WHITE PAPER

Quality Gates That Hold: Incoming, In-Process & Outgoing Inspection in ERPNext

How ERPNext's Quality Inspection turns 'we check our stuff' into gates that actually stop bad material — at the goods receipt, on the shop floor, and before dispatch.

For quality & production leaders · **8 min read**

Acube Innovations · Official ERPNext Partner · acubeinnovations.com · +91 62358 66111

EXECUTIVE SUMMARY

Most factories say they inspect. Fewer can stop a rejected lot from being received, consumed or shipped — because the check lives in a register or a person's head, not in the system that moves the stock. ERPNext's Quality Inspection closes that gap. You define what 'good' means as reusable parameters with real acceptance criteria — numeric tolerances, accepted values, or a formula — group them into a template per item, and then inspect against them at three points: Incoming (goods in), In Process (on the shop floor), and Outgoing (before dispatch). The important part is enforcement: when an item is flagged as requiring inspection, ERPNext will warn you on save and refuse to submit the Purchase Receipt, Stock Entry or Delivery Note until a passing inspection exists — and it blocks outright if the inspection was rejected. This paper explains the real doctypes behind that, how the three gates map to your goods receipt, manufacturing and delivery flows, and what it takes to make the gates actually hold rather than become a rubber stamp.

Why quality gates matter — and why most fail quietly

Almost every manufacturer inspects incoming material and checks finished goods. The problem is rarely the checking; it's that the check and the stock movement live in two different worlds. The QC register says a lot was rejected, but the store already received it, production already drew from it, and by the time anyone reconciles, the defect is in a customer's hands. A gate that doesn't stop anything isn't a gate — it's a record of what you failed to stop.

The fix is to make the inspection a precondition of the transaction, not a parallel activity. That means the acceptance criteria have to be defined in the system, the inspection result has to be attached to the exact receipt, batch or delivery line it governs, and the transaction has to refuse to complete when the material hasn't passed. ERPNext's Quality Inspection is built to do exactly this — it is a submittable document in the Stock module that carries the readings, the pass/fail status, and a hard link back to the transaction it belongs to.

The rest of this paper walks the real structure: how you define 'good', where the three inspection types fit, and how ERPNext turns a failed inspection into a blocked transaction instead of a footnote.

- The common failure: inspection happens, but nothing stops rejected stock from being received, consumed or shipped.
- A real gate makes the inspection a precondition of the transaction — receipt, stock entry or delivery.
- Quality Inspection is a submittable Stock document that holds the readings, the Accepted/Rejected status and a link to the transaction.
- Get the definition of 'good' and the enforcement right, and QC stops being a register nobody reads.

