Let's try to visualize this. We will name our customers by letters of the English alphabet. The timelines and number of customers in the store are shown in table A below.

At Hours (HH:MM)	At the Store	Left Store
00:00	A	
00:7.5	BA	
00:15	CBA	
00:22.5	DCBA	
00:30	EDCB	A
00:37.5	FEDC	BA
00:45	GFED	CBA
00:52.5	HGFE	DCBA

01:7.5	IHGF	EDCBA
01:15	JIHG	FEDCBA
01:22.5	KJIH	GFEDCBA
01:30	LKJI	HGFEDCBA
01:37.5	MLKJ	IHGFEDCBA
01:45	NMLK	JIHGFEDCBA
01:52.5	ONML	KJIHGFEDCBA
01:60	PONM	LKJIHGFEDCBA
02:00	QPON	MLKJIHGFEDCBA
02:7.5	RQPO	NMLKJIHGFEDCBA
…	…	…

Table A

As we can see, except for the first few minutes, the number of customers at the store is indeed four. Please note that customers will not enter or leave the store at such a uniform rate. But Little's Law is based on long-term averages. So, this simulation assumes a uniform arrival/departure rate and time spent at the store based on averages. Table A visualizes Little's Law.

Little's Law is remarkably simple, but it conveys critical information about any workflow. This is how it applies to a workflow.

Average number of work in progress items (WIP) = Average cycle time (T) * Average completion rate of work items (R)

In other words, Average cycle time (T) = Work in progress (WIP) / Average completion rate (R)

If you wish to produce items more quickly and reduce average cycle time, you can do one of the following:

a. Reduce WIP

b. Increase average completion rate

There are lower and upper limits to how much WIP we can have in a system, but limit WIP has direct impact on how much time we spend on each work item. This is the far-reaching conclusion from this deceptively simple law. Kanban proposes the idea of limiting WIP.

Kanban Scenario Two: WIP Limits = 1. This scenario A workflow has two steps: validate and implement. The validate step takes 10 hours per person per work item and the implement step takes 20 hours per person per work item. In this example, we set the WIP limit at each step to one, which assumes that one person is assigned to each step.

Table B shows how work items, named after alphabets, flow through the workflow.

At Time (Hours)	Validate (WIP: 1) - 10 hours	Implement (WIP: 1) - 20 hours	Items Processed
0	A		
10	B	A	
20	B	A	
30	C	B	A
40	C	B	A
50	D	C	BA
60	D	C	BA
70	E	D	CBA
80	E	D	CBA
90	F	E	DCBA
100	F	E	DCBA
110	G	F	EDCBA
120	G	F	EDCBA

Table B

The first item A enters the system at zero hours and leaves at 30 hours. It has a cycle time of 30 hours. All other items that enter the system after A also have cycle time of 40 hours. But, other than item A, each item is in the workflow for an extra 10 hours of “idle time” because of the WIP limit of both the steps were set to one. The implement step takes more time per person per work item. So, the flow of items through the workflow has an additional lag of 10 hours per item except for the first item. As we can see, setting WIP limits arbitrarily to any value is not a good idea. What if we added one additional person to the implement step? That depends on the nature of the job. Some jobs must be done by one person while other jobs are better candidates to be done by a group. If we assume that two people are assigned to the implement step, it would reduce the time taken to process each item considerably. In that case, the idle time may be reduced or even eliminated. So, in addition to setting WIP limits, how a team approaches the work also impact’s a system’s efficiency. Setting WIP limits requires understanding a few general rules (covered in the lessons and the next scenario) and a little bit of hit and trial. Over a period of time, your team can adjust optimal WIP limits that work best for you. And you should continue to adjust the WIP limits as your team gains experience. The key thing to learn is that WIP needs to be **limited** to an optimal value, which is not the same as **minimizing** WIP.

Kanban Scenario Three - Calculate WIP Limits with a Buffer: Set our WIP based on the slowest step. Adjust WIP proportionately for all other steps.

Since the implement step is the slowest, the team would have more people at this step. Let’s say we have two people at that step. With a 100 percent buffer, the WIP limit of the implement step will be two: 1.5 = 3. For the faster validate step of half the size, we can have half the number of resources. So, we have one person assigned to the validate step, which means a WIP limit = one: 1.5 = 1.5 rounded up to 2.

Table C shows how work items are processed through the workflow. For the implement step, it is assumed that only one person works on an item at a given point in time. So, each person is viewed as its own independent queue.

At Time (Hours)	Validate (WIP: 2) - 10 hours (1 Person)	Implement (WIP: 3) - 20 hours (Person 1, Person 2)	Items Processed
0	A	Idle, Idle	
10	B	A, Idle	
20	C	A, B	
30	D	C, B	A
40	E	C, D	BA

50	F	E, D	CBA
60	G	E, F	DCBA
70	H	G, F	EDCBA
80	I	G, H	FEDCBA
90	J	I, H	GFEDCBA
100	K	I, J	HGFEDCBA
110	L	K, J	IHGFEDCBA
120	....	....	....

Table C

The first item A enters the system at zero hours and leaves at 30 hours. It has a cycle time of 30 hours. B enters the system at 10 hours and leaves the system at 40 hours. So, B has a cycle time of 30 hours. All other items have a cycle time of 30. The extra idle time has been eliminated now.

As we can see, setting more optimized WIP limits reduces cycle time and also makes the system more efficient. Other key observations from Table C are:

1. A WIP limit of two does not mean the one person working on the validate step pulls two items to the queue. WIP limits are maximum limits, but a queue does not have to items to match the WIP limits. If a work item gets blocked at the validate step, the person working at that step could pull another item to continue working.
2. In Table C above, the team members work on exactly one work item at a given point in time. What would happen if they worked together on the same item? For example, if the two people worked together on item A as soon as it entered the implement step at 10 hours? We would not have any idle worker at 10 hours. What would happen if all three people worked on each item as it entered the queue at 0 hours? We would not have any idle worker at all! Multiple people working on the same item is often called “swarming” in the agile world. Whether “swarming” is a good idea is dependent on the nature of the job.
3. Based on scenarios two and three, it is evident that there are multiple permutations and combinations on which we can manage a workflow. Setting appropriate WIP limits is just one of the considerations of managing a workflow with Kanban. This is the reason why Kanban is considered a deceptively simple workflow management system.