

TRAINING MODULE

# Production Study Analysis Generator

*The hub every other tool in this series quietly depends on*

A working module for engineers with 1–2 years on the floor. By the end you will understand why a single wrong column offset here can silently corrupt everything downstream — and why this tool documents its own bugs on purpose.

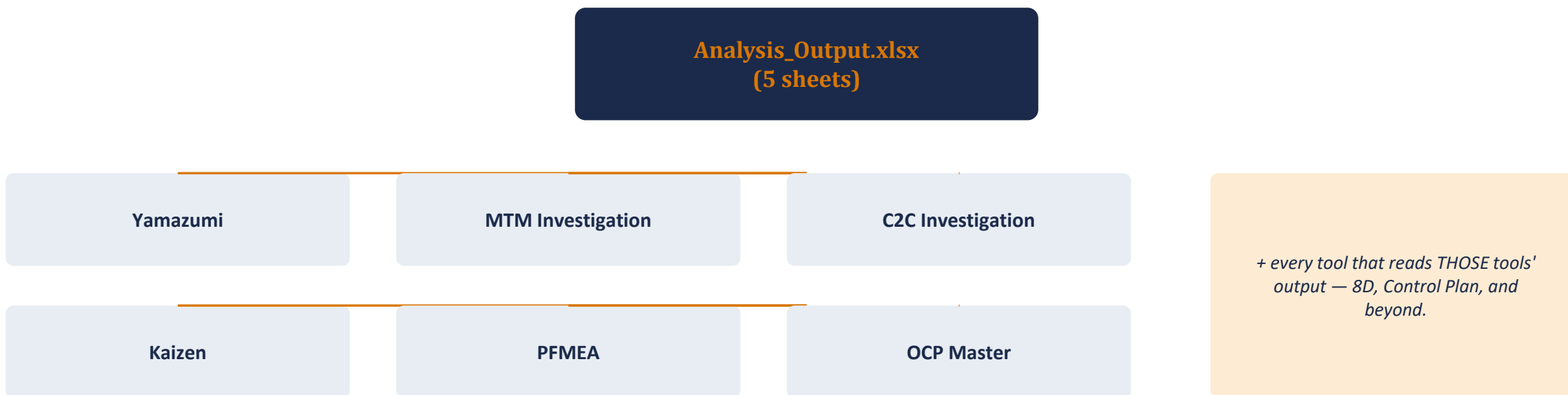
*Grounded in classical Motion & Time Study data-integrity discipline — raw data fidelity as the foundation of every predetermined-time system*

# What You Will Be Able to Do After This Module

- 1 Explain what this tool's five output sheets are, and which other tools in this series depend on each one.
- 2 Explain why a sheet-naming mismatch can cause a station to be silently skipped — zero stations read, no error shown.
- 3 Explain the checkmark-vs-numeric-value bug, and why it's so easy to introduce and so hard to notice.
- 4 Explain why new columns are always appended at the END, never inserted — a fixed-column-position contract other tools rely on.
- 5 Recognize the two most common mistakes junior IEs make when a downstream report looks wrong.

# Almost Every Tool in This Series Reads THIS Tool's Output

Production\_Study\_Analysis\_Output.xlsx isn't just another report — it's the shared foundation underneath most of what you've trained on so far:



A silent bug here doesn't just produce one wrong report — it propagates through every tool downstream of it, often without any of them ever throwing an error. That's exactly why this tool's own docstring is a list of bugs it found and fixed in itself.

# One Column Off, and a Checkmark Becomes Your Data

The flagship bug this tool fixes in itself. A raw station sheet, seen exactly as Excel stores it:

| Col 19            | Col 20                   | Col 21          | Col 22                   |
|-------------------|--------------------------|-----------------|--------------------------|
| ✓                 | 4.0                      | ✓               | 6.5                      |
| <i>Eliminate?</i> | <i>Elim. Savings (s)</i> | <i>Combine?</i> | <i>Comb. Savings (s)</i> |

**ORIGINAL BUG:** code read Column 19 (the ✓ checkmark) hoping for the Eliminate savings value — got a checkmark string instead of a number. Every downstream ECRS savings total was silently wrong.