Defining 'good': parameters, acceptance criteria and templates

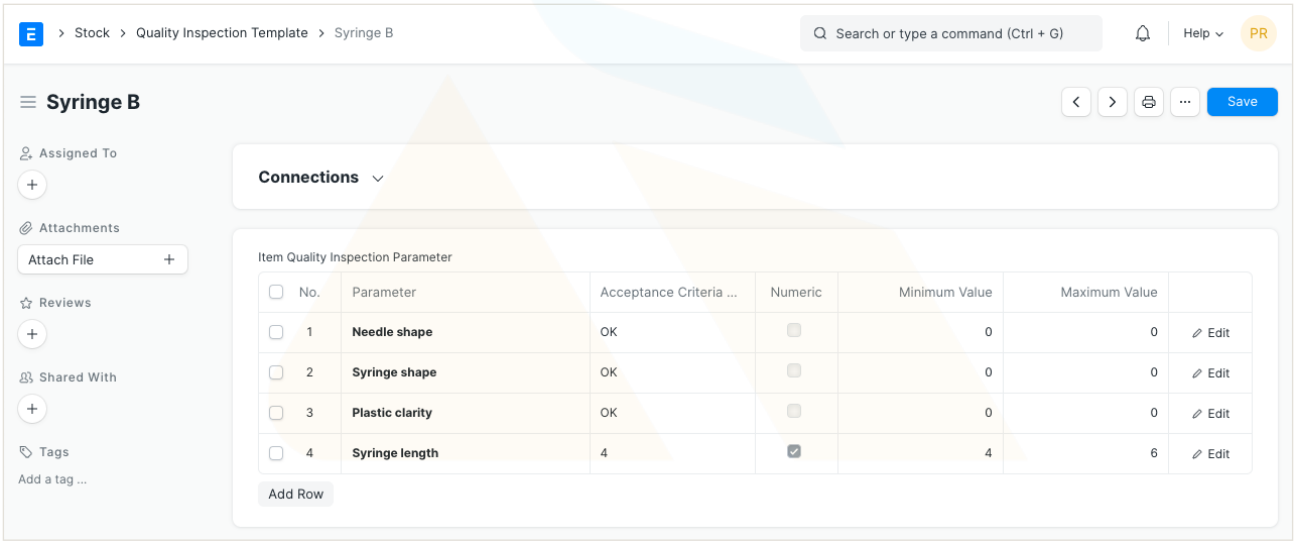
Everything starts with what you measure. In ERPNext a Quality Inspection Parameter is a reusable master — 'Tensile strength', 'Moisture %', 'Needle shape', 'Plastic clarity' — optionally organised into parameter groups. You define a parameter once and reuse it across every item and every template, so

your quality vocabulary stays consistent instead of being retyped differently on each check.

Against each parameter you set an acceptance criterion, and ERPNext supports three honest kinds. A numeric parameter uses a minimum and maximum value — the reading has to fall inside the tolerance band (e.g. length between 4 and 6). A value-based parameter uses an accepted value — the reading has to match (e.g. 'OK', or a specific grade). And a formula-based parameter uses a simple Python expression evaluated on the readings — for example a mean across sample readings above a threshold, or a reading value that must be in a set of allowed grades. That covers the three ways real specifications are written: 'within tolerance', 'must equal', and 'passes this rule'.

You then bundle the parameters an item needs into a Quality Inspection Template — a named set of parameters with their criteria (the screenshot below shows one). Attach that template to the item, and every inspection for that item is pre-populated with the right checks and thresholds. Define the standard once; apply it consistently forever.

- Quality Inspection Parameter — a reusable, named check (optionally grouped) so your quality vocabulary is consistent.
- Numeric criteria — a minimum/maximum tolerance band each reading must fall within.
- Value-based criteria — an accepted value the reading must match exactly (e.g. 'OK' or a grade).
- Formula-based criteria — a Python expression on the readings (e.g. mean above a threshold, or reading in an allowed set).
- Quality Inspection Template — a per-item bundle of parameters + criteria, attached to the Item so every check is pre-loaded.



A Quality Inspection Template in ERPNext: each parameter carries its acceptance criteria — a matched value for pass/fail checks, or a numeric minimum and maximum tolerance band.

The three gates: Incoming, In-Process and Outgoing

A single Quality Inspection document carries an Inspection Type, and ERPNext offers exactly three: Incoming, In Process and Outgoing. They are not cosmetic labels — each maps to a different transaction and a different moment in the flow, and together they form three gates around your stock.

Incoming inspection governs material arriving from outside: it attaches to a Purchase Receipt (or Purchase Invoice that updates stock), and increasingly to a Subcontracting Receipt when finished

goods come back from a job-work vendor. This is the gate that keeps a supplier's defect from ever entering your warehouse. In-Process inspection governs work on your own shop floor: it attaches to a Stock Entry — most importantly the finished-good row of a Manufacture entry, but also material transfers and repacks — so a batch is checked at the point it's made or moved, not after it's already downstream. Outgoing inspection governs what leaves you: it attaches to a Delivery Note (or a Sales Invoice that updates stock), so the last thing that happens before dispatch is a check that the customer is getting conforming goods.

Each inspection records the item, and optionally the batch or serial number it covers, the sample size, who inspected and who verified — so the result is tied to a specific quantity of specific stock, not a vague 'this looks fine'.

