Processes Designed to Reveal Problems Immediately – As They Occur

SELF-DIAGNOSTIC PROCESSES

PDCA = The Scientific Method

This is DO and CHECK

- Carry out your plan, or run your process *exactly as you planned.*

- Carefully observe the result.
- Compare the actual result to your planned result.

PROBLEM:
If the *process does not watch itself,* Then *people must be vigilant.*

Where this all began

Don't waste resources making defects.

Continuously verify "what is" to "should be."

Immediately signal any problem.

A look at a modern loom…

So far nobody has found a better way to detect a broken warp thread.

Where this all began – Gregg’s Insight

“I realized all of this started with a son trying to make life easier for his mother.”

Gregg Horne
Terex
Motherwell, Scotland
Debrief comments after a visit to the Toyota museum in Nagoya.

Question:

How does quickly revealing problems result in a job that is *easier to do?*

Question:

How does quickly revealing problems demonstrate *respect for your people?*

Question:

How does quickly revealing problems help *challenge* people in ways that *build their confidence and capability*?

Revealing Problems With 5S

5S *continuously* asks and answers two very simple questions:

- **What is needed to do this work?**
- **Does the team member have it?**

PLAN	**Specify what is needed.**
DO	**Remove everything else.**
CHECK	**Visual Controls**

Everything here.
Where it belongs.
Nothing extra.

Revealing Problems With 5S

Original Problem:

"Standard" router bit was replaced with "non-standard" bit to run a specific program.

The standard was <u>not</u> restored before the next program was run, resulting in a scrapped part.

Proposed Countermeasure:

A list of the locations and standard bits that belonged in them.

Revealing Problems With 5S

- What result do you expect from that step?
- What do you expect to see?
- How will you know?

What, *exactly* do you want the operator to do?

Revealing Problems With 5S

Why is this better?

- **What result do you expect from that step?**
- **What do you expect to see?**
- **How will you know?**

So what is *the problem* here?

Do we even know what the problem is?

Or is this just *<u>evidence</u> of a problem.*

~~Revealing~~ *Creating* Problems With 5S

What is the problem with this approach?

Video produced by Mark Grabon

The Flow of People

The Standard (what to strive for):

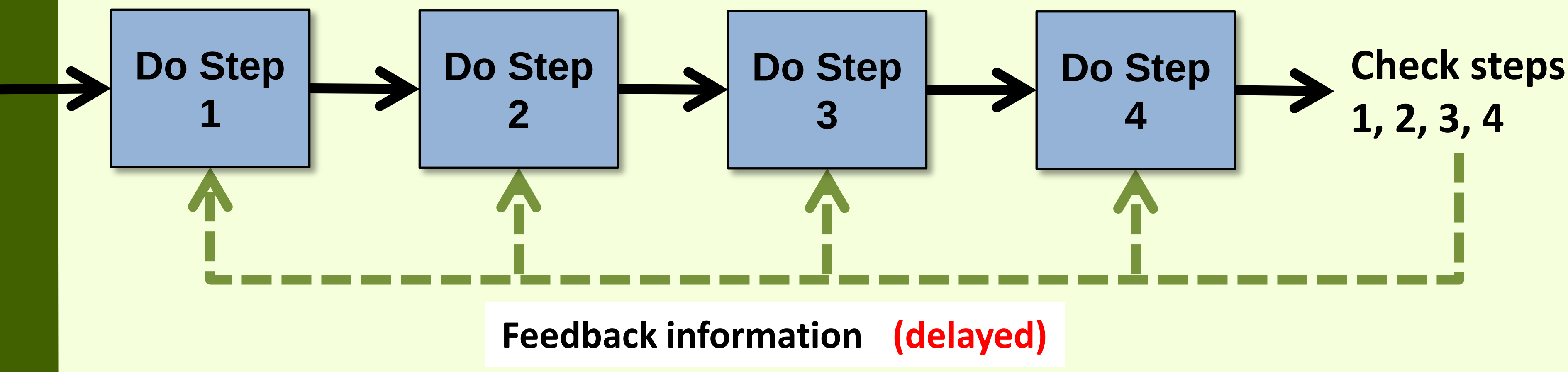
- All work is specified in terms of *content, sequence, timing and outcome*.
- The work contains *built-in-checks* <u>at each step</u> that compare the actual work against what was expected.
- The work contains *build-in-checks* <u>at each step</u> that compare the actual outcome against the expected outcome.
- Any deviation from the standard triggers immediate response and problem solving.

The Importance of Frequent Checks

Why is it important to have *frequent* checks of actual vs. planned work steps?

What is the consequence of waiting until the end of a major work flow?

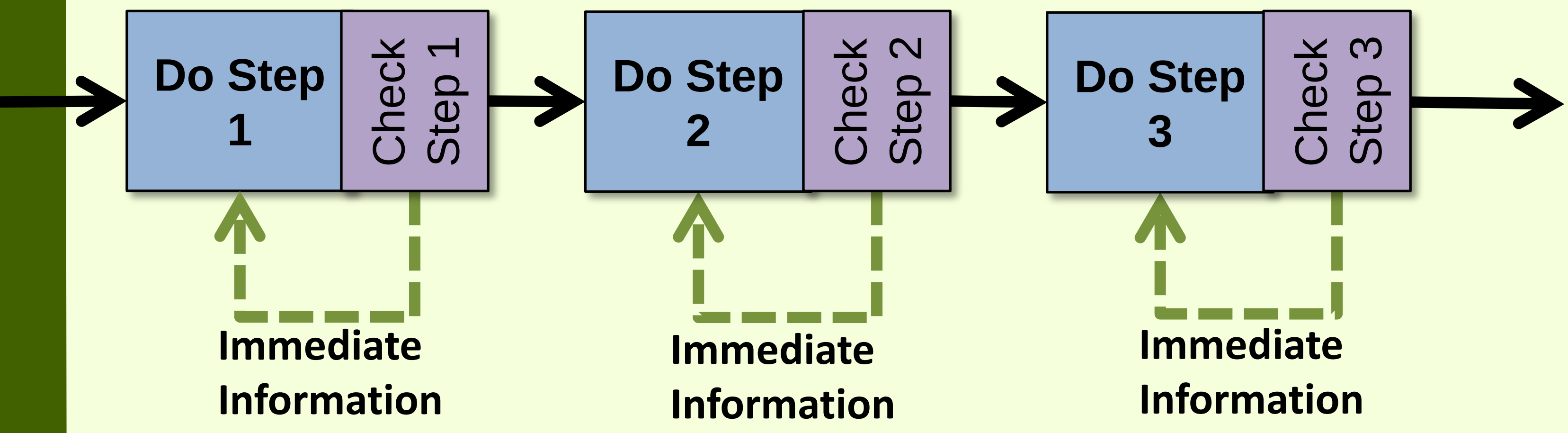
The Importance of Frequent Checks



Where did the problem occur?

If there is time / distance between the steps – When did the problem occur?

The Importance of Frequent Checks



Where did the problem occur?

Known right away.

When did the problem occur?

Known right away.

From *The Toyota Production System: An Example of Managing Complex Social/Technical Systems,* 1999, Steven J Spear

