The Corrupting Influence of Variability

*When luck is on your side, you can do without brains.*

– Giordano Bruno, burned at the stake in 1600

*The more you know the luckier you get.*

– “J.R. Ewing” of *Dallas*

---

Performance of a Serial Line

**Measures:**
- Throughput
- Inventory (RMI, WIP, FGI)
- Cycle Time
- Lead Time
- Customer Service
- Quality

**Evaluation:**
- Comparison to “perfect” values
- Relative weights consistent with business strategy?

**Links to Business Strategy:**
- Would inventory reduction result in significant cost savings?
- Would CT (or LT) reduction result in significant competitive advantage?
- Would TH increase help generate significantly more revenue?
- Would improved customer service generate business over the long run?
Influence of Variability

Variability Law: Increasing variability always degrades the performance of a production system.

Examples:
- process time variability pushes best case toward worst case
- higher demand variability requires more safety stock for same level of customer service
- higher cycle time variability requires longer lead time quotes to attain same level of on-time delivery

Variability Buffering

Buffering Law: Systems with variability must be buffered by some combination of:

1. inventory
2. capacity
3. time.

Interpretation: If you cannot pay to reduce variability, you will pay in terms of high WIP, under-utilized capacity, or reduced customer service (i.e., lost sales, long lead times, and/or late deliveries).
Variability Buffering Examples

Ballpoint Pens:
- can’t buffer with time (who will backorder a cheap pen?)
- can’t buffer with capacity (too expensive, and slow)
- must buffer with inventory

Ambulance Service:
- can’t buffer with inventory (stock of emergency services?)
- can’t buffer with time (violates strategic objectives)
- must buffer with capacity

Organ Transplants:
- can’t buffer with WIP (perishable)
- can’t buffer with capacity (ethically anyway)
- must buffer with time

Simulation Studies

TH Constrained System (push)

WIP Constrained System (pull)

\[ r_a = \text{arrival rate} \]
\[ c_a = \text{CV of interarrival times} \]
\[ t_e(i) = \text{effective process time at station } i \]
\[ c_e(i) = \text{effective CV at station } i \]
\[ B(i) = \text{buffer size in front of station } i \]
Variability Reduction in Push Systems

<table>
<thead>
<tr>
<th>Case</th>
<th>$t_1$, $i = 1, 2, 4$ (min)</th>
<th>$t_3$ (min)</th>
<th>$c(i), i = 1-4$ (unitless)</th>
<th>TH (j/min)</th>
<th>CT (min)</th>
<th>WIP (jobs)</th>
<th>$\sigma_{CT}$ (min)</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1.2</td>
<td>0</td>
<td>0.8</td>
<td>4.2</td>
<td>3.4</td>
<td>0.0</td>
<td><strong>best case</strong></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1.2</td>
<td>1</td>
<td>0.8</td>
<td>44.6</td>
<td>35.7</td>
<td>26.8</td>
<td><strong>WIP buffer</strong></td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1.0</td>
<td>0</td>
<td>0.8</td>
<td>20.0</td>
<td>16.0</td>
<td>10.3</td>
<td><strong>capacity buffer</strong></td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1.2</td>
<td>0.3</td>
<td>0.8</td>
<td>7.8</td>
<td>6.2</td>
<td>3.5</td>
<td><strong>reduced variability</strong></td>
</tr>
</tbody>
</table>

Notes:
- $r_p = 0.8, c_j = c_j(i)$ in all cases.
- $B(i) = \infty, i = 1-4$ in all cases.

Observations:
- TH is set by release rate in a push system.
- Increasing capacity ($r_p$) reduces need for WIP buffering.
- Reducing process variability reduces WIP for same TH, reduces CT for same TH, and reduces CT variability.