- Incoming — attaches to a Purchase Receipt / stock-updating Purchase Invoice / Subcontracting Receipt: catches supplier defects at the gate.
- In Process — attaches to a Stock Entry (notably the finished-good row of a Manufacture entry, plus transfers/repacks): checks your own output as it's made.
- Outgoing — attaches to a Delivery Note / stock-updating Sales Invoice: the last check before goods reach the customer.
- Every inspection ties to a specific item, batch/serial and sample size — with an inspected-by and verified-by trail.

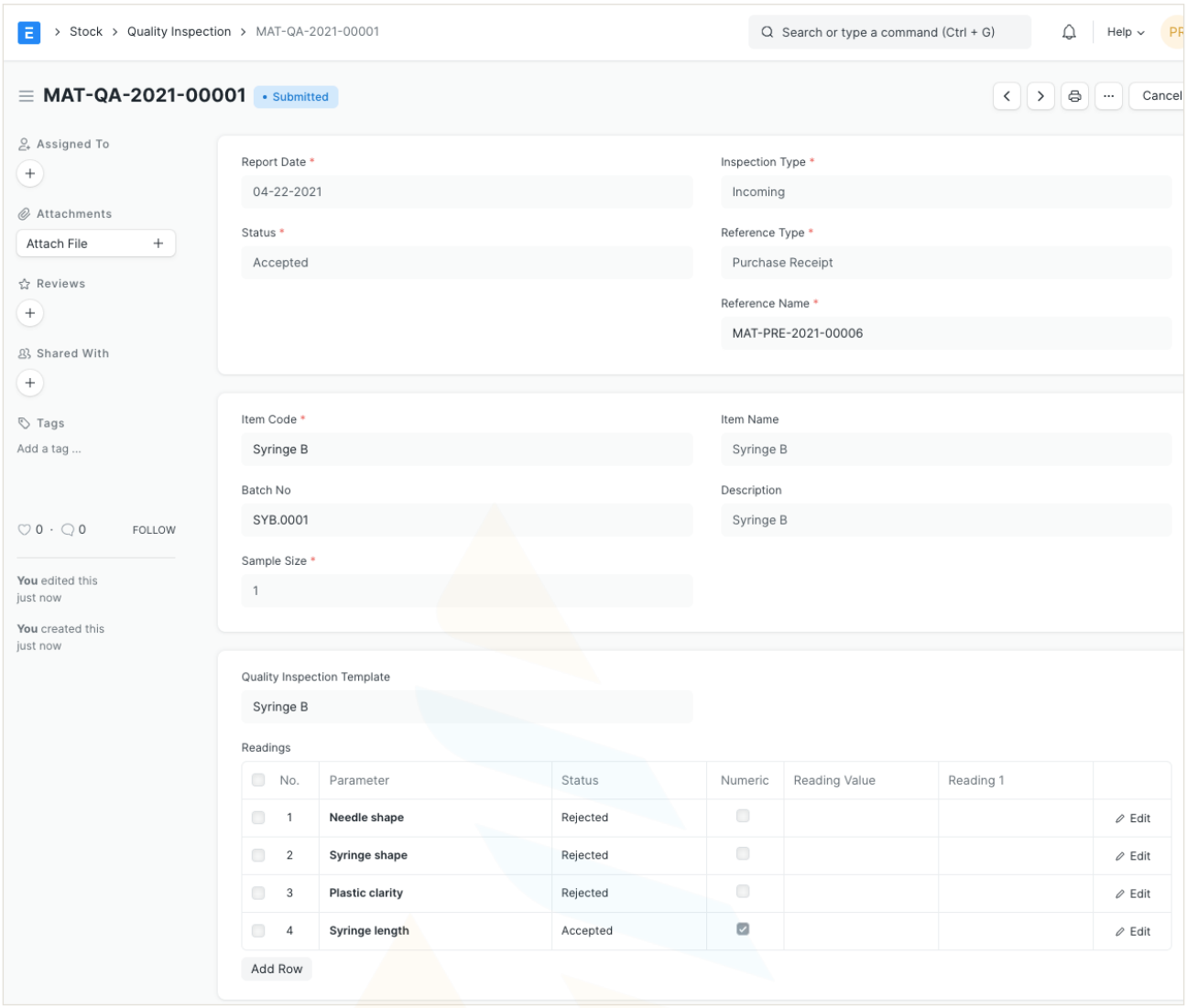
Three inspection gates, mapped to their transactions

