IMPROVING PERFORMANCE OF PROCESS FLOWS

Chang-sun Chin[1] and Jeffrey S. Russell[2]

ABSTRACT

A process flow is a sequence of processes and stock points through which entities pass in sequence. At the level of a flow, the performance metrics related to overall system performance are throughput, cycle time and work-in-process. Understanding relationships between these metrics and flow behaviour is most important part to improve process flow performance and design high efficiency flows. A system can perform completely differently under different conditions. By comparing flow performance in a present state with those in theoretically possible states that a system can reach, we can determine whether a process flow is good or bad. The research defines process flow performance metrics as well as their relationships, and suggests a method to evaluate process flow performance using the flow metrics. The outcome will provide an internal benchmark of a process flow and different routes for process flow improvement.

KEY WORDS

process flow, bottleneck rate, raw process time, critical WIP, practical worst case performance, internal benchmark

INTRODUCTION

In most cases, we don't know when a process performs best or worst and what the best and/or the worst performance is. This is mainly because of the complexity of process interactions between workstations and their flow components (Hopp and Spearman 2000). Hence, it is critical to use appropriate key metrics which can reflect process interactions and flow behaviours when evaluating the process performance. Defining key flow performance metrics makes it possible to understand process flow components that make up a production or supply chain system, and improve the process flow performance (Hopp and Spearman 2000).

Let's start discussing the process flow performance metrics. We know that the metrics are related by Little's Law, i.e., WIP = TH × CT (Work-In-Process = Throughput x Cycle Time). But how else are they related? For instance, how is TH affected by WIP? If we reduce or increase WIP level in a given flow without making any other changes, what will happen to output? What factors make a system capable of achieving a high level of TH with a low WIP? These are important questions at the root of lean production practices. Hence, understanding the

[1] Ph.D. Candidate, Construction Engineering and Management Program, Department of Civil Engineering, University of Wisconsin, Madison, chin2@wisc.edu

[2] Professor and Chair, Department of Civil and Environmental Engineering, University of Wisconsin, Madison, russell@engr.wisc.edu

essence of flow behaviour and relationships between flow metrics is most important part to design and maintain high efficiency flows.

For the selected topic, reinforcing bar (hereinafter, rebar) detailing process was chosen. One of authors conducted observations on rebar detailing process from a couple of projects in different regions in the United States and could understand that in general, rebar detailing involves two distinct processes – 1) bar lists and placing drawings (aka, bar bending schedule or BBS) production and 2) review/approval. The research used data from one of projects which was to build an 8-story building in CA, USA. From the observation, we could see that lead time for BBS production is about one month and review/approval requires an additional one month after BBS production is completed. Looking at physical production (fabrication and on-site assembly) lead time for rebar, we could observe that it just takes a couple of weeks. Hence, reducing detailing lead time will have a greater impact on overall rebar supply chain performance. The research will focus on review/approval process rather than BBS production. Review/approval process is an obviously non-value added activity but can not be removed immediately from the process due to its unique function – i.e., risk reduction from the business perspective (George 2002). Figure 1 represents the rebar detailing review/approval process which clearly defines each step including process rates and cycle times of each process step. Note that the inverse of the cycle time gives the instantaneous rate of throughput of each workstation.

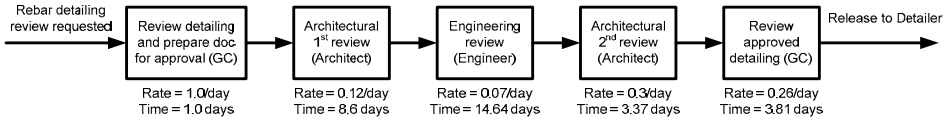

Figure 1: Rebar Detailing Review/Approval Process

CHARACTERIZING FLOWS

The basic flow behaviors can be described by the two parameters: bottleneck rate (r_b) and raw process time (T_0). The bottleneck rate decides the capacity of the process flow (i.e. the rate of the process with the highest utilization) and the raw process time is the time entities spend being processed in the flow (Hopp and Spearman 2000). One can expect the maximum TH and minimum CT with a certain level of WIP. Conversely, one can expect the minimum CT by reducing the WIP level with holding the same TH. Hence, we can estimate the best possible performance for a production line with a given bottleneck rate and raw process time. Controlling WIP, however, is much easier than improving completion rate (throughput) because WIP can be observed directly (Spearman and Zazanis 1992). In other words, one can speed up any process flow by reducing the amount of WIP even if one does nothing to improve completion rate.