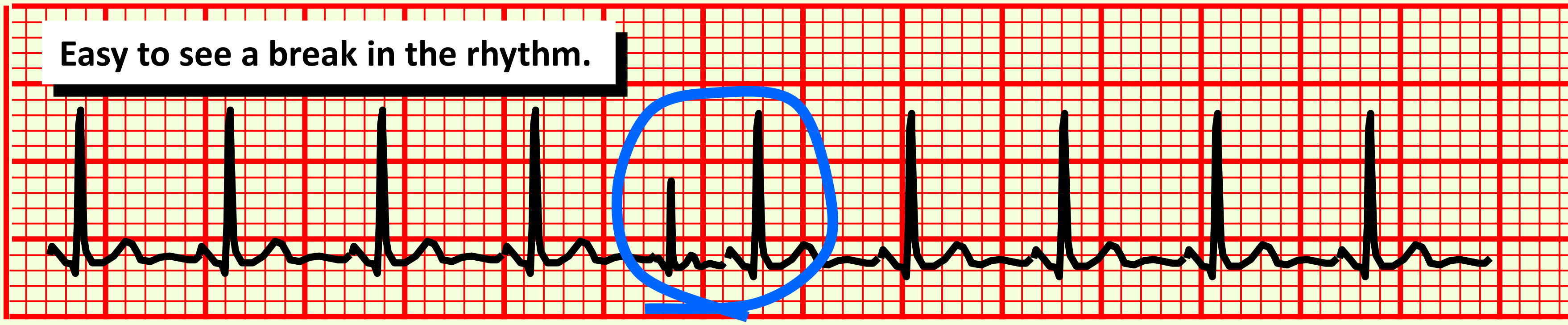
Remember the Marshmallow Challenge

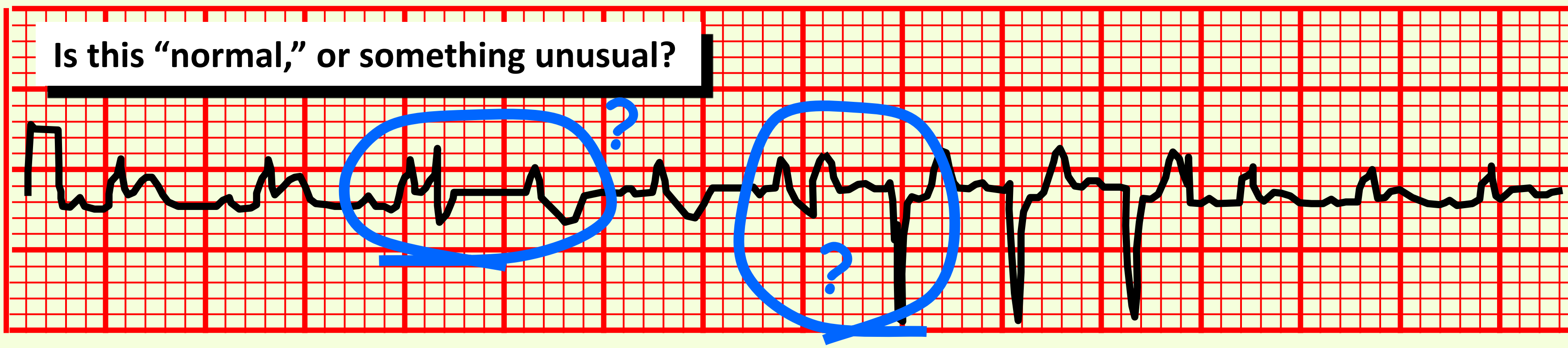
Learning occurs with frequent cycles of

- **Try**
- **Check**
- **Fail**
- **Reflect**

Frequent checks help the organization learn to “build taller structures and keep the marshmallow on top.”

Pacing as a Diagnostic Tool





Takt Time

Cycle Time

What *Should* Be Happening

What is *really* happening.

To reveal problems:

Continuously compare the actual pace vs. the specified pace.

Takt Time

Cycle Time

What *Should* Be Happening

What is *really* happening.

To reveal problems:
Continuously compare the actual pace vs. the specified pace.

Other Examples of Frequent Checks

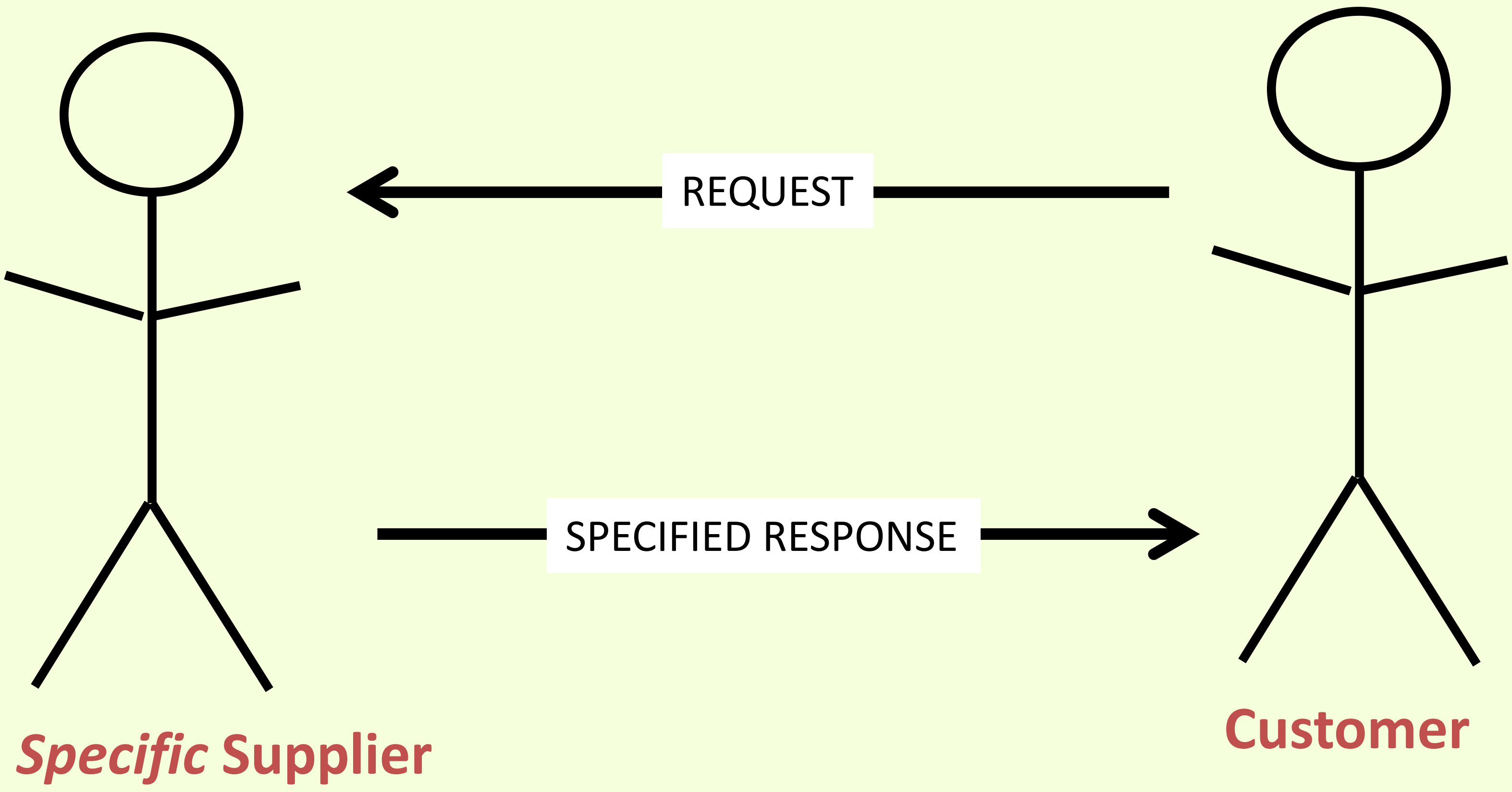
- Mistake Proofing (“Poka-Yoke”)
  - Quick, easy check, every work cycle.
- Setting parts / tools / machines in sequence.
  - Verifies actual usage sequence vs. planned.
- Kanban
  - Verifies order / receipt each cycle.

The Flow of Information

The Standard (what to strive for):

- All information flows are binary (yes/no; need/don't need) and *direct* from the requester (customer) to the provider.
- The signal triggers a specified response.
- There are *built-in-checks* .
  - Verify that the signal was given as specified.
  - Verify that the response was as specified.

Direct, Binary Signals

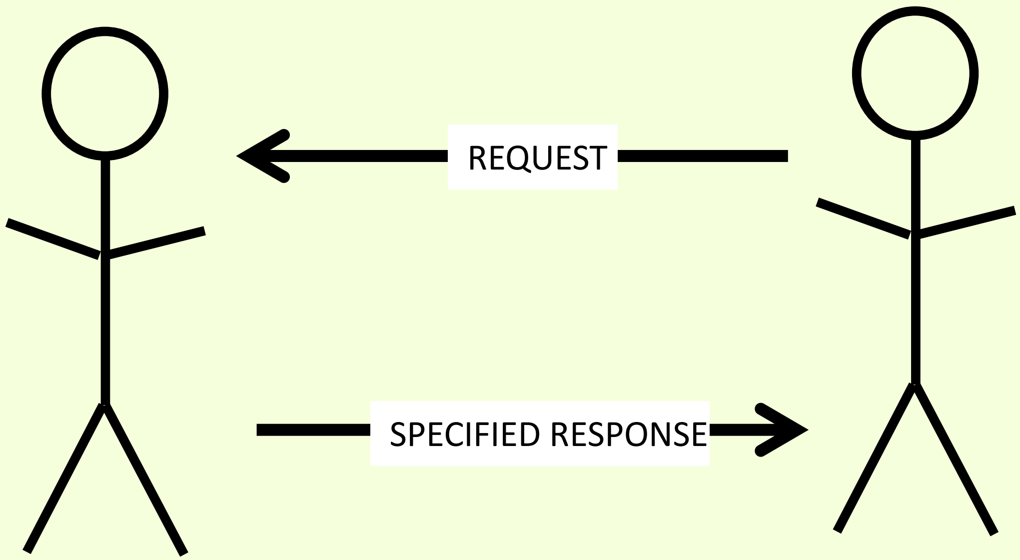


Built-in CHECKS

Did the appropriate signal trigger the response?

Was the response as specified?

Was the response early, on time, late?



