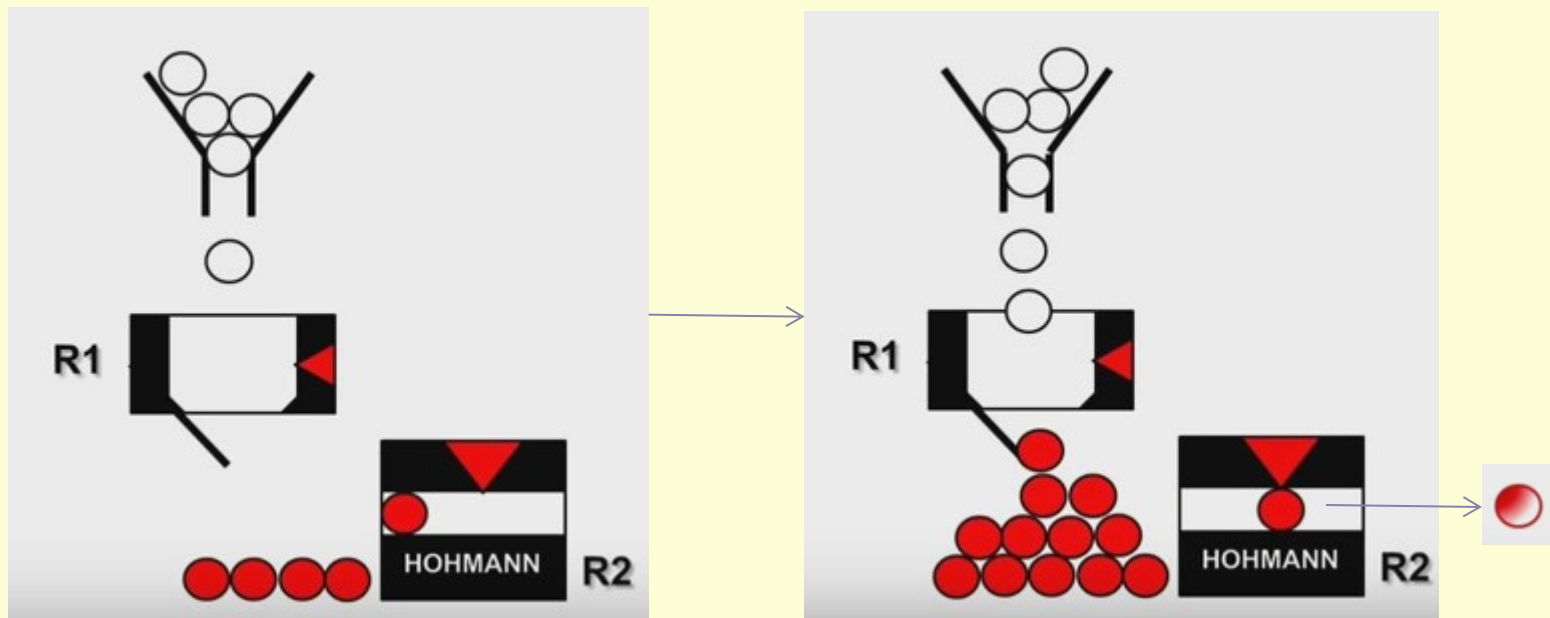


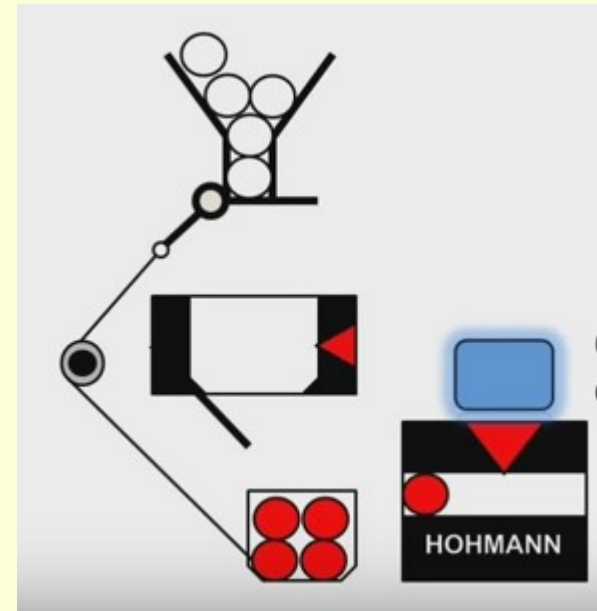
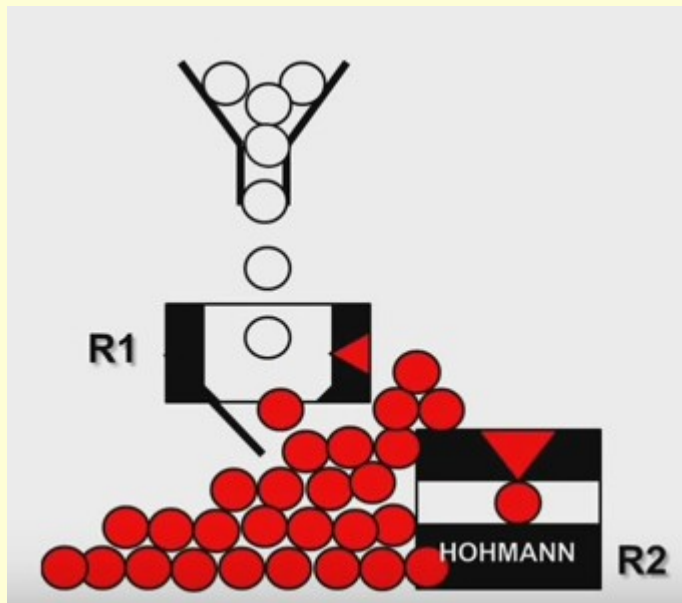
# Drum –Buffer-Rope

Based on : R. Holt, Ph.D., PE

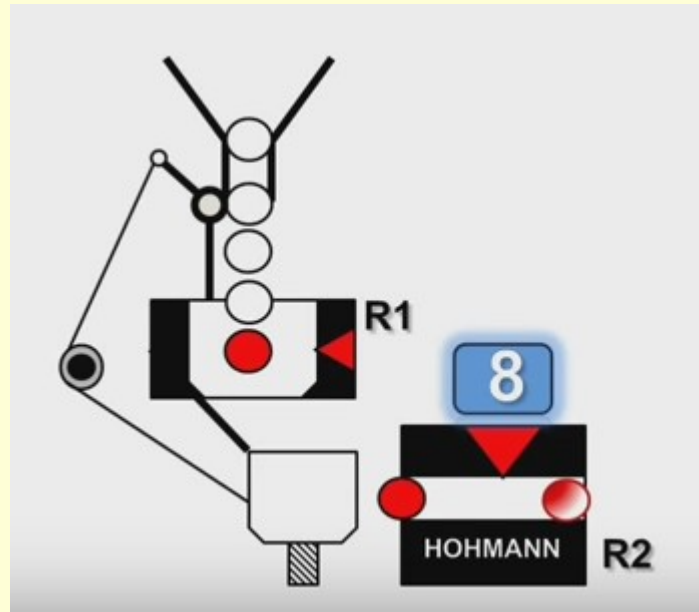
# System not controlled (neregulovaný systém)