BEST-CASE PERFORMANCE

The best-case performance is made under the situation where the system does not reach a special WIP level which leads both maximum throughout and minimum cycle time in a process flow, which is called the critical WIP and computed as W0 = rb·T0 (Hopp and Spearman 2000). If we could observe the larger WIP level than W0 in the current system, the process flow is to experience queueing delay. In order to evaluate the best-case performance, first we need to measure the bottleneck rate (rb) and raw process time (T0). If we assume there is no detractors which reduce the workstation capacities in the form of rework, repair, yield loss etc, the bottleneck of the detailing review/approval process is obviously "Engineering review" since it has least capacity (i.e. longest processing time) and all jobs pass through all processes. In other words, "Engineering review" has the highest utilization among the processes in the flow.

Table 1: Process Times and Rates

Process	No. of Server	Process Time (days)	Rate (jobs/day)	Remark
Review detailing and prepare doc for approval (GC)	1	1.00	1.00	
Architectural 1st review (Architect)	1	8.60	0.12	
Engineering review (Engineer)	1	14.64	0.07	Bottleneck
Architectural 2nd review (Architect)	1	3.37	0.30	
Review approved detailing (GC)	1	3.81	0.26	
Total Process Time (T_0)		31.42		

The raw process time (T0) is simply a sum of process times, which is T0 = 31.42 days. Therefore, the critical WIP for the process is computed:

$$W_0 = r_b \times T_0 = 0.07 \times 31.42 = 2.2 \text{ jobs}$$

WORST-CASE PERFORMANCE

Any processes involve some degree of variability which always degrades their performance. In this case, throughput decreases and cycle time will increase given parameters r_b and T_0. The worst cycle time will occur when variability reaches a maximum level. One possible maximum variability scenario is the case that if all the jobs in the line move together between workstations. Under this condition, if there are w jobs in the production line, the time to get through each workstation will be the average WIP level (w) times the process time at that workstation. However, one thing we should know is that both the best and worst case performances are not related to randomness or uncertainty in the process times (i.e. process times of both the best and worst cases are deterministic). The reason for the worst performance is definitely variability and one of reasons of this type of variability is a bad control (Hopp and Spearman 2000).

PRACTICAL WORST CASE (PWC) PERFORMANCE

We can observe a big gap between Best-Case and Worst-Case performances. These theoretical extreme cases are not very practical to evaluate actual system performance because most systems perform intermediately. Hence, we need a more realistic point for comparison. The

because most systems perform intermediately. Hence, we need a more realistic point for comparison. The question arise at this moment, "Can we find an intermediate case that divides "good" and "bad" regions, and is computable?" We can do this with an experiment under a special condition. Hopp and Spearman (2000) illustrated the possible combination of jobs to be processed in the system in order to explain the Practical Worst Case. Suppose that a system consists of three jobs and four workstations. By varying the number of jobs in the system, one can observe different system states[1] caused by randomness. If there is one job each at workstations, no queueing delay will occur because the system never has chance to reach the critical WIP level (W_0). If all three jobs are at the first workstation, the rest workstations will have no jobs and its performance will become worst cases. However, if we could assume 1) a balanced flow 2) single server workstations 3) moderately high variability, then all possible states become *equally likely*[2]. (Hopp and Spearman 2000) Under this special condition, the system will become a maximum randomness case which is termed Practical Worst Case and will be an intermediate case between the best and worst cases.

INTERNAL BENCHMARKING

As we observed, the system parameters of the selected process are: W_0 =2.2 jobs, T_0 = 31.42 days, r_b = 0.07 jobs/day. Based upon the concept illustrated in the previous section, we can compute CT and TH with respect to the each case – best, worst, and PWC. The computation can be simply done using spreadsheet by varying the WIP level.