Specific Supplier

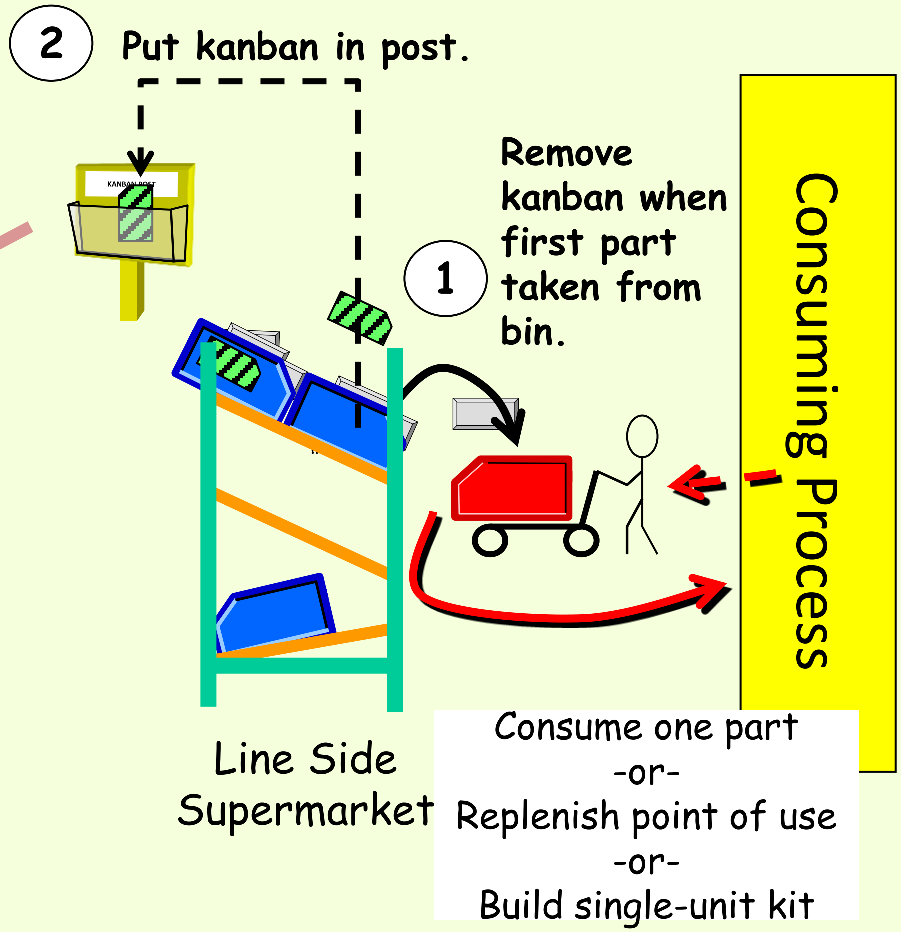
Customer

Was the request initiated as expected?

When expected?

Kanban: Direct, Binary Connections

A clear, binary, unambiguous signal:
- **Please deliver a part of this type,**
- **Packaged this way,**
- **To this location,**
- **By this time.**



What are the indicators?

Shortage or supply drops below safety stock level. Demand outstrips supply.

Signals are coming ***faster*** than expected.
Demand is ***greater*** than planned.

OR

Supplier process (cycle time) is slower than expected.
Supply is less than expected or required.

We *do not know* which of these conditions is true without further investigation. GO AND SEE!

"Direct from Customer to Supplier"

- Why is it critical to have a *specifically designated* supplier receive the signal?

“Direct from Customer to Supplier”

- Why is it critical to have a *specifically designated* supplier receive the signal?

  Who you gonna call?

You can’t say “It didn’t work” until:

- **You tried it.**
- **You verified.**
- **You studied and understand why.**

- **Must be able to verify:**
  - **Did the signal get to the right person?**
  - **Was that person able to respond?**
  - **Was the response as expected?**

MAX 15 MIN 5
CHASSIS 3
SHO 1 2 3 4 5 6
7 8 9 10 11 TIRE
EQU ENT COV EXI
TEMP
L 1 2 3 4 5 6
7 8 9 10 11 340
F-2 INS 350 KANBAN F-1 SHO
TRIM 3
6 5 4 3 2 1 SHO
FUL 12 11 10 9 8 7
TEMP
EXI COV ENT EQU
IP
T3E-15 5

F4
CHASSIS 1
CODE 1
5 4 3 2 1 SHO
FULL
8 7 6 AVI
TEMP F3 AIR PRESS EXIT CONV CONV ENT CONV EQU
F2 F4 C5 C4 C3 C2 C1
KANBAN CALL PART SHORT 4 3 2 1

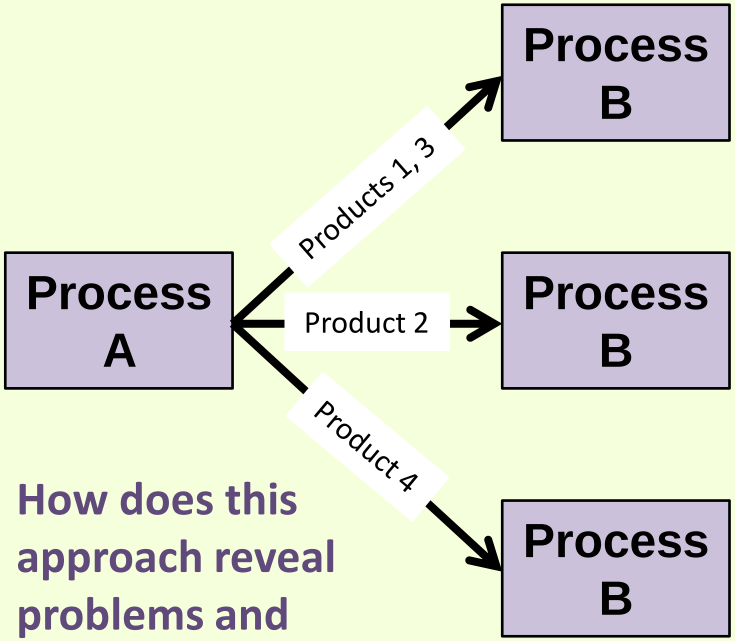
The Flow of Materials (and services)

- **The Standard to strive for:**
  - **The path a product follows is specified, simple and direct.**
  - **If a product actually follows a path *other than* specified, a problem is signaled.**

A problem triggers an *immediate response*.

A *Target:* Simple and Direct Paths

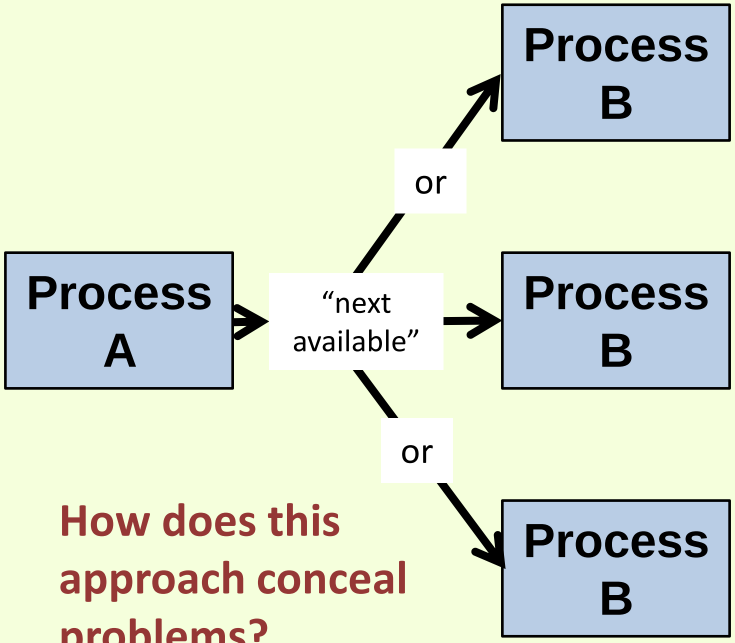
What you are striving to achieve.



How does this approach reveal problems and promote learning?