# System not controlled and modification DBR

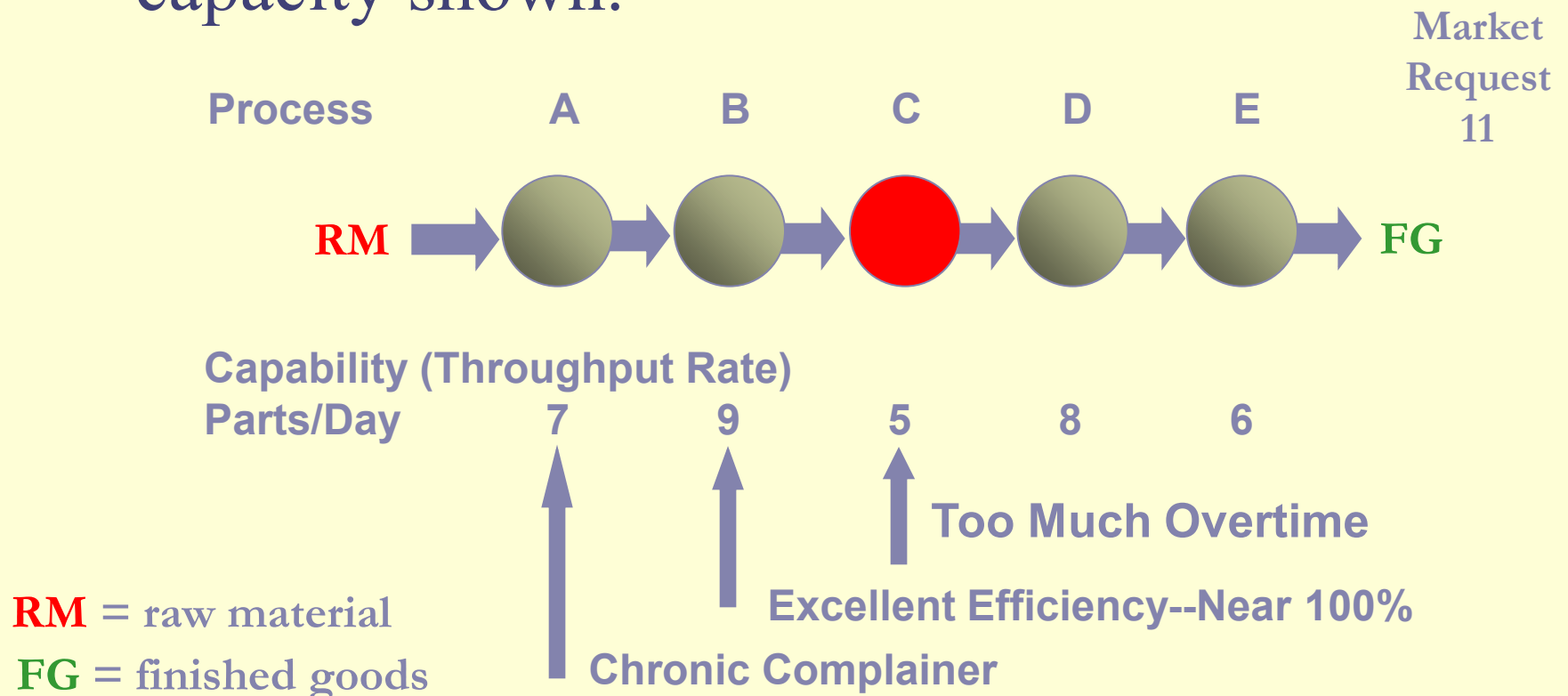


# Rope opened raw material valve



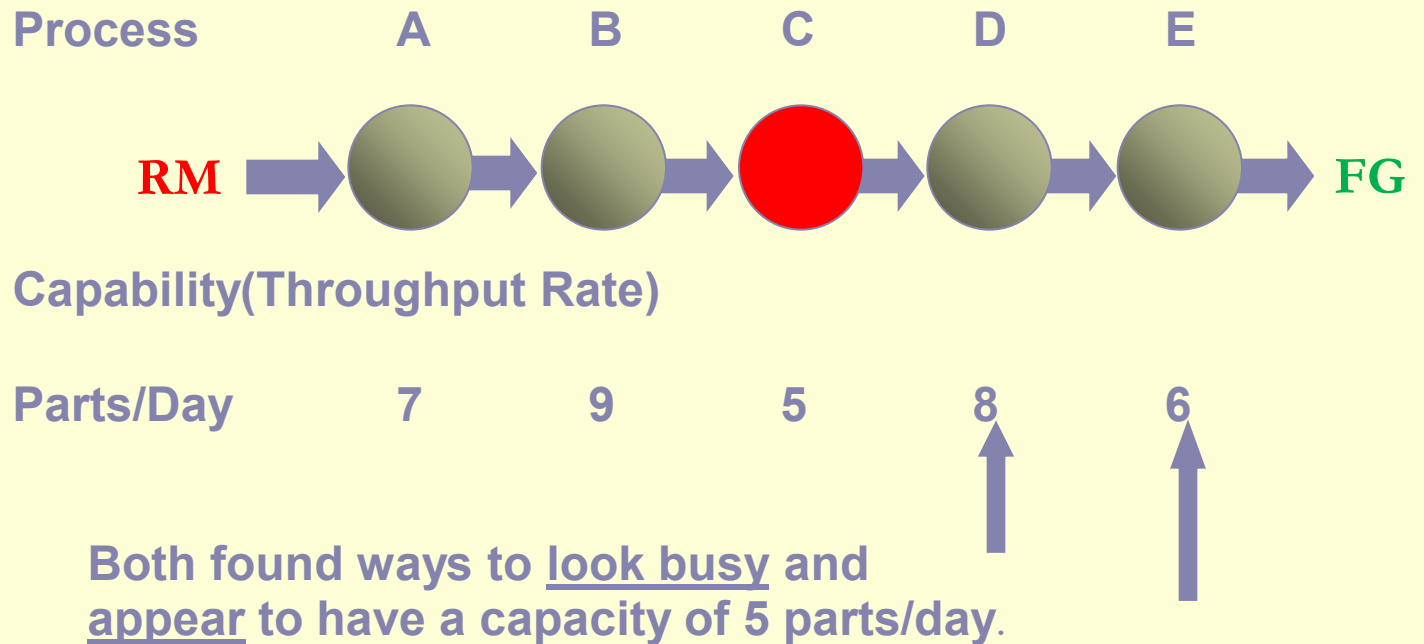
# We Measure Operational Efficiency

- Work flows from left to right through processes with capacity shown.