[1] The state of the system is a complete description of the jobs at all the workstations: how many there are and how long they have been in process. (Hopp and Spearman 2000)

[2] For more details, see "Factory Physics (2nd edition)," p.231

Table 2: WIP, CT and TH of Rebar Detailing Review/Approval Process

WIP (w)	Best-Case		Worst-Case		PWC	
	CT_{best}	TH_{best}	CT_{worst}	TH_{worst}	CT_{PWC}	TH_{PWC}
	T_0 if $w \le W_0$; w/r_b otherwise.	w/T_0 if $w \le W_0$; r_b otherwise	wT_0	$1/T_0$	$T_0 + (w - 1)/r_b$	$[w/(W_0 + w - 1)] \cdot r_b$
1	31.42	0.03	31.42	0.03	31.42	0.03
2	31.42	0.06	62.84	0.03	45.71	0.04
3	42.86	0.07	94.26	0.03	59.99	0.05
4	57.14	0.07	125.68	0.03	74.28	0.05
5	71.43	0.07	157.10	0.03	88.56	0.06
6	85.71	0.07	188.52	0.03	102.85	0.06
7	100.00	0.07	219.94	0.03	117.13	0.06
8	114.29	0.07	251.36	0.03	131.42	0.06
9	128.57	0.07	282.78	0.03	145.71	0.06
10	142.86	0.07	314.20	0.03	159.99	0.06
11	157.14	0.07	345.62	0.03	174.28	0.06
12	171.43	0.07	377.04	0.03	188.56	0.06
13	185.71	0.07	408.46	0.03	202.85	0.06
14	200.00	0.07	439.88	0.03	217.13	0.06
15	214.29	0.07	471.30	0.03	231.42	0.06
16	228.57	0.07	502.72	0.03	245.71	0.07
17	242.86	0.07	534.14	0.03	259.99	0.07
18	257.14	0.07	565.56	0.03	274.28	0.07
19	271.43	0.07	596.98	0.03	288.56	0.07
20	285.71	0.07	628.40	0.03	302.85	0.07

As expected, in the best case, CT extremely increases but TH does not increase once the WIP level exceeds the critical WIP (W_0) since the system is not capable to work beyond its capacity. In the worst case, CT increases as WIP level increases but TH was constant because of the assumption that all the jobs in the line move together between workstations. In the PWC case, we could observe the intermediate performance level. This is because the PWC involves the maximum randomness in the system flow different from both the best and worst cases which have no randomness. Since the PWC causes the system to progress through the best case, the worst case, and all states in between, we can expect the PWC to show performance between that of the best and the worst cases. For a system given r_b and T_0, we can determine that it is "bad" if its performance is worse than that of the PWC and "good" if it is better than that of the PWC. (Hopp and Spearman 2000) Hence, simply by observing throughput (or cycle time) and average number of jobs in the system (WIP level) and comparing them with the results of three cases (i.e., best, worst and PWC), we can determine whether a process flow is good or bad.

Suppose that TH of current system has averaged 0.05 jobs/day and the average number of jobs in progress (WIP level) has been 10 jobs. Putting this numbers on the plots resulting from best, worst, and PWC calculation, we will see where the system is placed in. Figure 2 indicates the current system is in the bad region (between the worst case and the PWC) which meaning the current process flow is bad, i.e., has a serious problem (or has

room for improvement). In order to improve the process flow performance (i.e. move to "Good" region), we can think of two different options.

- **Option 1**: Reduce WIP level – because we know the critical WIP level (2.2 units), we can have a target to reduce WIP level (i.e. below or close to 2.2 units). By reducing WIP to critical WIP, then we can expect the higher TH.
- **Option 2**: Increase TH rate – this is a direct method to increase the process flow performance by adding more resources (servers) into workstations.
- Increasing the bottleneck rate by means of adding resources (equipment or staff), training, use of flexible labor, quality improvement etc (Hopp and Spearman 2000).
- Increasing the bottleneck utilization by means of use of buffer (WIP, capacity or time) which reduces blocking and starving of the bottleneck. In this case, we should consider that high utilization without restriction on WIP causes infinite queueing and hence increases CT (See Appendix II). Hence, we should examine the trade-off between CT increase and buffering benefit.

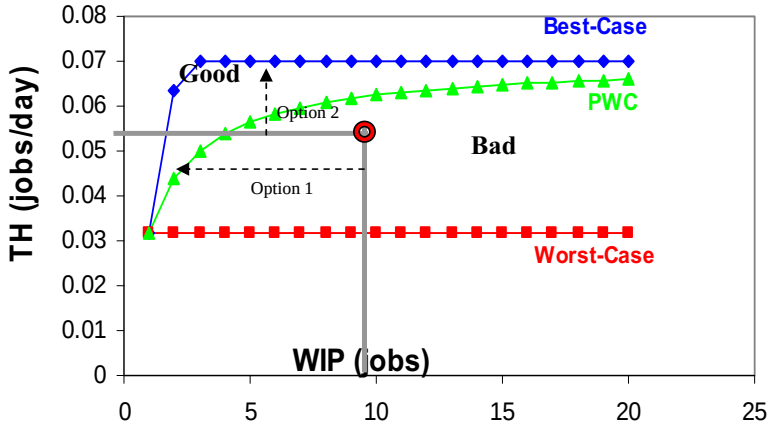

Figure 2: TH vs. WIP and current performance efficiency

In addition, we can compare the system performance with WIP vs. CT graph (Figure 3) by converting TH to CT by the Little's Law (WIP = CT·TH), i.e. CT = WIP/TH = 10/0.05, so we will get CT = 200 days. The WIP vs. CT graph also indicates the current system is under PWC performance and placed in the bad region.

- **Option 1**: Reduce CT - by the Little's Law (TH = WIP/CT), CT

reduction implies WIP reduction while throughput remains constant. Hence, large queues are an indication of opportunities for reducing CT (Hopp and Spearman 2000). There are many ways to accomplish the cycle time reduction from the production system perspectives but there are three key points involved in cycle time reduction. (Hopp et al 1990)

- Queueing and waiting time reduction – queueing and waiting times are large fraction of the process time so that it makes sense that reducing them results in flow time reduction.
- WIP reduction – WIP and flow time are proportional to each other for a given level of throughput. In other words, causes of excessive lead time can be determined by identifying locations with large inventories (WIP).
- Variation reduction – cycle time is related not only to the average of flow time but also to the variation of flow time (See Appendix II).
- **Option 2**: Increase WIP - one might argue that performance (TH) can be increased when WIP increases by the Little's Law (TH = WIP/CT). Yes, it might make sense mathematically but increasing WIP directly violates the basic lean production concept. The definition of WIP is all the unfinished parts or products that have been released to a production line so that large WIP means loss of production opportunity and lots of wastes, generating additional negative impacts to the process flow. However, if a process has a large variation, increase of WIP level to a certain level will work as a buffer so as to prevent the possible congestion or shortage of entities in the flow (Conway et al 1988).

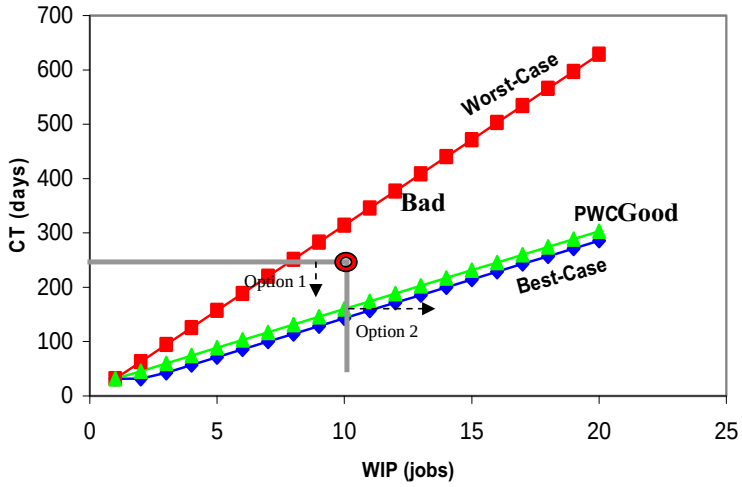

Figure 3: CT vs. WIP and Current Process Flow Performance Efficiency

IMPROVING PERFORMANCE OF PROCESS FLOWS

The method suggested in this paper explains us how to determine the process flow performance given parameters r_b and T_0. One thing to be addressed is that process flow performance also can be improved if we could eliminate detractors from the system. Based on the previous discussion about flow components relationship and flow behavior, we can summarize improvement strategies as following:

Firstly, we can attain better process flow performance by improving system parameters (i.e. improving r_b and T_0).