1. **Incoming inspection**
   attaches to a Purchase Receipt, stock-updating Purchase Invoice or Subcontracting Receipt: stops a supplier's or job-work vendor's defect from entering your warehouse.

2. **In-Process inspection**
   attaches to a Stock Entry, notably the finished-good row of a Manufacture entry (plus material transfers and repacks): checks your own output at the point it's made or moved.

3. **Outgoing inspection**
   attaches to a Delivery Note or stock-updating Sales Invoice: the last check before goods leave you, so the customer only receives conforming stock.

4. **The enforcement**
   flagged items warn on save, hard-stop on submit if the inspection is missing or unsubmitted, and block outright if it was Rejected.



A submitted Quality Inspection: type Incoming, linked to a Purchase Receipt and batch, with each parameter in the readings table individually marked Accepted or Rejected.

Blocking bad stock from flowing — the enforcement that makes it real

This is where ERPNext earns the word 'gate'. On the Item master you flag which items demand inspection: 'inspection required before purchase' and 'inspection required before delivery' (and a stock entry can carry its own inspection-required flag). Once an item is flagged, the transaction validates against it automatically.

The enforcement is graded, which matters in practice. When you save a transaction whose item requires inspection but has no Quality Inspection attached, ERPNext warns you — a visible prompt, not yet a wall, so data entry isn't paralysed. When you try to submit that transaction, the warning becomes a hard stop: it throws a Quality Inspection Required error and refuses to submit. If an inspection exists but hasn't been submitted, submission is blocked until it is. And critically, if the attached inspection was submitted with a Rejected status, ERPNext blocks the transaction outright — you cannot submit a receipt or delivery for material your own quality team has failed. That single rule is the difference between a QC system and a QC report.

ERPNext keeps one deliberate escape hatch: a Stock Settings option to allow creating the quality

inspection after the purchase or delivery, for shops whose process genuinely inspects post-receipt. It's a conscious setting, not an accident — the default posture is that the gate holds.

- Item flags — 'inspection required before purchase' / 'before delivery' turn an item into a gated item; stock entries carry their own flag.
- On save — a missing inspection triggers a warning, so entry isn't blocked prematurely.
- On submit — a missing or unsubmitted inspection is a hard stop (Quality Inspection Required / Not Submitted).
- Rejected inspection — the transaction is blocked outright; you can't submit a receipt or delivery for failed material.
- One deliberate escape hatch — a Stock Settings option to inspect after receipt/delivery, for processes that genuinely need it.

Linking inspection to the receipt, stock entry and delivery

The reason enforcement works is that a Quality Inspection isn't a floating document — it holds a reference type and reference name (a dynamic link) pointing at the exact transaction it governs: this Purchase Receipt, this Stock Entry, this Delivery Note, this Subcontracting Receipt or Job Card. The transaction line, in turn, carries the inspection back. The link is two-way, per row, so nothing is ambiguous about which check covers which quantity.

Inside the inspection, the readings child table is where measurement meets criteria. Each parameter row pulls its acceptance rule from the template, records the actual sample readings (ERPNext allows up to ten readings per parameter for value-based sampling, plus a reading value), and gets its own Accepted or Rejected status. The document's overall status rolls up from the rows — one rejected parameter can fail the inspection. There's also a manual-inspection option for genuine tolerance calls a human has to make, so the system supports judgement without abandoning structure.

Because the whole thing is a submittable document with a naming series and its own permissions (the Quality Manager role owns it by default), inspections are auditable records, not scratch notes — you can trace, for any receipt or dispatch, exactly what was checked, by whom, against which thresholds, and whether it passed.