Variability Reduction in Pull Systems

<table>
<thead>
<tr>
<th>Case</th>
<th>$t_1$, $i = 1, 2, 4$ (min)</th>
<th>$t_3$ (min)</th>
<th>$c(i), i = 1-4$ (unitless)</th>
<th>$B(i)$ (jobs)</th>
<th>TH (j/min)</th>
<th>CT (min)</th>
<th>WIP (jobs)</th>
<th>$\sigma_{CT}$ (min)</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1.2</td>
<td>0</td>
<td>0.84</td>
<td>4.6</td>
<td>3.8</td>
<td>0.0</td>
<td><strong>best case</strong></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1.2</td>
<td>1</td>
<td>0.48</td>
<td>6.4</td>
<td>3.1</td>
<td>2.4</td>
<td><strong>plain JIT</strong></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1.2</td>
<td>1</td>
<td>0.53</td>
<td>7.2</td>
<td>3.8</td>
<td>2.6</td>
<td><strong>inv buffer</strong></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1.2</td>
<td>0.3</td>
<td>0.72</td>
<td>5.0</td>
<td>3.6</td>
<td>0.6</td>
<td><strong>var reduction</strong></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1.2</td>
<td>0.3</td>
<td>0.76</td>
<td>6.0</td>
<td>4.5</td>
<td>0.8</td>
<td><strong>inv buffer + var reduction</strong></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1.2</td>
<td>0.3</td>
<td>0.73</td>
<td>6.3</td>
<td>4.6</td>
<td>0.7</td>
<td><strong>non-bottleneck buffer</strong></td>
<td></td>
</tr>
</tbody>
</table>

Notes:
- Station 1 pulls in job whenever it becomes empty.
- $B(i) = 0, i = 1, 2, 4$ in all cases, except case 6, which has $B(2) = 1.$
Variability Reduction in Pull Systems (cont.)

Observations:
- Capping WIP without reducing variability reduces TH.
- WIP cap limits effect of process variability on WIP/CT.
- Reducing process variability increases TH, given same buffers.
- Adding buffer space at bottleneck increases TH.
- Magnitude of impact of adding buffers depends on variability.
- Buffering less helpful at non-bottlenecks.
- Reducing process variability reduces CT variability.

Conclusion: if you can’t pay to reduce variability now, you will pay later with lost throughput, wasted capacity, inflated cycle times, excess inventory, long lead times, or poor customer service!

Variability from Batching

VUT Equation:
- CT depends on process variability and flow variability

Batching:
- affects flow variability
- affects waiting inventory

Conclusion: batching is an important determinant of performance
Process Batch Versus Move Batch

Dedicated Assembly Line: *What should the batch size be?*

**Process Batch:**
- Related to length of setup.
  - The longer the setup the larger the lot size required for the same capacity.

**Move (transfer) Batch:** *Why should it equal process batch?*
- The smaller the move batch, the shorter the cycle time.
- The smaller the move batch, the more material handling.

**Lot Splitting:** *Move batch can be different from process batch.*
1. Establish smallest economical move batch.
2. Group move batches of like families together at bottleneck to avoid setups.
3. Implement using a “backlog”.

Process Batching Effects

**Types of Process Batching:**
1. **Serial Batching:**
   - processes with sequence-dependent setups
   - “batch size” is number of jobs between setups
   - batching used to reduce loss of capacity from setups

2. **Parallel Batching:**
   - true “batch” operations (e.g., heat treat)
   - “batch size” is number of jobs run together
   - batching used to increase effective rate of process
Process Batching

**Process Batching Law:** In stations with batch operations or significant changeover times:

- The minimum process batch size that yields a stable system may be greater than one.
- As process batch size becomes large, cycle time grows proportionally with batch size.
- Cycle time at the station will be minimized for some process batch size, which may be greater than one.

**Basic Batching Tradeoff:** WIP versus capacity

---

Serial Batching

**Parameters:**

- $k =$ serial batch size (10)
- $t =$ time to process a single part (1)
- $s =$ time to perform a setup (5)
- $c_v =$ CV for batch (parts + setup) (0.5)
- $r_a =$ arrival rate for parts (0.4)
- $c_a =$ CV of batch arrivals (1.0)

**Time for batch:** $t_e = kt + s$  
($t_e = 15$)

---

http://factory-physics.com
Process Batching Effects (cont.)

Arrival of batches: \( r_i/k \)
\((0.4/10 = 0.04)\)

Utilization: \( u = (r_i/k)(kt + s) = r_i(t + s/k) \)
\((0.4(5/10 + 1) = 0.6)\)

For stability: \( u < 1 \) implies
\[ k > \frac{sr_i}{1-tr_i} \quad \text{or} \quad (k > \frac{5.0(0.4)}{1-(1.0)(0.4)} = 3.33) \]
minimum batch size required for stability of system...