- Increase of bottleneck rate (r_b) by adding capacity by means of adding resources (equipment or staff), training, use of flexible labor, quality improvement etc); or increasing the bottleneck utilization by means of use of buffer (WIP, capacity or time) as described previously.
- Reduction of process time (T_0) by shortening the process time as described in the previous section – i.e., queueing and waiting time reduction; WIP reduction; or variation reduction.

Secondly, we can achieve better process flow performance by improving performance given parameters (r_b and T_0). This is quite related to the batching and variability effect (Hopp and Spearman 2000). In other to understand this, we need little knowledge about batching effect and queueing theory (See Appendix II).

- Reducing batching delay at or between processes by means of setup reduction, better scheduling, and/or more efficient material handling.
- Reducing delays caused by variability by means of changes in products, processes, operators, and management that enable smoother flows through and between workstations. In the RFI process, we can find the step whose variability is significantly larger than those of other steps – i.e. Architectural 2nd review and then we can select this step as a first target for variability reduction.

CONCLUSION

The research presented how to diagnose process flow performance efficiency by measuring key flow parameters. Suppose that you are involved in a process and want to know about how efficiently the process performs. From the lean production perspective, you might start measuring Process Cycle Efficiency (PCE) which is equivalent to the ratio of value-added time to total lead time required for producers to deliver goods to customers (George et al 2005). It will tell you about how fast your system can response to the customer's demand – indicating healthiness of the system, i.e. "LEANNESS." The larger the PCE the leaner the system because the system has less fraction of non-value added times. However, PCE never tell you about how flow parameters are related and how you can improve process flow performance. Well, you can eliminate non-value added activities from the process. It looks simple and easy to remove non-value added portions from your process but is not simple enough particularly when activities in the process have complex

interactions with others (e.g. approval process can not be eliminated from the current system immediately because it has an important function of risk reduction even if it creates no value from the customer's perspective). Hence, you will need more sophisticated method to improve the process flow performance.

The method suggested in the research clearly explains how to diagnose the process flow performance efficiency, how flow parameters are related, and how you can improve process flow performance simultaneously. Just by collecting four flow parameters: bottleneck rate (r_b), raw process time (T_0), average WIP level (w), and actual throughput (TH), you can immediately evaluate your system flow performance – i.e., "is it good or bad performance?" as well as you will have clear directions for process flow improvement. The steps you should take are: 1) understand process flow (by means of process flow diagram); 2) collect four flow parameters (bottleneck rate (r_b), raw process time (T_0), average WIP level (w), and actual throughput); 3) compute TH_{PWC}; 4) compare actual TH with TH_{PWC} – if $TH > TH_{PWC}$, your process flow is good, otherwise it is bad, and 5) plot TH vs. WIP or CT vs. WIP graphs and look where your process is placed. Then, it will give you directions for improvement.

As Zipkin (1991) described, one might have a dream of romantic Just-In-Time (JIT) but real world is so complex that one cannot achieve the true JIT in practice without understanding of flow parameters and their relationships as discussed in this research.

REFERENCES

Conway, R., Maxwell, W., McClain, J.O. and Thomas, L.J. (1988). "The Role of Work-In-Process Inventory in Serial Production Lines." *Operations Research*, Vol. 36, No. 2.

George, M.L. (2002). *Lean Six Sigma, Combining Six Sigma Quality with Lean Speed*. McGraw-Hill.

George, M.L., Rowlands, D., Price, M. and Maxey, J. (2005). *Lean Six Sigma Pocket Tool book*. McGraw-Hill.

Hopp, W.J. and Spearman, M.L. and Woodruff, D.L. (1990). "Practical Strategies for Lead Time Reduction." American Society of Mechanical Engineers. *Manufacturing Review,* Vol. 3, No. 2.

Hopp, W.J. and Spearman, M.L. (2000). *Factory Physics: Foundations of Manufacturing Management.* Irwin/McGraw-Hill.

Lambrecht, M., and Vandaele, N. (1994). "Queueing Theory and Operations Management." Tijdschrift voor Economie en Management. Vol. XXXIX, No. 4, p. 415-424.