- Reference type + name — a dynamic link ties each inspection to its Purchase Receipt, Stock Entry, Delivery Note, Subcontracting Receipt or Job Card.
- Readings table — each parameter records actual sample readings against its criteria and gets an Accepted/Rejected status; the document status rolls up.
- Up to ten readings per parameter, plus a manual-inspection option for legitimate human tolerance calls.
- Submittable, named, permissioned (Quality Manager) — inspections are auditable records tied to real stock, not loose notes.

What it takes to make the gates actually hold

The doctype is capable; whether it protects you comes down to setup and discipline. First, the definition of 'good' has to be real — parameters with criteria that reflect your actual specification, not a template someone half-filled. A numeric band with no min and max, or an accepted value that's always 'OK', is a

gate that opens for everything. Second, the right items have to be flagged for inspection; an unflagged item is invisible to the enforcement, however diligent the team.

Third, sampling and readings have to be honest — the sample size should mean something, and readings should be recorded, not defaulted through. Fourth, someone has to own the rejected path: a blocked receipt or a failed batch needs a defined route (return to supplier, rework, scrap, or concession), or people will quietly reach for the escape-hatch setting and the gate erodes. Get these four right and Quality Inspection stops rejected material at three points automatically; get them wrong and it becomes exactly the rubber stamp it was meant to replace.

- Real criteria — numeric bands, accepted values or formulas that reflect the true spec, not placeholders.
- Flag the right items — inspection-required flags on the items that matter, or enforcement never fires.
- Honest sampling — a meaningful sample size and readings actually recorded, not defaulted.
- Own the rejected path — a defined route (return, rework, scrap, concession) so nobody quietly bypasses the gate.

When to get help

You can configure Quality Inspection yourself, and for a handful of items many teams do. The place an experienced partner earns their fee is in translating a real quality plan into the system: mapping your incoming, in-process and outgoing checks onto the right transactions, writing acceptance criteria (especially the formula-based ones) that behave correctly, deciding which items to gate and how strictly, and wiring the rejected-material workflow so a block leads somewhere instead of festering.

As an official ERPNext partner working with Indian manufacturers, we set up the parameters, templates and inspection-required flags to match how your material actually moves — so Incoming catches supplier defects at the gate, In-Process catches your own before they travel, and Outgoing is a genuine last line before dispatch. If quality is something you're audited on or paid for, the difference between a gate that holds and a register nobody reads is worth getting right the first time.

KEY TAKEAWAYS

1. Quality Inspection is a submittable Stock document that links to a specific transaction and holds the readings and an Accepted/Rejected status — the check lives where the stock moves.
2. Define 'good' once: reusable parameters with numeric tolerance bands, accepted values or Python formulas, bundled into a per-item Quality Inspection Template.
3. Three inspection types form three gates — Incoming (Purchase/Subcontracting Receipt), In-Process (Stock Entry) and Outgoing (Delivery Note) — each mapped to a real transaction.
4. Enforcement is graded and real: flagged items warn on save, hard-stop on submit if the inspection is missing or unsubmitted, and block outright if it was Rejected.
5. The gates only hold with discipline: real acceptance criteria, the right items flagged, honest sampling, and an owned route for rejected material.

FAQ

Can ERPNext actually stop us from receiving or shipping rejected material?
Yes. When an item is flagged as requiring inspection, ERPNext validates the transaction: it warns you on save if no Quality Inspection is attached, and on submit it becomes a hard stop — the Purchase Receipt, Stock Entry or Delivery Note won't submit without a submitted inspection. If the attached inspection has a Rejected status, the transaction is blocked outright. There's an optional Stock Settings switch to allow inspecting after receipt or delivery for processes that genuinely need it, but by default the gate holds.

What kinds of acceptance criteria can we set?
Three. Numeric parameters use a minimum and maximum value, so each reading has to fall inside a tolerance band. Value-based parameters use an accepted value the reading must match (for example 'OK' or a grade). