

Multi-Plant OEE Standardization Kit

Shared OEE definitions, a two-level downtime reason-code taxonomy, a governance model and an 8–12 week pilot-to-scale plan for rolling OEE across sites.

Why standardise before you scale

Cross-plant OEE comparison is meaningless unless every site defines the same things the same way. If one plant counts planned maintenance as downtime and another doesn't, your league table is noise. This kit gives you the shared definitions, a reason-code taxonomy and a governance model so ten factories feed one source of truth instead of ten spreadsheets.

1 · Standard OEE definitions (adopt verbatim)

Planned production time	Calendar time minus time you deliberately did not schedule (no demand, no shift).
Operating time	Planned production time minus planned stops (breaks, meetings, planned maintenance).
Run time	Operating time minus all unplanned downtime.
Availability	$\text{Run time} \div \text{operating time}$.
Performance	$(\text{Ideal cycle time} \times \text{total count}) \div \text{run time}$.
Quality	$\text{Good count} \div \text{total count}$.
OEE	$\text{Availability} \times \text{Performance} \times \text{Quality}$.

2 · Standard downtime reason-code taxonomy

Two-level codes: a fixed top level (comparable across sites) and a free local sub-level. Never let sites invent new top-level codes.

A1	Breakdown	Motor fault, jam, sensor failure
A2	Setup / changeover	Tooling change, format change, warm-up
A3	Starved / blocked	Waiting material, downstream full
P1	Minor stop	Misfeed, clearing, quick adjust

P2	Reduced speed	Worn tooling, cautious run, wrong recipe
Q1	Startup reject	Scrap until process stabilises
Q2	Production reject	Defect, rework, out-of-spec
N1	Planned (not a loss)	Break, meeting, planned maintenance

3 · Governance model

Central OEE lead	Definitions, code taxonomy, cross-site benchmarks	Monthly review
Site champion	Local rollout, operator training, data quality	Weekly
Line owner	Reason coding accuracy on the shift	Every shift
Sponsor	Prioritisation and investment decisions	Quarterly

Rule: local exceptions to a definition are allowed, but must be documented centrally, or they silently corrupt every cross-site comparison.

4 · The 8–12 week pilot-to-scale plan

1–2	Instrument one bottleneck line at the pilot site	Live A×P×Q on that line
3–6	Prove value vs the line's own baseline	Quantified OEE gain, operators coding stops
7–8	Freeze definitions and reason codes centrally	Signed standard, documented exceptions
9–12	Duplicate the config to the next sites	Second site live by config, not new project

5 · Cross-site rollout checklist

- Definitions and reason codes identical and centrally owned
- Each site feeds one platform / one source of truth, not local sheets
- Adding a site duplicates a configuration rather than launching an integration
- Local-language operator screens at every plant
- Group → plant → line → machine roll-up available in one place
- Data survives network outages at every site