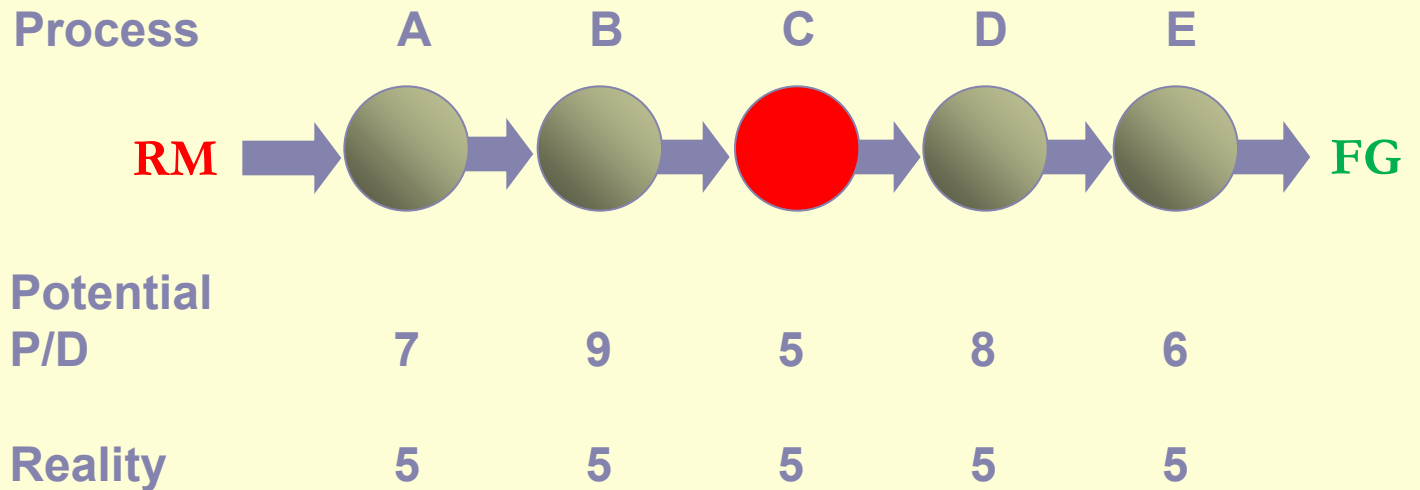
# Reward Based on Efficiency

- Work flows from left to right.



# In reality...

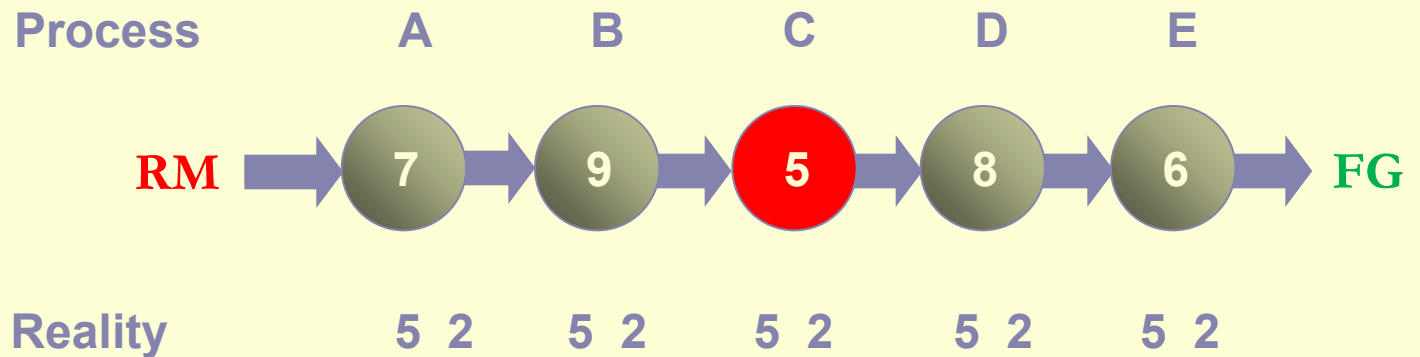
- Processes A and B won't produce more than Process C for long.



P/D=parts/day

# Then Variability Sets In

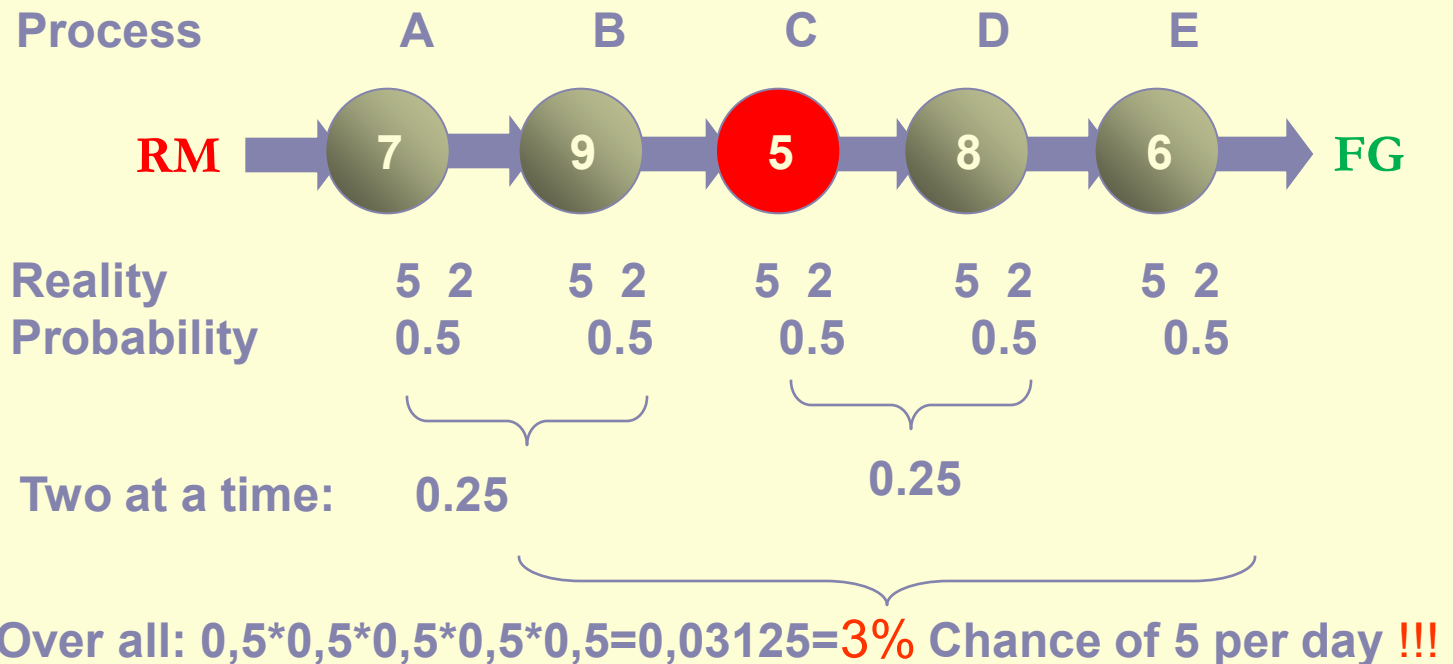
- Processing times are just **AVERAGE** Estimates