**FIX:** read Column 20 instead — the numeric value sitting one column over from its own checkbox. `elim=row[20]`, `comb=row[22]`, `rear=row[24]`, `simp=row[26]`.

## Every Predetermined-Time System Lives or Dies on Raw Data

---

Classical Motion & Time Study has always insisted on this: an MTM standard, a C2C%, a Yamazumi bar — every number this whole series has taught you is only as trustworthy as the raw cell it was read from. This tool's docstring lists FOUR bugs it found and fixed in its own column-reading logic. That's not a weakness to hide — it's exactly the discipline this field has always demanded: verify the raw reference before trusting the derived number.

*A published bug-fix list is a sign of a tool you can trust more, not less.*

### STANDARD DATA

*Motion & Time Study  
discipline*

# Five Sheets, Five Different Downstream Jobs

| Sheet                    | Feeds Into                                                                                            |
|--------------------------|-------------------------------------------------------------------------------------------------------|
| ECRS Observation Summary | One row per improvement observation per station — feeds Kaizen's ECRS → TPM Loss mapping.             |
| MTM RCA & CAPA           | Every flagged element vs. MTM standard — feeds MTM Golden Zone Investigation and PFMEA's Severity.    |
| CAPA Standard Library    | Static Therblig reference — the shared vocabulary every CAPA-generating tool draws from.              |
| Elemental Compilation    | Every element, every station, all columns — the raw material Yamazumi and SWCS build their bars from. |
| C2C Variation Analysis   | Elements with Var% > 15% — feeds the C2C Stability Investigation Engine directly.                     |

## 1

## BUG FIX 1

# Sheet-Name Mismatch

The most dangerous kind of bug: no error, no crash — just silently zero stations read.

*build\_sheet\_map() now resolves by NUMERIC SUFFIX, case-insensitive, regardless of prefix.*

## THE MISMATCH

| Index Says | Actual Sheet Name | Naive Match | Fixed Match                 |
|------------|-------------------|-------------|-----------------------------|
| Stn-1      | JM_Stn-1          | ✗ fails     | ✓ matches (suffix 1)        |
| Stn-3      | Mach_Stn-3        | ✗ fails     | ✓ matches (suffix 3)        |
| Stn-5      | Mach_stn-5        | ✗ fails     | ✓ matches, case-insensitive |

**Original code: “if sno in wb.sheetnames” — always False, because the Index's short name never literally appears in the sheet list. Every single station silently returned zero rows.**

**Nobody would necessarily notice — the workbook would just open with empty sheets, easy to mistake for “no issues found” instead of “nothing was ever read.”**

## 2

## BUG FIX 3

## Header Row Off-by-One

R3 is the LABEL row. R4 is the DATA row. Reading the wrong one gives you column headers as if they were numbers.

*Every column offset was re-verified against the actual live workbook, not assumed from an old template.*

### THE TWO ROWS

Row 3 (index 2): “Obs CT” | “Takt” | “Dist” | “WIP” — these are LABELS, plain text

Row 4 (index 3): 38.0 | 45.0 | 3.2 | 6 — the actual numeric VALUES

### THE VERIFIED COLUMN MAP

| Field              | Column (0-based)  |
|--------------------|-------------------|
| obs_ct             | 2                 |
| takt               | 4                 |
| dist               | 5                 |
| wip                | 6                 |
| ecrs_e / c / r / s | 11 / 12 / 13 / 14 |

# 3

## BUG FIX 4

# “Man Machine” Routing Bug

A single space in a string comparison sent an entire station category to the wrong reader function.