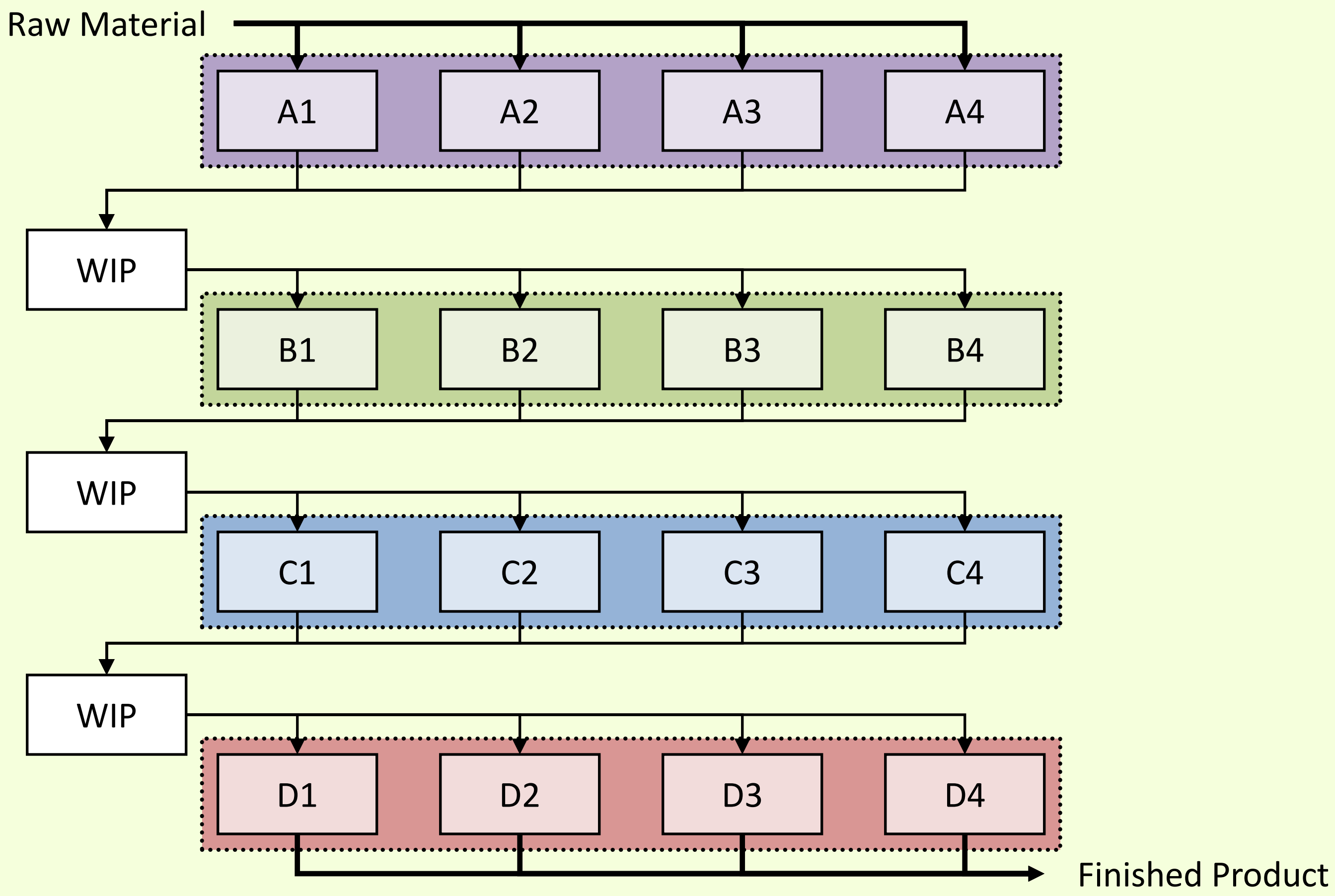
What are the “Ta-Da!” and “Oh-Oh!” moments?

What you are striving to avoid or eliminate.

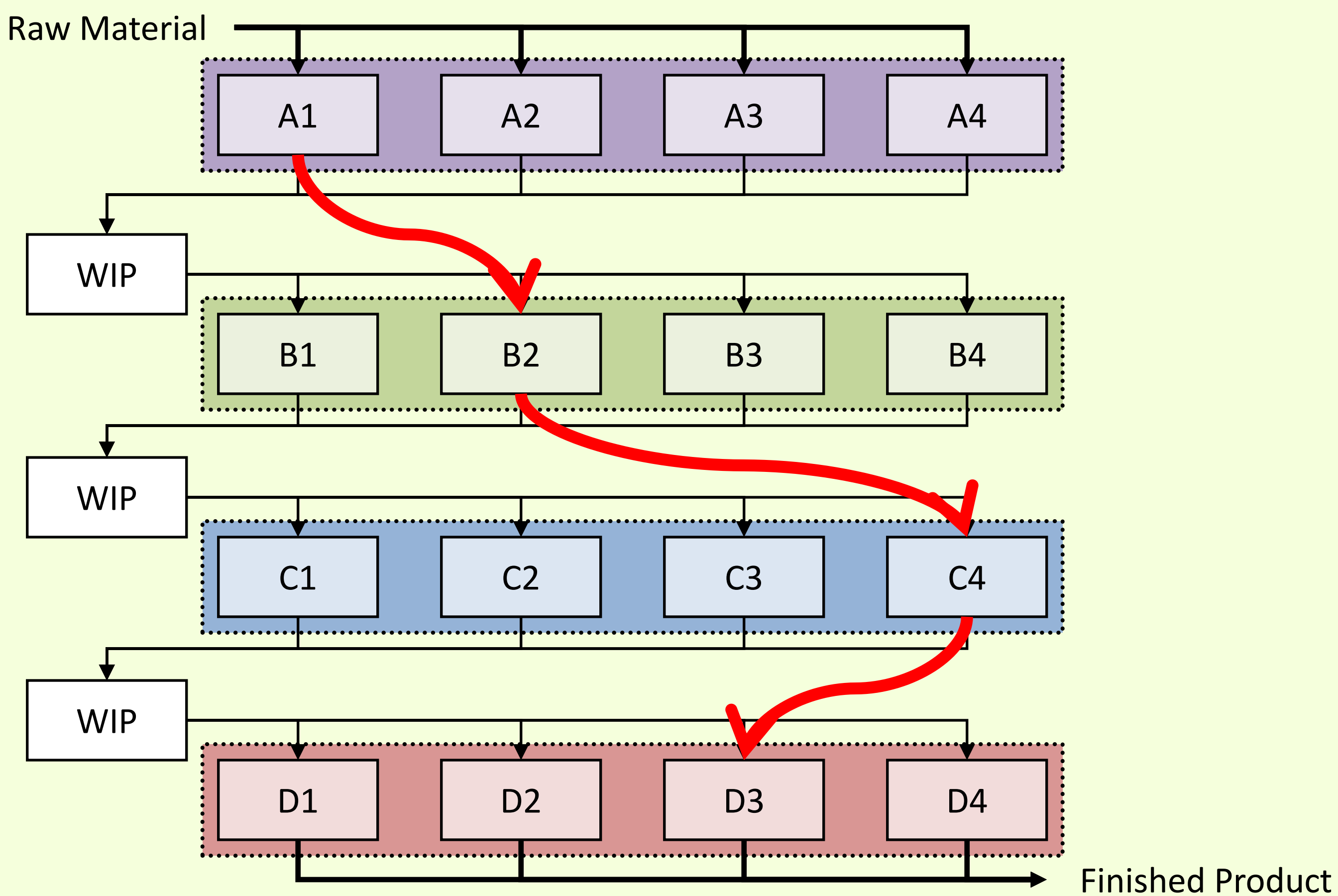


How does this approach conceal problems?

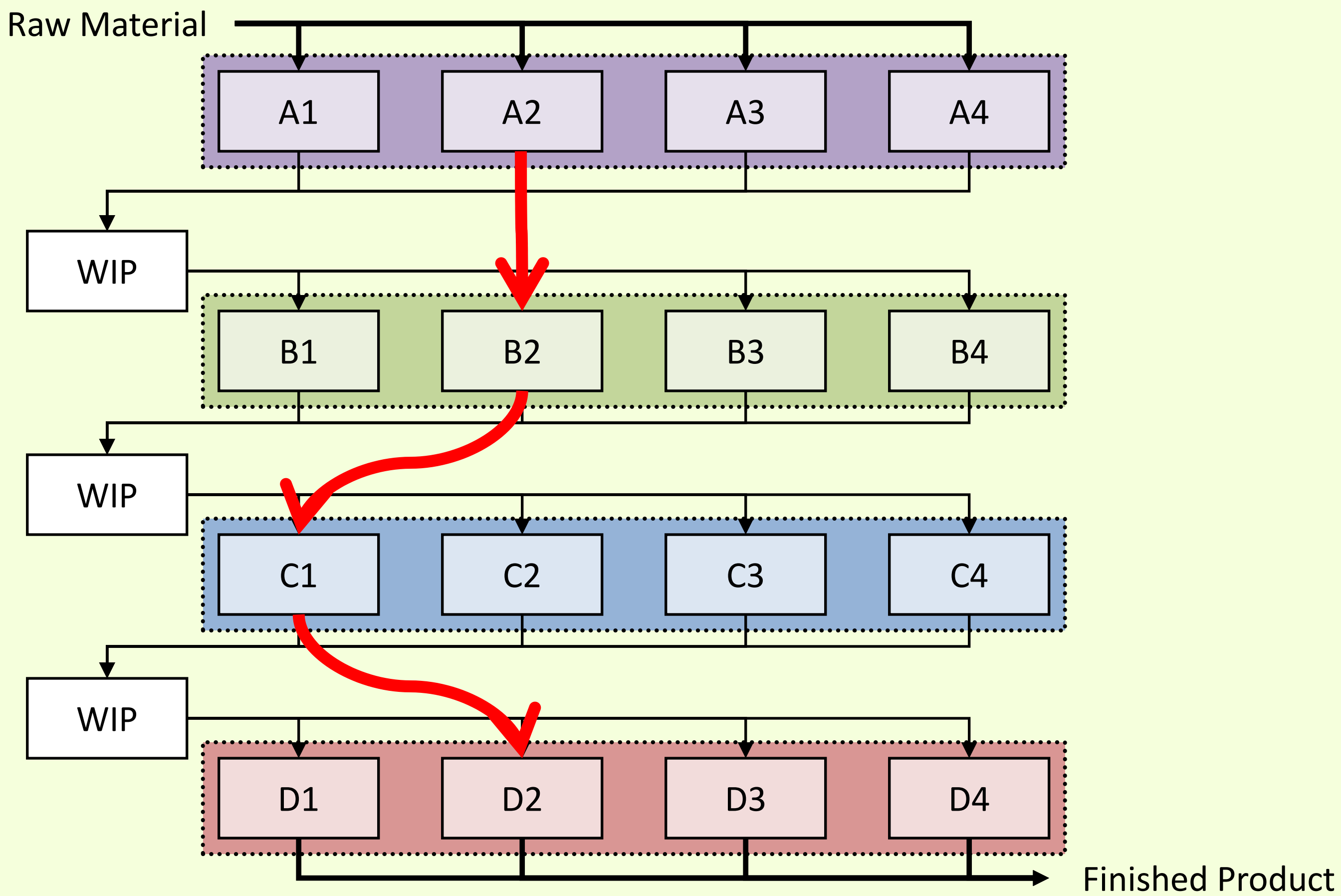
“Any Available Machine or Tool”