Average queue time at station:
\[ CT = \left( \frac{c^2_o + c^2_e}{2} \right) \left( \frac{u}{1-u} \right) t \]
\[ = \left( \frac{1 + 0.5}{2} \right) \left( \frac{0.6}{1-0.6} \right) \]
\[ = 15 = 16.875 \]

Average cycle time depends on move batch size:

- Move batch = process batch
\[ CT_{\text{non-split}} = CT_o + t_c = CT_o + s + kt \]
\[ = 16.875 + 15 = 31.875 \]

- Move batch = 1
\[ CT_{\text{split}} = CT_o + s + \frac{k+1}{2} t \]
\[ = 16.875 + 10 + \frac{10+1}{2}(1.0) = 27.375 \]

Note: we assume arrival CV of batches is \( c_o \) regardless of batch size – an approximation...

Note: splitting move batches reduces wait for batch time.
Cycle Time vs. Batch Size – 5 hr setup

Cycle Time vs Batch Size – 2.5 hr setup
Setup Time Reduction

Where?
- Stations where capacity is expensive
- Excess capacity may sometimes be cheaper

Steps:
1. Externalize portions of setup
2. Reduce adjustment time (guides, clamps, etc.)
3. Technological advancements (hoists, quick-release, etc.)

Caveat: Don’t count on capacity increase; more flexibility will require more setups.

Parallel Batching

Parameters:
- \( k = \) parallel batch size (10)
- \( t = \) time to process a batch (90)
- \( c_a = CV \) for batch (1.0)
- \( r_a = \) arrival rate for parts (0.05)
- \( c_a = CV \) of batch arrivals (1.0)
- \( B = \) maximum batch size (100)

Time to form batch: \( W = \frac{k - 1}{2} \frac{1}{r_a} \)  
\( \left( (10 - 1)/2(1/0.005) = 90 \right) \)

Time to process batch: \( t_e = t \)  
\( (t_e = 90) \)
Parallel Batching (cont.)

Arrival of batches: \( r_a/k \)

(0.05/10 = 0.005)

Utilization: \( u = (r_a/k)(t) \)

((0.005)(90) = 0.45)

For stability: \( u < 1 \) implies

\[ k > r_a t \quad \text{or} \quad (k > (0.05)(90) = 4.5) \]

minimum batch size required
for stability of system...

Parallel Batching (cont.)

Average wait-for-batch time:

\[ WT = \frac{k-1}{2} \frac{10-1}{r_a 0.05} = 90 \]

Average queue plus process time at station:

\[ CT = \left( \frac{c^2_i / k + c^2_j}{2} \right) \left( \frac{u}{1-u} \right) t + t = \left( \frac{0.1+1}{2} \right) \left( \frac{0.45}{1-0.45} \right) 90 + 90 = 130.5 \]

Total cycle time:

\[ CT + WT = 90 + 130.5 = 220.5 \]
**Move Batching**

*Move Batching Law:* Cycle times over a segment of a routing are roughly proportional to the transfer batch sizes used over that segment, provided there is no waiting for the conveyance device.

**Insights:**
- Basic Batching Tradeoff: WIP vs. move frequency
- Queuing for conveyance device can offset CT reduction from reduced move batch size
- Move batching intimately related to material handling and layout decisions
Move Batching

Problem:
- Two machines in series
- First machine receives individual parts at rate $r_a$ with CV of $c_a(1)$ and puts out batches of size $k$.
- First machine has mean process time of $t_e(1)$ for one part with CV of $c_e(1)$.
- Second machine receives batches of $k$ and put out individual parts.
- How does cycle time depend on the batch size $k$?

\[ r_a, c_a(1) \quad t_e(1), c_e(1) \quad k \quad t_e(2), c_e(2) \]

Station 1 \quad Station 2

Move Batching Calculations

Time at First Station:
- Average time before batching is:
  \[ \frac{c_a^2(1) + c_e^2(1)}{2} \frac{u(l)}{1-u(l)} t_e(1) + t_e(1) \]
- Average time forming the batch is:
  \[ \frac{k-1}{2} \frac{u(l)}{r_a} = \frac{k-1}{2u(l)} t_e(1) \]
- Average time spent at the first station is:
  \[ \text{CT}(l) = \frac{c_a^2(1) + c_e^2(1)}{2} \frac{u(l)}{1-u(l)} t_e(1) + t_e(1) + \frac{k-1}{2u(l)} t_e(1) \]
  \[ = \text{CT}(1, \text{no batching}) + \frac{k-1}{2u(l)} t_e(1) \]
Move Batching Calculations (cont.)

