Dimensional analysis. AKA Why units matter.

Every quantity is measured in what's called a dimension.

Ex. 100 **meters**, 60 **seconds**, 10 **dollars**.

For these examples the dimension the quantity exists in is **meters**, **seconds** or **dollars**. Dimensions are also known informally as units.

Why are units important?

Units provide us with a simple method to "check" that our calculations make logical sense, that is, are our calculations are correct? Consider the following.

Ex. Convert 1.0 meter to centimeters.

This calculation is trivial because we already know the **conversion factor** between meters and centimeters, multiply by a factor of **100**. But what is happening mathematically? Here the conversion factor is read as "**100.0 centimeters per 1.0 meters**"

$$(1.0\ meters) * (\frac{100.0\ centimeters}{1.0\ meters})$$

A basic rule of algebra allows us to rearrange the numerator and denominator allowing the grouping of similar terms.

$$(\frac{1.0}{1.0})\ *\ (100.0) * (centimeters) * (\frac{meters}{meters})$$

We know that a term divided by itself is equivalent to 1 so we can cancel out some terms.

$$(1)\ *\ (100.0) * (centimeters) * (1)$$

Yielding the correct answer

$$100.0\ centimeters$$

What if we made a mistake with our conversion factor? What if we invert it?

$$(1.0\ meters) * (\frac{1.0\ meters}{100.0\ centimeters})$$

Skipping to the end

$$0.01\ \frac{meters^2}{centimeters}$$

Not only is the quantity wrong, but the units are nonsense. Clearly, we made a mistake.

Percentages

Percentages don't have units; if yours do there is a mistake in the math.

Ex. Calculate the effective tax rate for **750.00** net pay on **1000.00** gross pay.

$$\frac{750.00\ dollars}{1000.00\ dollars}$$

Group terms again.

$$(\frac{750.00}{1000.00}) * (\frac{dollars}{dollars})$$

Cancel like terms and simplify. Notice the units, **dollars**, are gone.

$$0.75$$

But we aren't done yet, we need to multiply by another conversion factor to get a percentage.

$$(0.75) * (100\%)$$

$$75\%$$

So we take home **75%** of our pay, or put another way, pay a **25%** tax rate. Again notice there are no dollars present in the final answer.

How does this relate to Euro metrics?

We are tracking and comparing two different units, quantities in two different dimensions, and trying to compare them directly. What are we doing?

1. Tracking total errors performed by an assembler
2. Comparing to total units produced by an assembler.

1. Every time an assembler performs an action, that action is either a success or failure. The dimension, or units, can be considered **actions**.

   If we are interested the absolute failure rate of an assembler, we need to compare their bad actions against their total actions. Consider the following.

   Ex. An assembler crimps 1000 pins, 10 are bad, what is their failure rate?

$$(\frac{10\ actions}{1000\ actions}) * (100\%)$$

   Collect similar terms

$$(\frac{10}{1000}) * (\frac{actions}{actions}) * (100\%)$$

Simplify

$$1\%$$

So our assembler has a **1%** failure rate, again notice the units, **actions**, are gone.

2. The fault in our logic comes from the fact that we are comparing the assemblers failed actions against **their total items built**.

   Ex. An assembler crimps **1000** pins with **10** bad pins; the assembler has constructed **100** items.

$$(\frac{10\ actions}{100\ items}) * (100\%)$$

Simplify

$$10\frac{actions}{items}\%$$

This is the same situation as the previous example, but now our number is much larger and we have a nonsensical **actions per item percent** dimension. Remember percentages don't have dimensions.

It gets even worse, what if the assembler had the same pins fail but only constructed **10** units?

$$\frac{10\ actions}{10\ items} * (100\%)$$

Simplify

$$100\frac{actions}{items}\%$$

Even if we ignore the units, **which is what we are currently doing**, the math showing a **100%** failure rate, when in reality we know it is only **1%**.

The error rate we are using now depends entirely on how many items the assembler has made not how many actual errors they are making.

OK, so the numbers may be wrong, but at least we can determine trends right?

Unfortunately no.

The computed error rate is mostly dependent on the quantity and complexity of the cables but also how the mistakes are distributed to each cable. Consider the following.

Ex. An Assembler builds **3** cables, and has **3** crimp failures. Let's consider what effect the distribution of the errors has on the calculations.

Case I – There is a single error on each cable.

$$\frac{3\ actions}{3\ items} * (100\%)$$

$$100\frac{actions}{items}\%$$

If we are interested in how many items have failed, which is what we currently calculate, this number is correct, but the presence of **actions per items %** indicates there is a problem in the logic.

Case II – All three pin errors are on the same cable.

$$\frac{3\ actions}{3\ items} * (100\%)$$

$$100\frac{actions}{items}\%$$

The math is the same, but in this case we know only **1** cable had problems. Shouldn't the failure rate be **33%**?

Case III – Cable 1 has 1 error, cable 2 has 2 errors and cable 3 has zero errors.

$$\frac{3\ actions}{3\ items} * (100\%)$$

$$100\frac{actions}{items}\%$$

Again the math is the same, but in this case we know **2** cables have problems, the expected failure rate should be **66%**.

The fault in our results arises from the random combinations of quantity and complexity of cables an assembler may build. The more cables and the more actions performed result in more possible outcomes, known as **permutations**. Across thousands of cables and tens of thousands of actions, there are millions of combinations which make finding any trends in the data most likely impossible. We would require sampling in the millions of kits to parse any useful information.

What should we do then?

Proposed solution – Group discussion

Euro has been collecting assembly data for two primary goals, our quality objectives and assembler performance. These need to be tracked in two different ways.

1. Quality objective 3. **An internal defect rate of less than 10%**

   This objective is not clear in what it intends to track. Are we tracking how many units have a Q.C. failure, or are we tracking internal failed actions?

   I propose that we modify this objective to

   "**An internal unit failure rate of less than 10%**"

   where we keep track of how many units an assembler has built that have failed Q.C. In this situation a cable with 3 bad pins would be treated the same as a cable with 1 bad pin, both would count as a single unit failure. This method works best with a complete cable, and would be most useful for final Q.C., cami, etc. Q.C would need to decide and what constitutes a unit failure.

   This method provides a very simple, broad view of our production quality strictly for the purposes of meeting our ISO objectives.

2. Assembler performance.

   Without knowing how many total actions an assembler performs in a time period, it is impossible to calculate their exact failure rate. The effort to compute all total actions performed across all possible kits is not an effective use of resources which, I believe, Javier attempted to implement in some form.

   An alternative is to compute the relative error rate of an assembler versus themselves and each other.

   Ex.

	January	**February**	**March**
Assembler 1	500	550	650
Assembler 2	150	160	140
Assembler 3	500	400	300

For this example, we can’t compute an absolute percentage, but we can easily discern trends. Assembler 1 requires a performance improvement plan. Assembler 2 is stable and effective. Assembler 3’s improvement plan has been successful. This data could also be charted and provided to the assembler on a quarterly basis or during performance reviews.