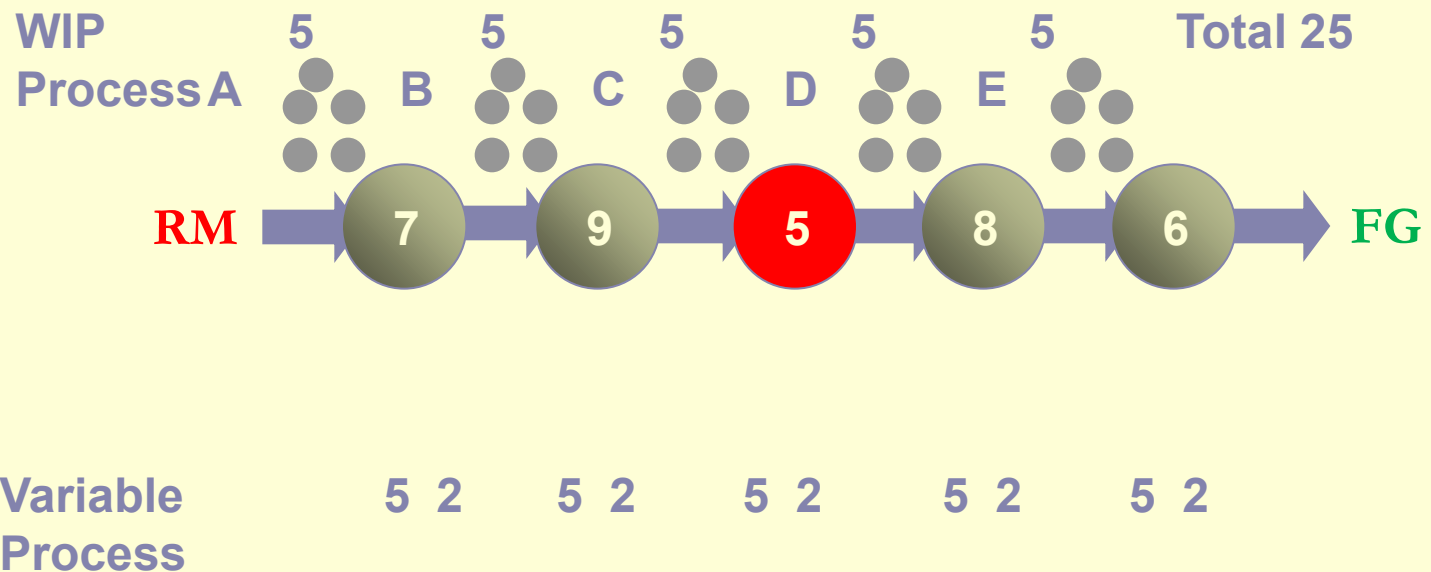
# What's an Average? 50%

- Half the time there are 5 or more per day at each process--Half the time less



# Previous Solution (not good one!): Inventory

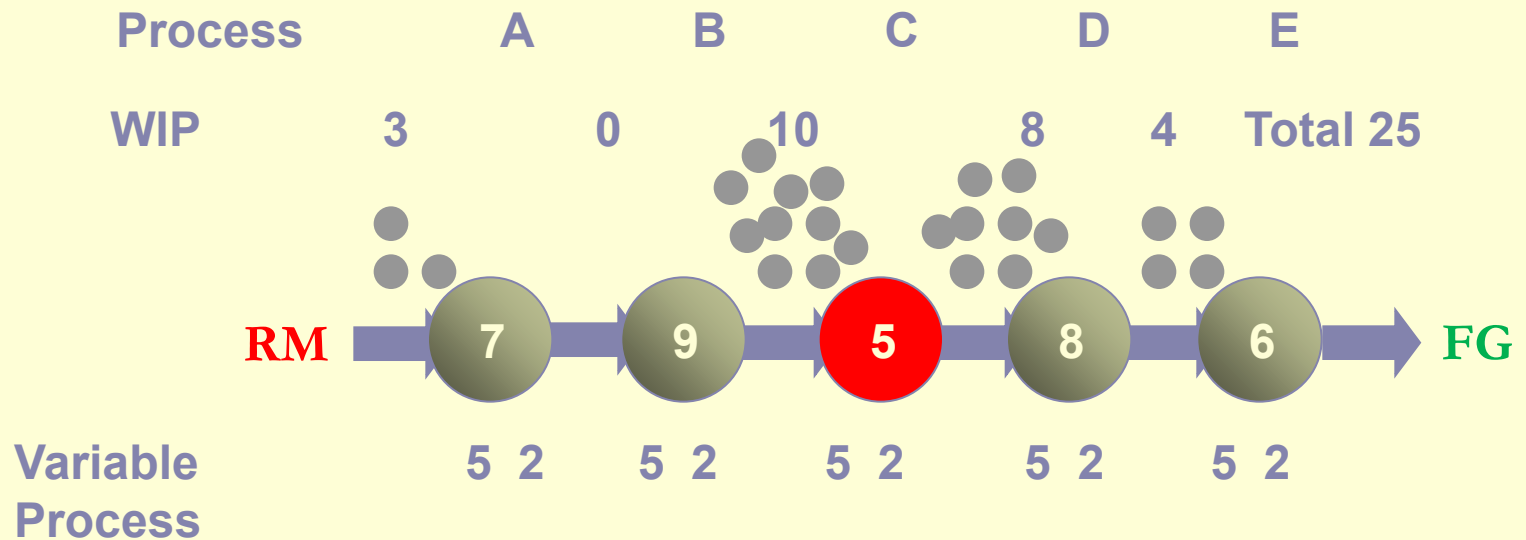
- Put a day of inventory (WIP) at each process!



WIP= Work in Progress = NV=Nedokončená výroba

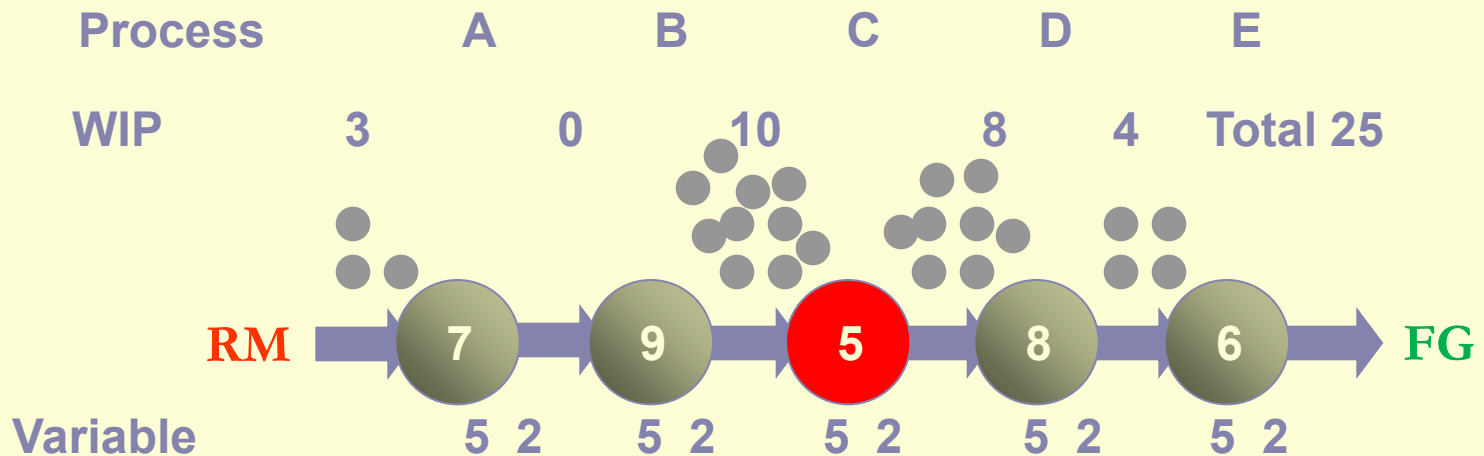
# System Variability Takes Over → Chaos

Inventory (WIP) quickly shifts position.  
Inventory manager tries to smooth it out.  
Distribution problems result. Costs go up !!!



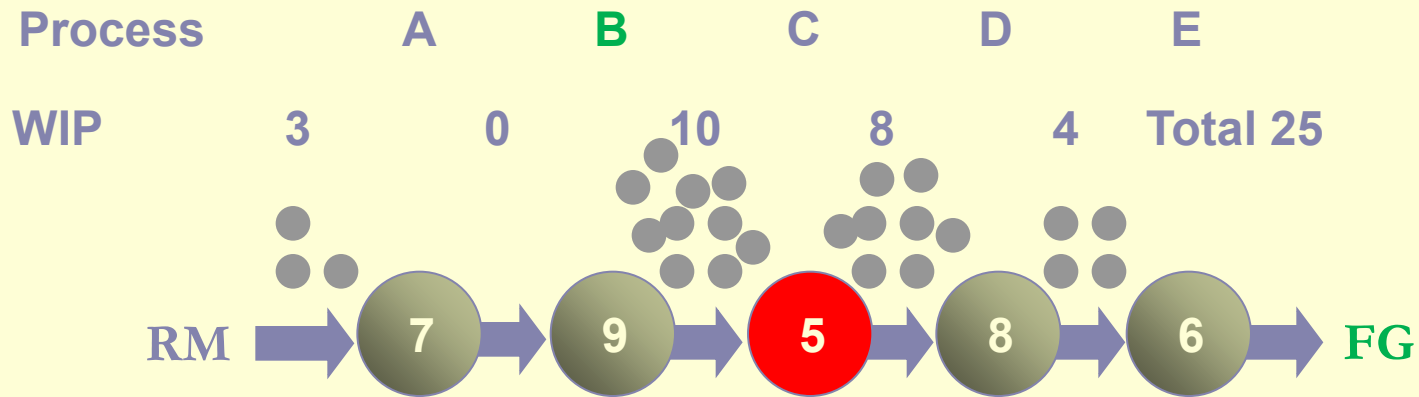
# System Variability Takes Over--Chaos

An Average of 5 means sometimes 3 and some times 7



Shifting work-in-process creates **large queues** at some locations. This makes work wait longer to be processed. (based on Little's law)

# System Variability Takes Over--Chaos

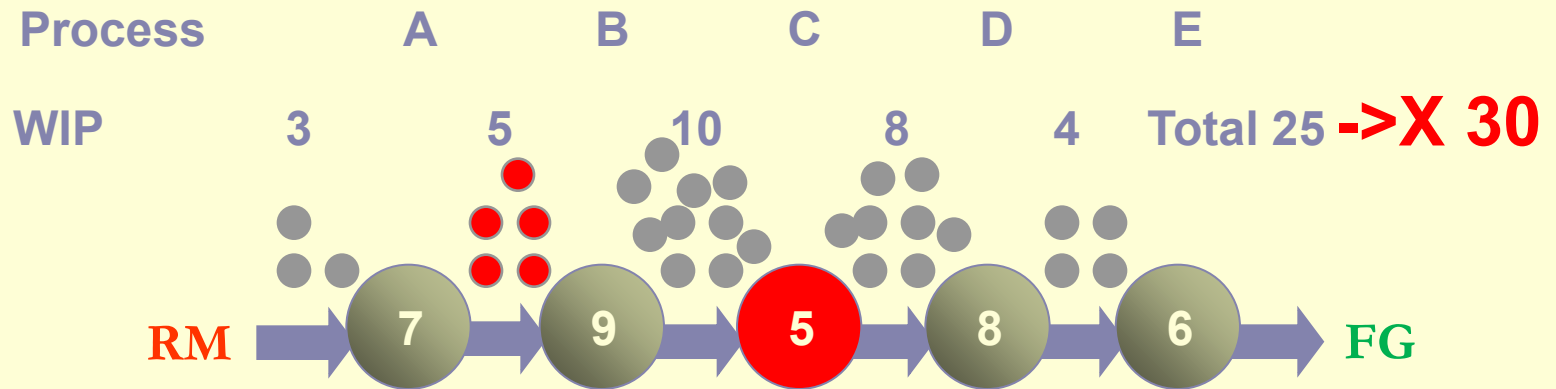


Variable  
Process

Shifting work-in-process creates **large queues** at some locations. This makes work wait longer to be processed.