Spearman, M.L. and Zazanis, M.A. (1992). "Push and Pull Production Systems: Issues and Comparisons." Operations Research Society of America, *Operations Research*, Vol. 40, No. 3, p. 521-532.

Zipkin, P.H. (1991), "Does manufacturing need a JIT revolution?" *Harvard Business Review*, Vol. 40, p. 40-50.

APPENDIX I: BEST, WORST AND PRACTICAL WORST CASE PERFORMANCES

BEST CASE PERFORMANCE

$$CT_{best} = \begin{cases} T_0 & \text{if } w \le W_0 \\ w/r_b & \text{otherwise.} \end{cases}$$

$$TH_{best} = \begin{cases} w/T_0 & \text{if } w \le W_0 \\ r_b & \text{otherwise.} \end{cases}$$

WORST CASE PERFORMANCE

$$CT_{worst} <= wT_0$$
$$TH_{worst} >= 1/T_0$$

PRACTICAL WORST CASE PERFORMANCE

Suppose that the system has N workstations with a single server each, a constant level of w jobs in the system, and the average processing time at each workstation is t. The raw process time (T_0) will be the number of workstations times the average processing time at each workstation (i.e. $N \cdot t$) and the bottleneck rate (r_b) will be $1/t$. Under the PWC conditions, when a marked job arrives at a workstation, the other w-1 jobs will be evenly distributed among the N workstations each time a marked job arrives at a workstation. Therefore, on average, the expected number of jobs ahead of a marked job will be (w-1)/N jobs. Since the average time a marked job spent at the workstation will be the sum of the process time for the marked job and the process times for jobs to be processed, the average time at a workstation would be:

$$t + [(w-1)/N]t$$

In addition, since all N workstations are assumed identical under the PWC conditions, the total flow time for the line can be computed by multiplying the average time a marked job spent at the workstation by the number of workstations,

$$CT_{PWC} = N\{t+[(w-1)/N]t\} = N \cdot t + (w-1)t = T_0 + (w-1)/r_b$$

By the Little's Law, TH = WIP/CT and the fact that $W_0 = r_b T_0$, we can compute the throughput as a function of WIP level as following:

$$TH_{PWC} = [w/(W_0 + w - 1)] \cdot r_b$$

APPENDIX II: QUEUEING AND BATCHING EFFECTS

QUEUEING EFFECT

Queueing delay occurs when a number of physical entities attempts to receive service from servers(s) having limited capacity and as a result the entities being arrived at workstation should sometimes wait in the line until server(s) is available (Lambrecht and Vandaele 1994).

One of well established GENRAL type of queueing models is G/G/1 model. The term “general” means the arrival time and service time can take on any probability distribution. Note that the first G denotes the type of distribution of inter-arrival time, the second G denotes the type of distribution of effective process times, and the last number “1” describes the number of servers at the workstation, respectively.

$$W_q^{G/G/1} \approx \left(\frac{c_a^2 + c_s^2}{2} \right) \left(\frac{\rho}{1-\rho} \right) \tau$$

Note that the C_a, C_s, ρ, and τ denote the inter-arrival time Coefficient of Variation (CV), the effective process time CV, the utilization, and the average service time respectively. The expression explains that queueing delay is the multiples of variation in the inter-arrival time and effective process time, utilization and the average service time. Hence, by exploring the underlying causes of these four parameters, we can systematically identify the factors that cause waiting in a given queueing system (Hopp and Spearman 2000)

BATCHING EFFECT

This happens whenever jobs are batched together for delivery to a workstation. Since arrivals always occur in this way with no randomness whatever, one might reasonably interpret the variation and the CV to be zero. However, a very different picture results from looking at the inter-arrival times of the jobs in the batch from the perspective of the individual job. Hopp and Spearman (2000) describes the reason for batching effect as following:

"The first effect is due to the batching itself. This is not really a randomness issue, but rather one of bad control. The second is the variation in the batch arrival themselves (i.e., as characterized by the arrival CV for the batches). If jobs are arrived in batches, the inter-arrival time of the first job in the batch will be the time the jobs arrive to the workstation but the inter-arrival times of the rest in a batch will be zero."