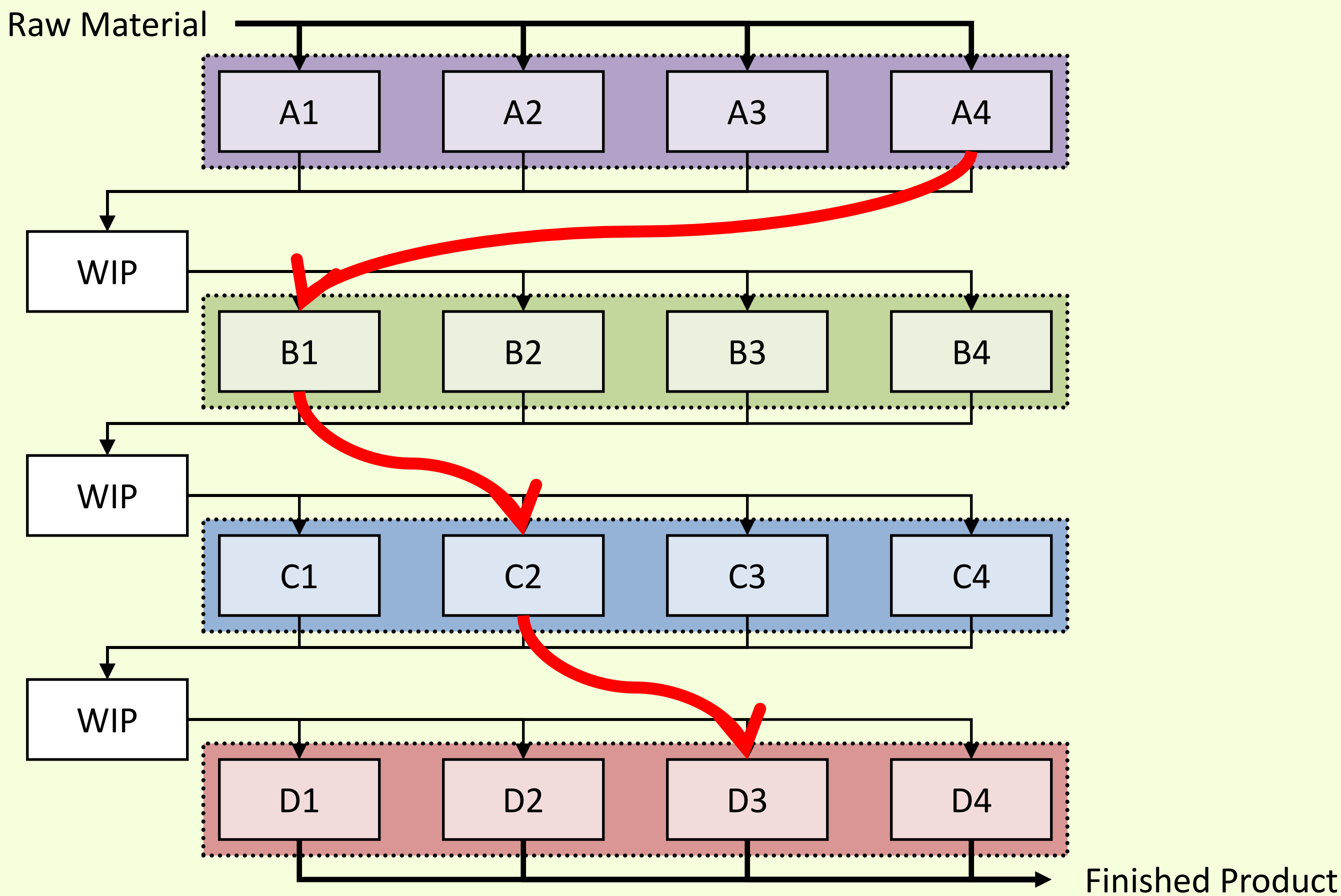
“Any Available Machine or Tool”



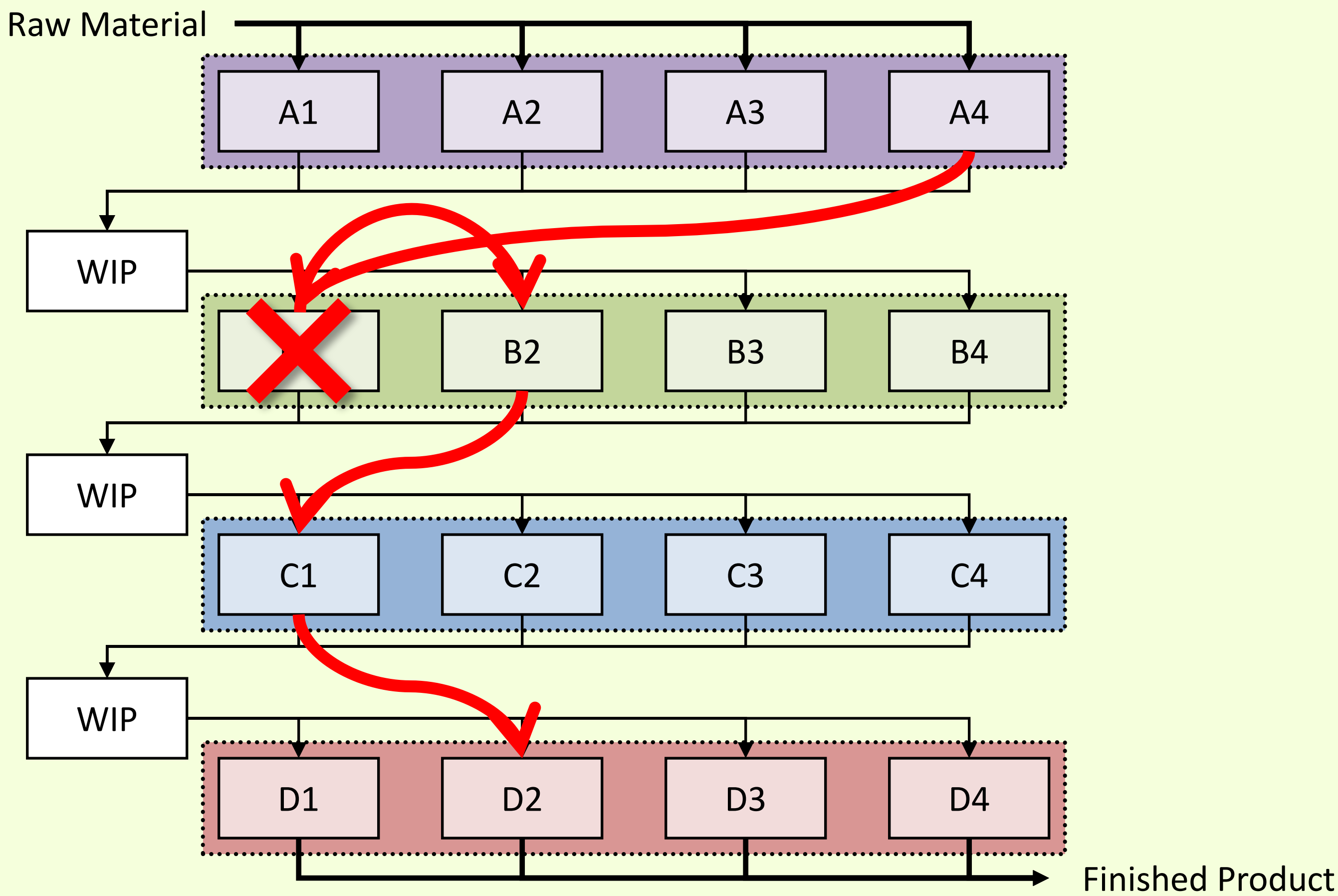
“Any Available Machine or Tool”