*Fixed by matching the EXACT normalized string, not a partial keyword check.*

### ORIGINAL CHECK

if 'machine' in stype.lower() and 'man' not in stype.lower()

↓ applied to Index value “Man Machine” (with a space)

*‘machine’ in ‘man machine’ → TRUE. ‘man’ in ‘man machine’ → ALSO TRUE (fails the ‘not’ check!) — so this check actually returns FALSE for Man Machine, misrouting it as a pure Machine station.*

**FIX:** `stype.strip().lower() == ‘machine’` — pure Machine stations only get this route. Man AND Man Machine both go to `read_man_station()` correctly.

Lesson: a keyword-‘in’ check is rarely safe when the very word you're excluding is also a substring of the category you meant to include.

# New Columns Get Appended — Never Inserted

The `guided_question()` column is a real, useful addition. Look at exactly where it's allowed to go:

## IF INSERTED IN THE MIDDLE

Every column reference in `generate_kaizen.py`, `generate_pfmea.py`, and `generate_ocp_master.py` — which all read this sheet by FIXED column position — would silently shift and start reading the wrong data.

## APPENDED AT THE END, INSTEAD

Every existing column keeps its exact position. Three downstream tools keep working unmodified. The new column is purely additive — present for anyone who wants it, invisible to anyone who doesn't.

This is the same discipline as the SOP module's verbatim-sourcing rule and the Control Plan's honest boundary — respecting what already depends on a piece of data is as important as producing the data correctly in the first place.

# If Yamazumi or PFMEA Looks Wrong, Check HERE First

1

Is a whole station missing from the downstream report? → Check the sheet-name match (Bug Fix 1 pattern).

2

Does an ECRS savings number look like zero when it clearly shouldn't be? → Check for a checkmark-vs-value column mixup (Bug Fix 2 pattern).

3

Are Takt or Obs CT numbers obviously wrong — like a label showing where a number should be? → Check the header row offset (Bug Fix 3 pattern).

4

Is a Man Machine station's data appearing in the wrong category? → Check the station-type string match (Bug Fix 4 pattern).

# Five Mistakes That Give Junior IEs Away



Assuming a downstream tool's wrong number is that tool's bug — when it's often a raw column read here, upstream.



Trusting a workbook with zero rows for a station as “no issues” instead of investigating whether it was ever actually read.



Inserting a new column mid-sheet because it “fits better there” — breaking every fixed-position reader downstream.



Copy-pasting a keyword-‘in’ string check without checking whether the excluded word is also a substring of the included one.



Never reading this tool's own docstring — it's a documented history of exactly the mistakes this dataset is prone to.

# What generate\_production\_analysis.py Does For You

Reads every station correctly, once, so every other tool in this training series can trust the result.

1

**Resolves sheet names correctly**

build\_sheet\_map() matches by numeric suffix, case-insensitive, regardless of prefix — Bug Fix 1, applied automatically.

2

**Reads the right ECRS columns**

Numeric savings values, not checkmarks — Bug Fix 2, applied to every element on every station.

3

**Reads the right header row and columns**

Verified against the live workbook — Bug Fix 3, no more label-as-number errors.

4

**Routes Man Machine stations correctly**

Exact string match, not a fragile keyword-‘in’ check — Bug Fix 4, fixed at the source.

5

**Produces all 5 output sheets, append-only**

Everything downstream keeps working, every time this tool is re-run.

## KEY TAKEAWAYS

# Recap — Say This Back In Your Own Words

- This tool's 5 output sheets are the shared foundation almost every other tool in this series reads from.
- The most dangerous bugs are silent — zero rows, a checkmark instead of a number — no crash, no obvious error.
- A documented bug-fix history is a sign of trustworthiness, not a weakness to hide.
- New columns get appended, never inserted — respecting every downstream tool's fixed-position contract.

## TALK IT THROUGH — 3 QUESTIONS FOR YOUR NEXT LINE WALK

1. If a Yamazumi chart on your line looked obviously wrong tomorrow, which of the four bug patterns would you check first — and why that one?
2. Have you ever seen a report with suspiciously “clean” zero results across a whole station — could it have actually just never been read?
3. Where else in your own tools or spreadsheets might a keyword-‘in’ check be hiding the same kind of substring trap as Bug Fix 4?