Other workstations are starving for work (**B**) The work they could do is delayed because they have no input material. They can't take advantage of their extra capability. So..... ?

# System Variability Takes Over--Chaos



Variable Process 5 2 5 2 5 2 5 2 5 2

So... **Management Helps!** Management puts in more work (Inventory) (rate of input RM) to give everyone something to do (Cost world)!  
 Result: It takes longer and longer from time of release until final shipping. **More and more delay!!!!!!!!!!!!!!**



# TOC Steps to Continuous Improvement


**Step 1. *Identify* the system's constraint.**

**Step 2. *Exploit* the system's constraint.**

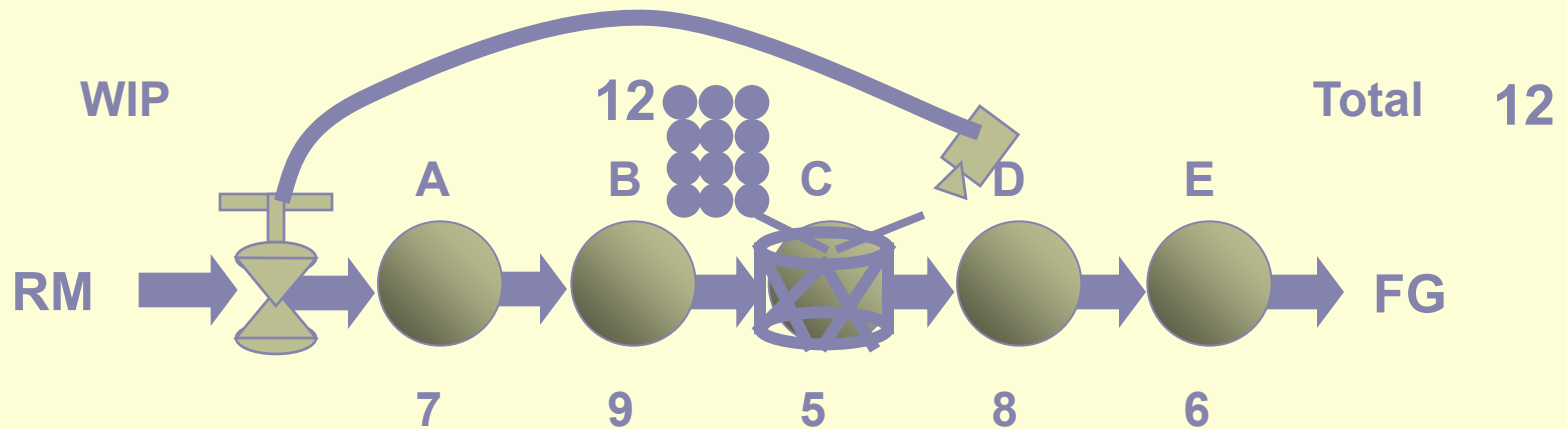
**Step 3. *Subordinate* everything else to the above decision.**

**Step 4. *Elevate* the system's constraint.**

**Step 5. If a constraint is broken (that is, relieved or improved), go back to Step 1. But don't allow *inertia* to become a constraint.**



# Five Steps Applied to Flow Operations



## Five Focusing Steps

Step 1. Identify the Constraint (The Drum) - **CRT**

Step 2. Exploit the Constraint (Buffer the Drum) – **time reserve**

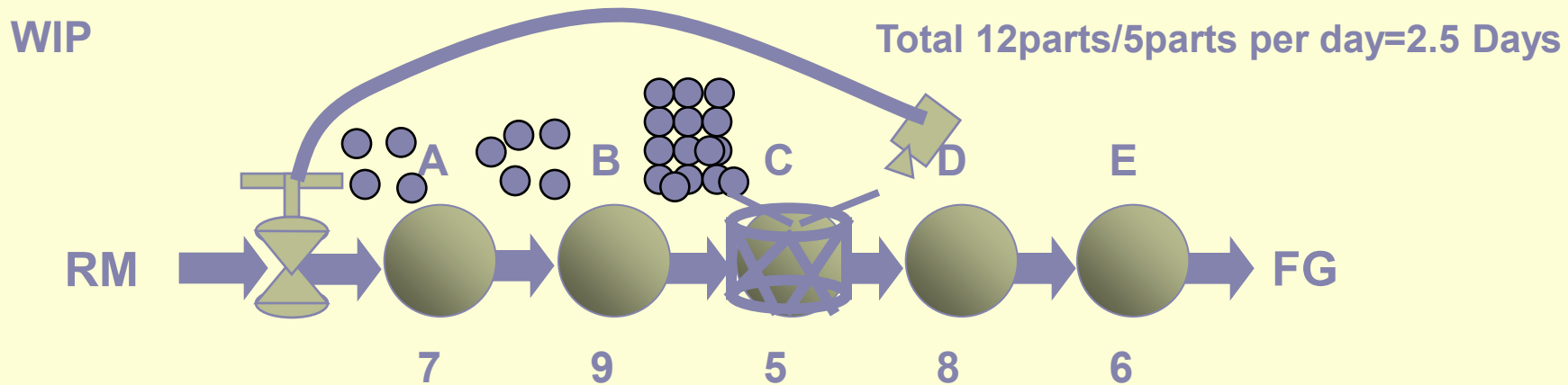
Step 3. Subordinate Everything Else (Rope) - **feedback**

Step 4. Elevate the Constraint (\$?->extra costs)

Step 5. If the Constraint Moves, Start Over

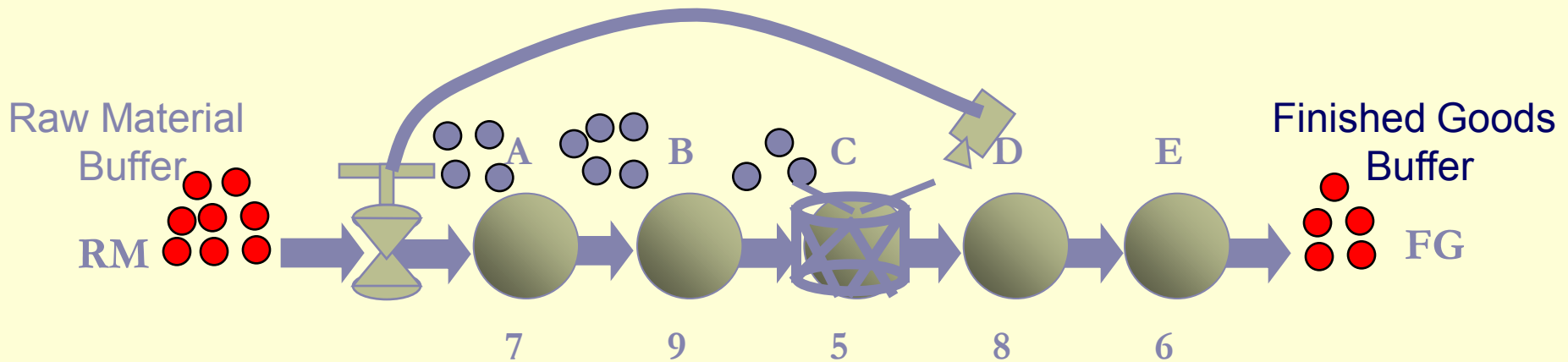


# Understanding Buffers



- The “Buffer” is Time!
- In general, the buffer is the total time from work release until the work arrives at the constraint.
- Contents of the buffer alters (see below)
- If different items spend different time at the constraint, then number of items in the buffer changes
- but **Time in the buffer remains constant.**