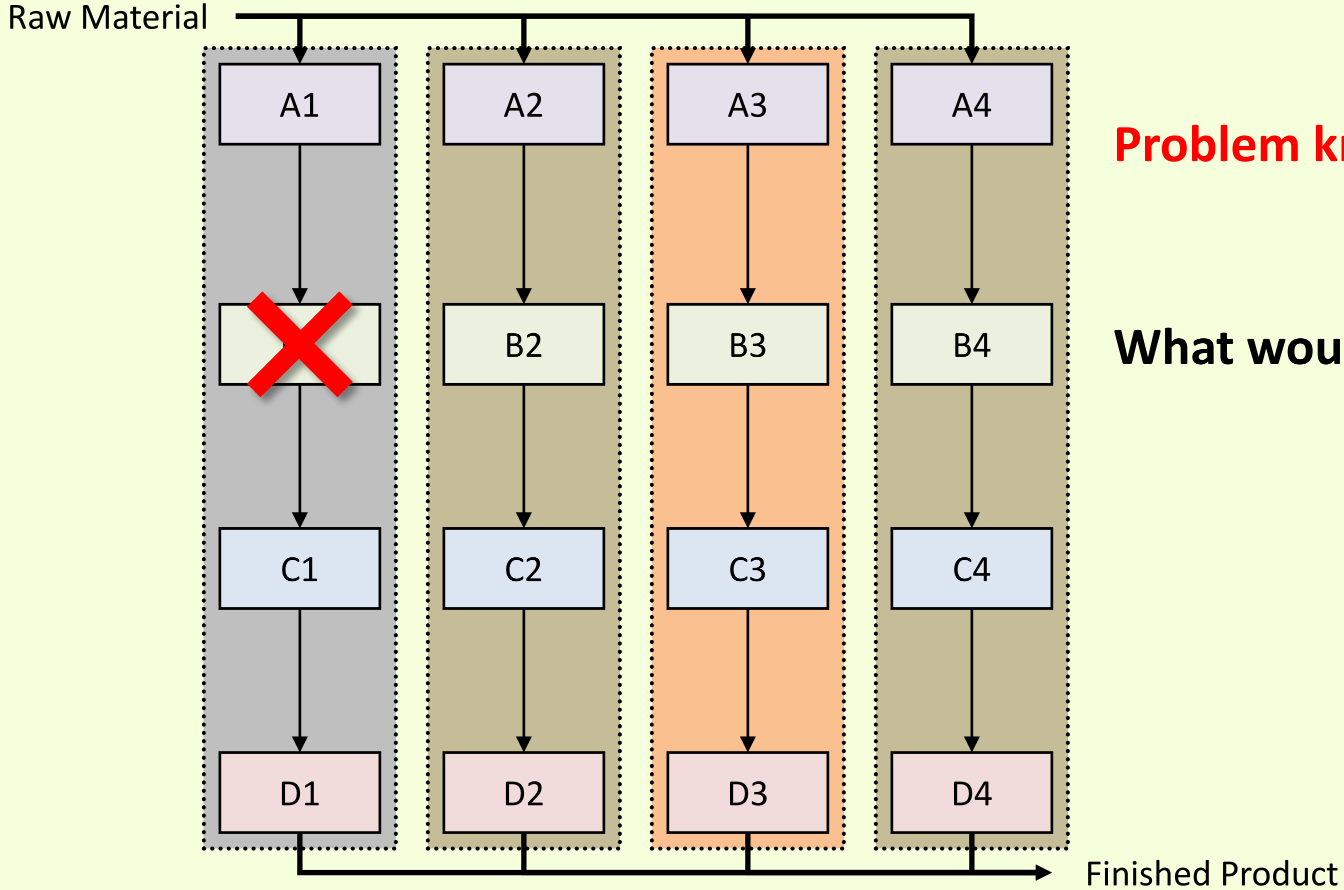
“Any Available Machine or Tool”



“Any Available Machine or Tool”



Simple, Direct Pathways



Problem known immediately.

What would be the response?

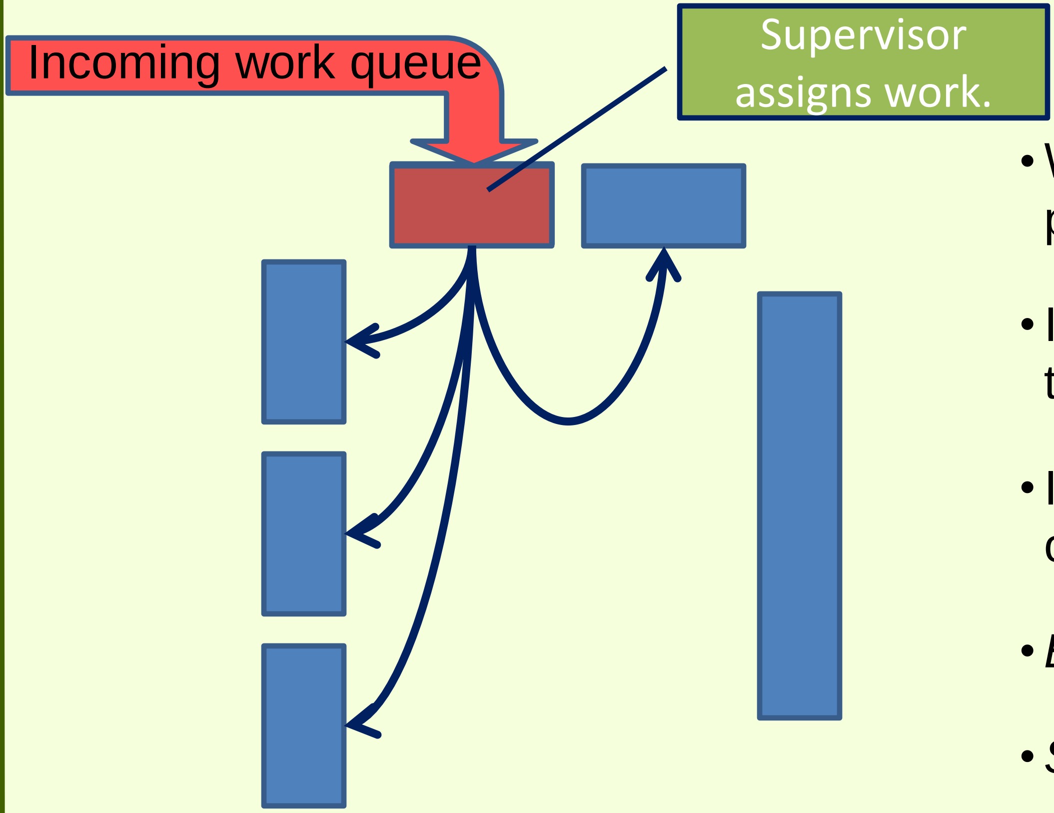
Flow Paths: Discussion

- What happens if the intended flow path *can't* be followed?
- What does that tell you that you didn't know?

Question:

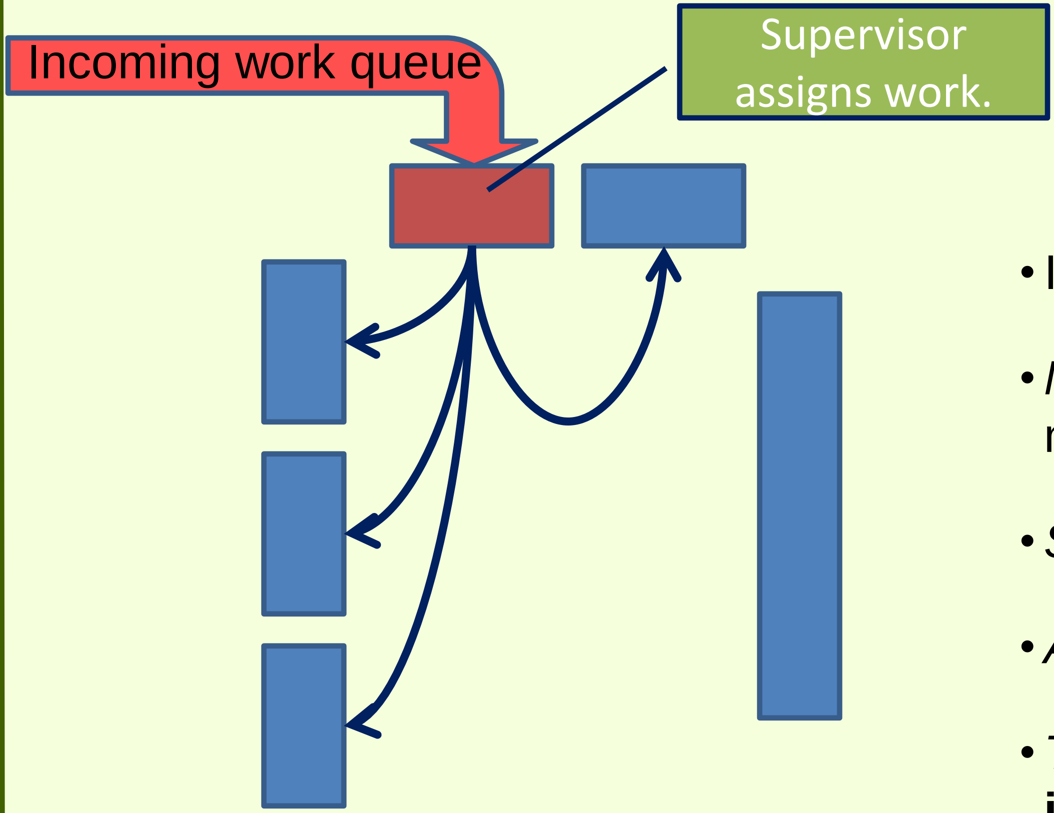
What about an administrative process?