Output of First Station:

- Time between output of individual parts into the batch is $t_e$.
- Time between output of batches of size $k$ is $k t_e$.
- Variance of interoutput times of parts is $c^2_j(1) t_e^2$, where
  
  $$c^2_j(1) = (1-u(1))^2 c^2_j(1) + u(1) t_e^2$$

- Variance of batches of size $k$ is $k c^2_j(1) t_e^2$.
- SCV of batch arrivals to station 2 is:
  
  $$c^2_a(2) = \frac{k c^2_j(1) t_e^2}{k^2 t_e^2} = \frac{c^2_j(1)}{k}$$

  because departures are independent, so variances add variance divided by mean squared...

Time at Second Station:

- Time to process a batch of size $k$ is $k t_e(2)$.
- Variance of time to process a batch of size $k$ is $k c^2_j(2) t_e^2(2)$.
- SCV for a batch of size $k$ is:
  
  $$\frac{k c^2_j(1) t_e^2(2)}{k^2 t_e^2(2)} = \frac{c^2_j(1)}{k}$$

  independent process times...

- Mean time spent in partial batch of size $k$ is:
  
  $$\frac{k-1}{2} t_e(2)$$

  first part doesn’t wait, last part waits $(k-1) t_e(2)$, so average is $(k-1) t_e(2)/2$

- So, average time spent at the second station is:
  
  $$CT(2) = \frac{c^2_j(1) + c^2_j(2) / k}{2} \frac{u(2)}{1-u(2)} t_e(2) + k \frac{k-1}{2} t_e(2) + t_e(2)$$

  VUT equation to compute queue time of batches...

  $\Rightarrow CT(2, \text{no batching}) + \frac{k-1}{2} t_e(2)$
Move Batching Calculations (cont.)

Total Cycle Time:
\[
CT(\text{batching}) = CT(\text{no batching}) + \frac{k-1}{2u(1)} t_1(1) + \frac{k-1}{2} t_1(2)
\]
\[
= CT(\text{no batching}) + \frac{(k-1)}{2} \left( \frac{t_1(1)}{u(1)} + t_1(2) \right)
\]

Insight:
- Cycle time increases with \( k \).
- Inflation term does not involve CV’s.
- Congestion from batching is more bad control than randomness.

Assembly Operations

Assembly Operations Law: The performance of an assembly station is degraded by increasing any of the following:

- Number of components being assembled.
- Variability of component arrivals.
- Lack of coordination between component arrivals.

Observations:
- This law can be viewed as special instance of variability law.
- Number of components affected by product/process design.
- Arrival variability affected by process variability and production control.
- Coordination affected by scheduling and shop floor control.
Attacking Variability

Objectives
- reduce cycle time
- increase throughput
- improve customer service

Levers
- reduce variability directly
- buffer using inventory
- buffer using capacity
- buffer using time

Cycle Time

Definition (Station Cycle Time): The average cycle time at a station is made up of the following components

\[
\text{cycle time} = \text{move time} + \text{queue time} + \text{setup time} + \text{process time} + \text{wait-to-batch time} + \text{wait-in-batch time} + \text{wait-to-match time}
\]

delay times typically make up 90% of CT

Definition (Line Cycle Time): The average cycle time in a line is equal to the sum of the cycle times at the individual stations less any time that overlaps two or more stations.
Reducing Queue Delay

\[ CT_q = V \times U \times t \]

\[ \left( \frac{C_o^2 + C_e^2}{2} \right) \left( \frac{u}{1-u} \right) \]

- **Reduce Variability**
  - failures
  - setups
  - uneven arrivals, etc.

- **Reduce Utilization**
  - arrival rate (yield, rework, etc.)
  - process rate (speed, time, availability, etc.)