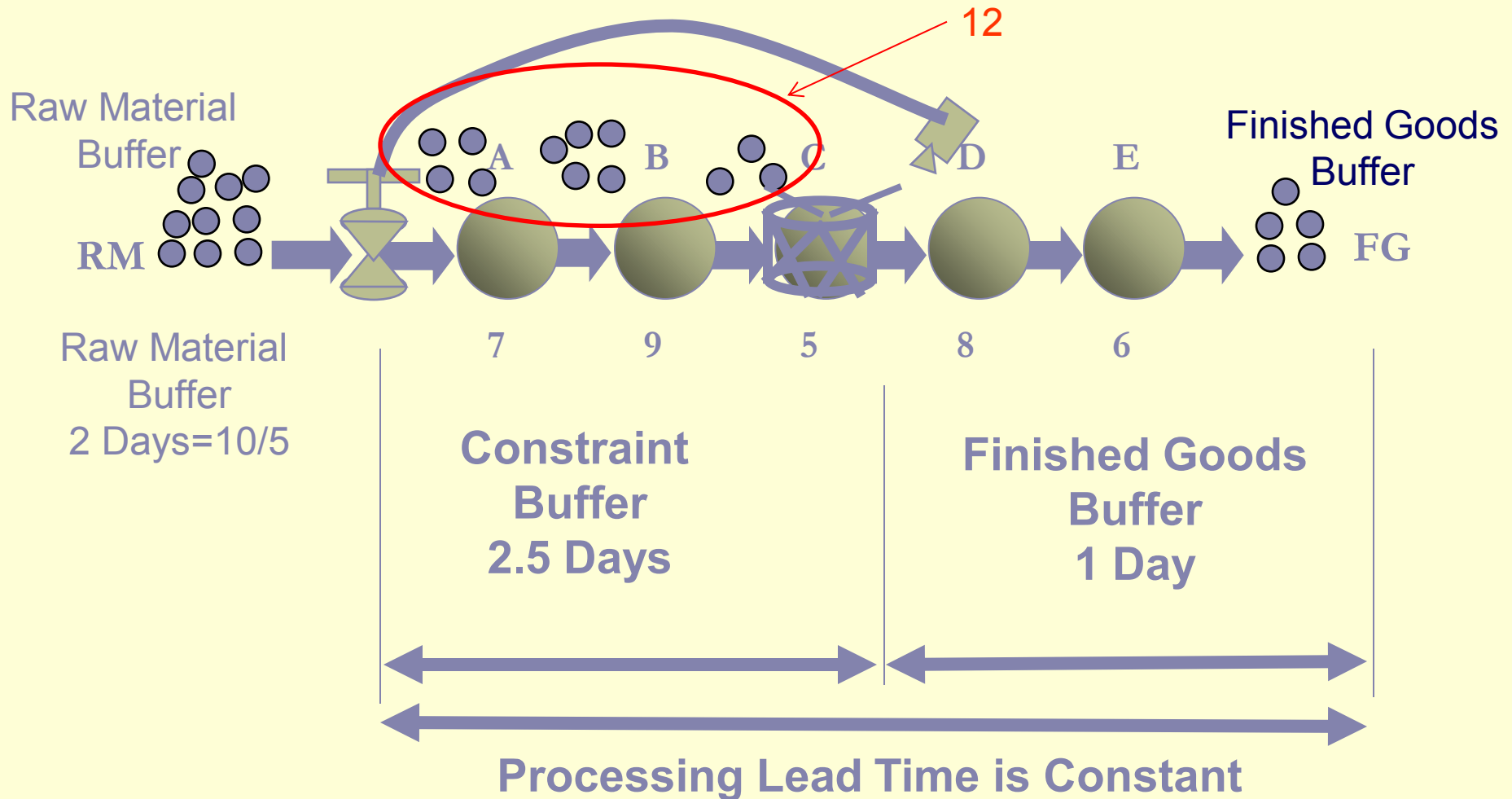
## We need more than one Buffer



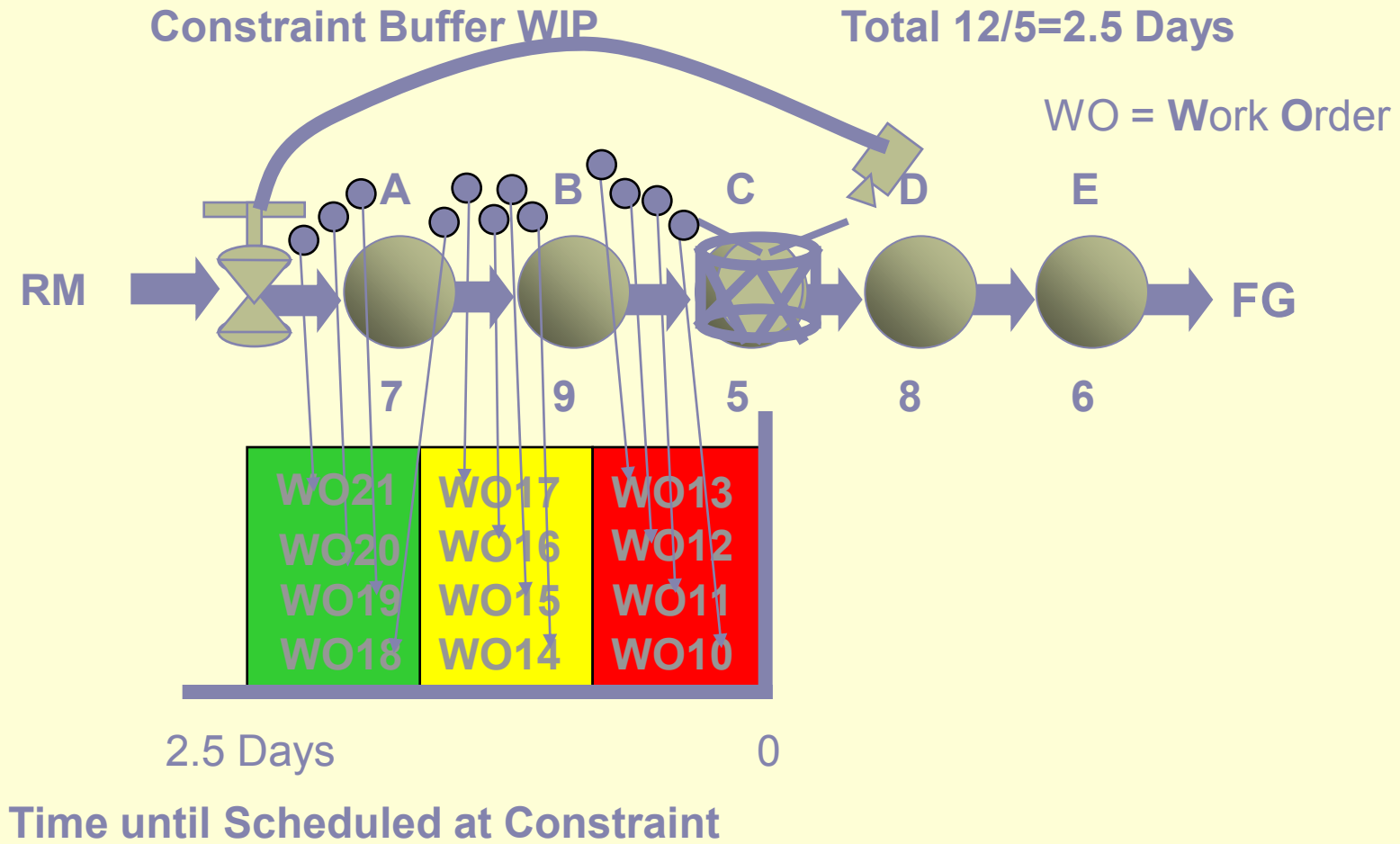
There is variability in the Constraint.  
To protect our delivery to our customer we  
need a finished goods buffer.

There is variability in our suppliers.  
We need to protect ourselves from unreliable  
delivery.

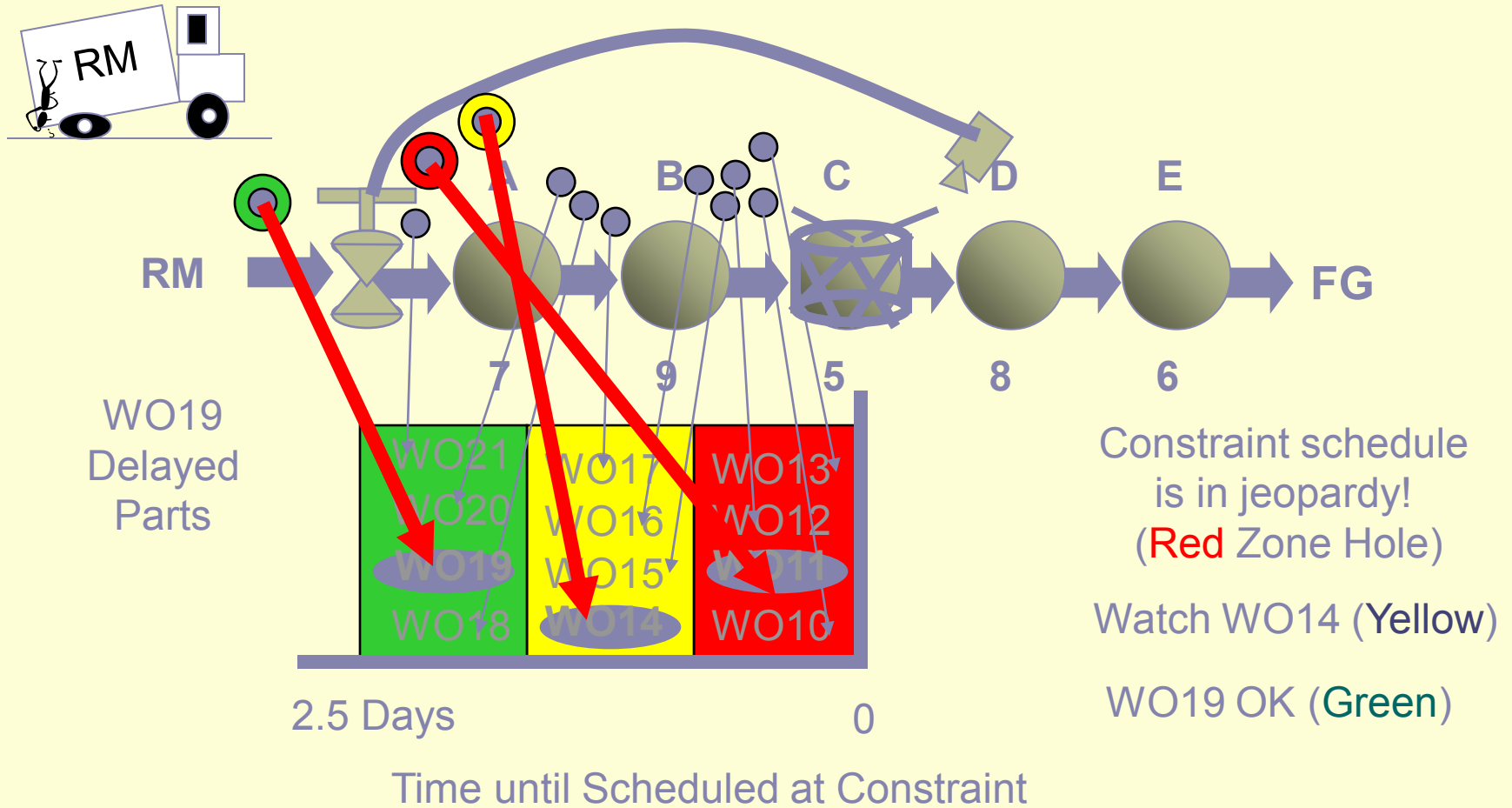
# Buffer Time is Constant-Predictable



# Buffer Management



# Problem Identification



# Additional Buffers

- Constraint Buffer (as we discussed)
  - Protects the Constraint from running out of work
- Finished Goods Buffer
  - Protects customer delivery from Constraint variation
- Raw Material Buffer
  - Protects the Release of material from suppliers
- Assembly Buffer
  - Facilitates speedy flow of products

**See interesting video**

<https://www.youtube.com/watch?v=8yehd2ZsKH0>