An Actual Administrative Work Flow



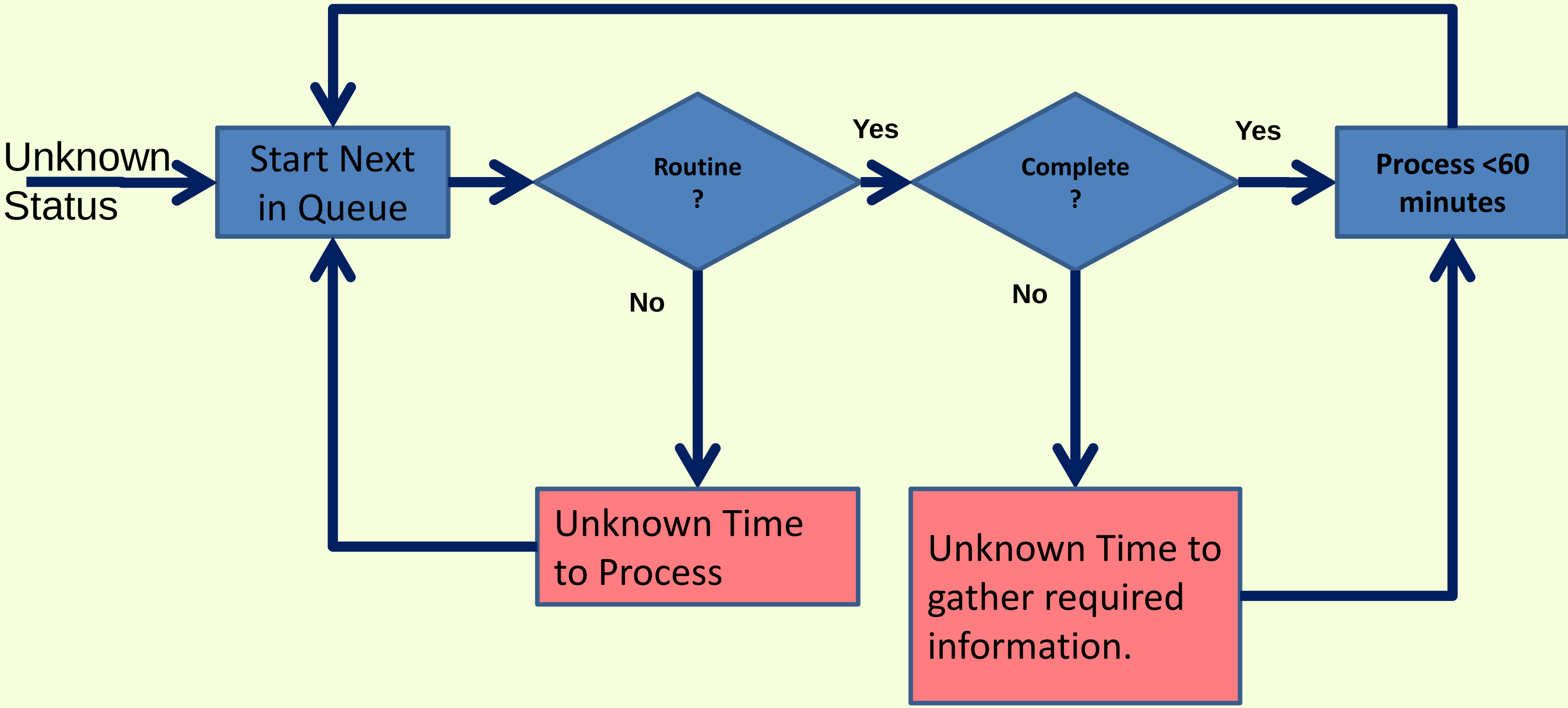
- Work assigned to “next available” person.
- Individual workers responsible for their queues.
- If behind, can ask for help or transfer of work.
- *Everybody* is behind.
- *Saturdays at the office.*

An Actual Administrative Work Flow



- Incoming load 7-10 / day.
- *Most* could be processed in < 60 minutes. (~ 70%)
- *Some* took a few hours.
- *A few* took a few days.
- *Thus the **output cycle*** **was irregular.**

The Problem:



= Unpredictable Throughput

= High variation in output cycle

The Result:

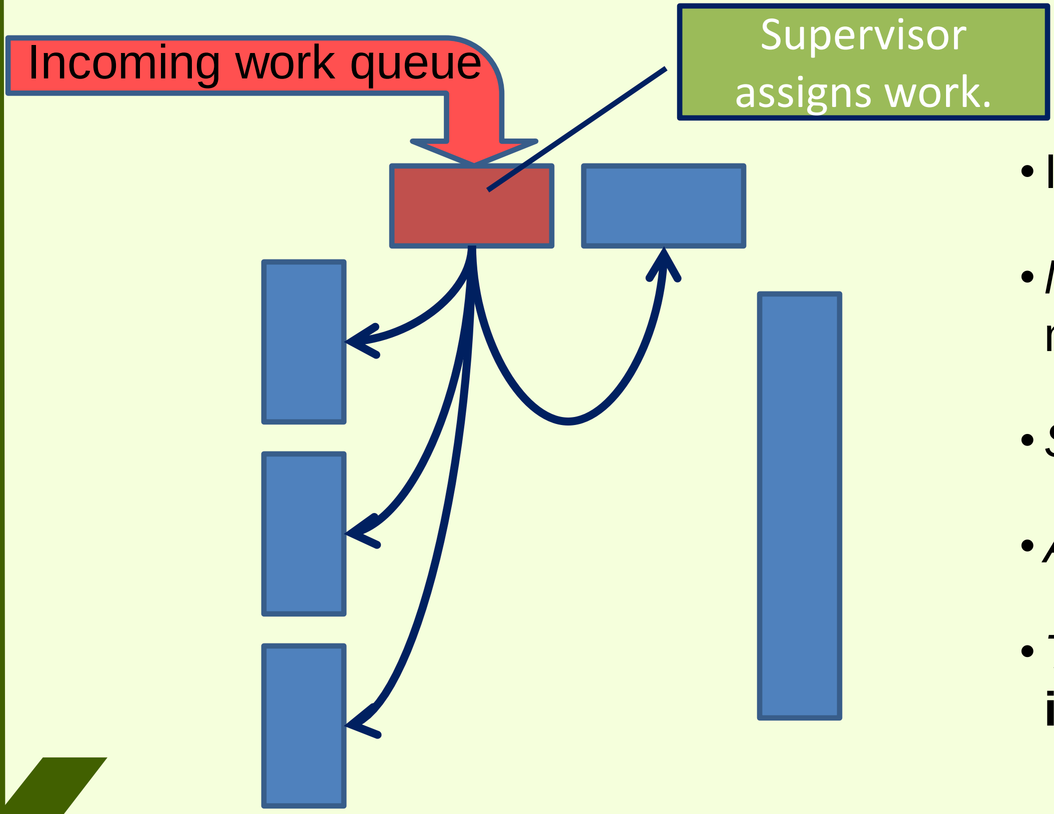
Quick, routine work piles up behind unpredictable work.

Work enters queue faster than it can be processed.

“We need more people.”