Reducing Batching Delay

\[ CT_{batch} = \text{delay at stations} + \text{delay between stations} \]

- **Reduce Process Batching**
  - Optimize batch sizes
  - Reduce setups
    - Stations where capacity is expensive
    - Capacity vs. WIP/FT tradeoff

- **Reduce Move Batching**
  - Move more frequently
  - Layout to support material handling (e.g., cells)
Reducing Matching Delay

\[ CT_{\text{batch}} = \text{delay due to lack of synchronization} \]

Reduce Variability
- High utilization fabrication lines
- Usual variability reduction methods

Improve Coordination
- scheduling
- pull mechanisms
- modular designs

Reduce Number of Components
- product redesign
- kitting

Increasing Throughput

\[ TH = P(\text{bottleneck is busy}) \times \text{bottleneck rate} \]

Reduce Blocking/Starving
- buffer with inventory (near bottleneck)
- reduce system “desire to queue”

\[ CT_q = V \times U \times t \]

Increase Capacity
- add equipment
- increase operating time (e.g. spell breaks)
- increase reliability
- reduce yield loss/ rework

Reduce Variability
Reduce Utilization

Note: if WIP is limited, then system degrades via TH loss rather than WIP/CT inflation
Improving Customer Service

\[ LT = CT + z \sigma_{CT} \]

**Reduce Average CT**
- queue time
- batch time
- match time

**Reduce CT Visible to Customer**
- delayed differentiation
- assemble to order
- stock components

**Reduce CT Variability**
- generally same as methods for reducing average CT:
  - improve reliability
  - improve maintainability
  - reduce labor variability
  - improve quality
  - improve scheduling
  - etc…

---

Cycle Time and Lead Time

- \( CT = 10 \)
- \( \sigma_{CT} = 3 \)
- \( CT = 10 \)
- \( \sigma_{CT} = 6 \)
Diagnostics Using Factory Physics

**Situation:**
- Two machines in series; machine 2 is bottleneck
- $c_2^2 = 1$
- Machine 1: $t_b = 19$ min
  - $c_0^3 = 0.25$
  - MTTF = 48 hr
  - MTTR = 8 hr
- Machine 2: $t_b = 22$ min
  - $c_0^3 = 1$
  - MTTF = 3.3 hr
  - MTTR = 10 min
  - Space at machine 2 for 20 jobs of WIP
- Desired throughput 2.4 jobs/hr, not being met

**Diagnostic Example (cont.)**

**Proposal:** Install second machine at station 2
- Expensive
- Very little space

**Analysis Tools:**
- $CT_q = \frac{c_0^3 + c_2^2}{2} \frac{u}{1-u} t_b$
- $c_2^3 = u^2 c_1^3 + (1-u^2) c_0^3$

**Analysis:**

**Step 1:** At 2.4 job/hr
- $CT_q$ at first station is 645 minutes, average WIP is 25.8 jobs.
- $CT_q$ at second station is 892 minutes, average WIP is 35.7 jobs.
- Space requirements at machine 2 are violated!
Diagnostic Example (cont.)

Step 2: Why is $CT_q$ at machine 2 so big?
- Break $CT_q$ into

$$CT_q = \left(\frac{c_e^2 + c_a^2}{2}\right) \left(\frac{u}{1-u}\right)t_r = (3.16)(12.22)(23.11 \text{ min})$$
- The 23.11 min term is small.
- The 12.22 correction term is moderate ($u \approx 0.9244$)
- The 3.16 correction is large.

Step 3: Why is the correction term so large?
- Look at components of correction term.
- $c_e^2 = 1.04$, $c_a^2 = 5.27$.
- Arrivals to machine are highly variable.

Step 4: Why is $c_a^2$ to machine 2 so large?
- Recall that $c_a^2$ to machine 2 equals $c_d^2$ from machine 1, and

$$c_d^2 = u^2 c_a^2 + (1-u^2)c_e^2 = (0.887^2)(6.437) + (1-0.887^2)(1.0) = 5.27$$
- $c_a^2$ at machine 1 is large.