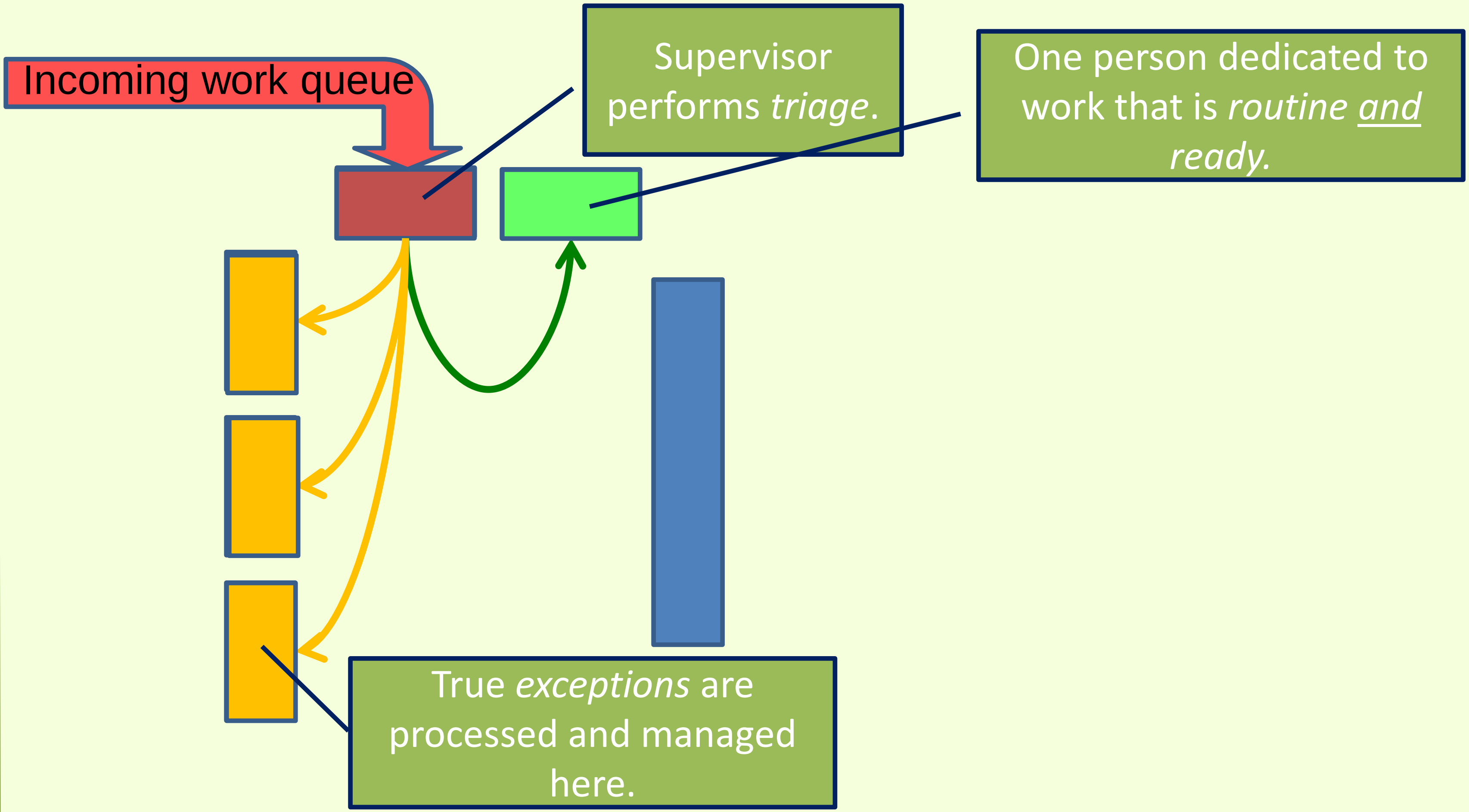
Question:

What would you establish as the *target condition*?

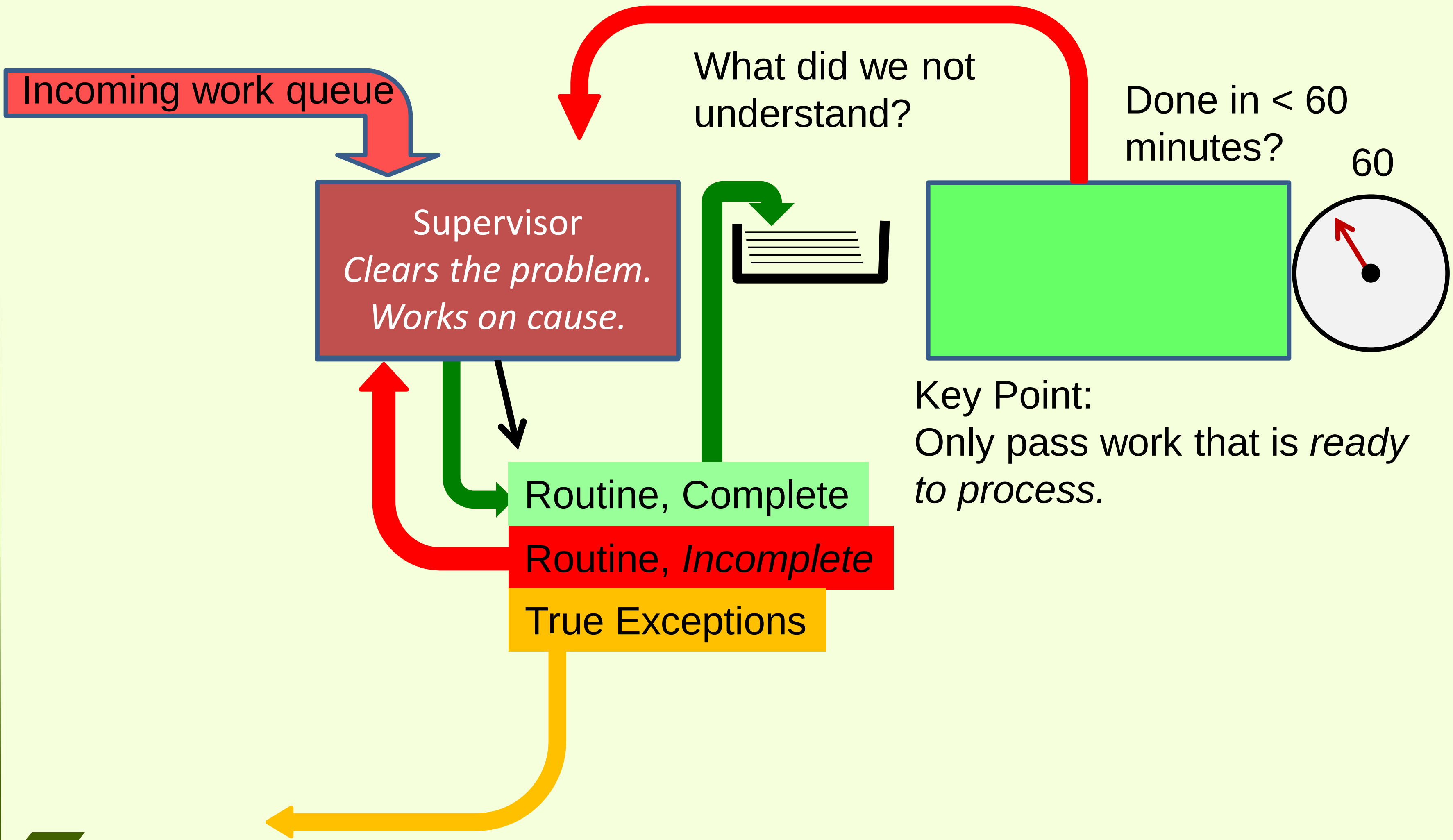


- Incoming load 7-10 / day.
- *Most* could be processed in < 60 minutes. (~ 70%)
- *Some* took a few hours.
- *A few* took a few days.
- *Thus the* ***output cycle*** **was irregular.**

The Assembly Line in the Office



Leadership for Improvement



Results

Backlogs gone in < 2 weeks.

No more Saturday overtime.

Steady reduction in “incomplete” submissions.

Predictable work flow and timing.

Reduced team by one person through attrition.

No more train wrecks.

All they did was change the way they manage work.

Question:

- How did this team apply the principles of:
  - Structuring people's work.
  - Simple and direct information flows.
  - Simple and direct work flows.
  - Immediate response and problem solving.

Boeing 737 Final Assembly

As you watch this assembly line, ask yourself:

- How do they know *each task* is starting in time?
- How do they know the work is progressing normally?
- What indicators are there for right thing, right place, right time?
- What happens when there is a problem?

Boeing 737 Final Assembly

Boeing 737 Final Assembly

Key Point: They did not start this way.

- They started with a simple question:
  - What would stop us if we tried to continuously move the plane?
  - Then an experiment: An RV winch bolted to the floor pulled an airplane through the last position.
  - Systematically worked to resolve sources of delay.

Boeing 737 Final Assembly

Why did they start with a winch, a cable, and one position?

Because Henry Ford started with a car, some skids and 8 feet of rope.

Thought Experiment:

This is an *actual* observation from an *actual* company's operation:

Every 15-20 minutes a pallet of boxed product is delivered from the packaging line.

The Team Member pulls individual boxes of product from a pallet, one by one.

He runs them over a scanner that verifies he has the correct product for that job, then places them in a carton.

Overall, the Team Member's work pace is fast enough to meet the current production requirements.

However, two or three times a minute the scanner's computer faults and he must stop and interact with the keyboard to clear the error and restart the program.

The Team Member is clearly irritated when this happens.

Question:

What do you *expect* a team member with a problem like this to do?

If you saw this, what would you ask the area supervisor?

Question:

What might be your *target condition* <u>for the supervisor's learning</u>?

What would you ask the supervisor to determine the *current condition* of his understanding?