Step 5: Why is $c_e^2$ at machine 1 large?
- Effective CV at machine 1 is affected by failures,

$$c_e^2 = c_f^2 + 2.4(1-A)\frac{m}{l_o} = 0.25 + 6.18 = 6.43$$
- The inflation due to failures is large.
- Reducing MTTR at machine 1 would substantially improve performance.
Procoat Case – Situation

Problem:
- Current WIP around 1500 panels
- Desired capacity of 3000 panels/day
- Typical output of 1150 panels/day
- Outside vendor being used to make up slack

Proposal:
- Expose is bottleneck, but in clean room
- Expansion would be expensive
- Suggested alternative is to add bake oven for touchups

Procoat Case – Layout
Procoat Case – Capacity Calculations

<table>
<thead>
<tr>
<th>Machine Name</th>
<th>Process or Load Time (min)</th>
<th>Std Dev Process Time (min)</th>
<th>Conv. Trip Time (min)</th>
<th>Number of Machines</th>
<th>MTTF (min)</th>
<th>MTTR (min)</th>
<th>Avail Time (min)</th>
<th>Setup Time (min)</th>
<th>Rate (p/day)</th>
<th>Time (min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clean1</td>
<td>0.33</td>
<td>0</td>
<td>15</td>
<td>1</td>
<td>80</td>
<td>4</td>
<td>0.95</td>
<td>0</td>
<td>3377</td>
<td>37</td>
</tr>
<tr>
<td>Coat1</td>
<td>0.33</td>
<td>0</td>
<td>15</td>
<td>1</td>
<td>80</td>
<td>4</td>
<td>0.95</td>
<td>0</td>
<td>3377</td>
<td>37</td>
</tr>
<tr>
<td>Coat2</td>
<td>0.33</td>
<td>0</td>
<td>15</td>
<td>1</td>
<td>80</td>
<td>4</td>
<td>0.95</td>
<td>0</td>
<td>3377</td>
<td>37</td>
</tr>
<tr>
<td>Expose</td>
<td>103</td>
<td>67</td>
<td>-</td>
<td>5</td>
<td>300</td>
<td>10</td>
<td>0.97</td>
<td>15</td>
<td>2879</td>
<td>118</td>
</tr>
<tr>
<td>Develop</td>
<td>0.33</td>
<td>0</td>
<td>2.67</td>
<td>1</td>
<td>300</td>
<td>3</td>
<td>0.99</td>
<td>0</td>
<td>3510</td>
<td>23</td>
</tr>
<tr>
<td>Inspect</td>
<td>0.5</td>
<td>0.5</td>
<td>-</td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>1.00</td>
<td>0</td>
<td>4680</td>
<td>1</td>
</tr>
<tr>
<td>Bake</td>
<td>0.33</td>
<td>0</td>
<td>100</td>
<td>1</td>
<td>300</td>
<td>3</td>
<td>0.99</td>
<td>0</td>
<td>3510</td>
<td>121</td>
</tr>
<tr>
<td>MI</td>
<td>161</td>
<td>64</td>
<td>-</td>
<td>8</td>
<td>-</td>
<td>-</td>
<td>1.00</td>
<td>0</td>
<td>3498</td>
<td>161</td>
</tr>
<tr>
<td>Touchup</td>
<td>9</td>
<td>0</td>
<td>-</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>1.00</td>
<td>0</td>
<td>7800</td>
<td>9</td>
</tr>
</tbody>
</table>

rb = 2,879 p/day
T_θ = 542 min = 0.46 days
W_0 = rbT_θ = 1,334 panels

Procoat Case – Benchmarking

TH Resulting from PWC with WIP = 37,400:

\[
TH = \frac{W}{W + W_0} r_θ = \frac{1,500}{1,500 + 1,334} r_θ = 1,524
\]

Higher than actual TH

Conclusion: actual system is significantly worse than PWC.

Question: what to do?
Procoat Case – Factory Physics Analysis

1) Expose has highest utilization:
   ⇒ increase capacity via break spelling

2) Expose has high \( V \) coefficient (and limited room for WIP):
   ⇒ some is due to operator variability ⇒ implement training program
   ⇒ more is due to arrival variability ⇒ look at coater line

3) Coater line has long, infrequent failures:
   ⇒ maintain field ready replacements

---

Procoat Case – Outcome

![Graph showing WIP (panels) vs. TH (panels/day) with Best Case, Practical Worst Case, and Worst Case regions.](http://factory-physics.com)
Corrupting Influence Takeaways

Variance Causes Congestion:
- many sources of variability
- planned and unplanned

Variability and Utilization Interact:
- congestion effects multiply
- utilization effects are highly nonlinear
- importance of bottleneck management

Corrupting Influence Takeaways (cont.)

Variability Propagates:
- flow variability is as disruptive as process variability
- non-bottlenecks can be major problems

Variability Must be Buffered:
- inventory
- capacity
- time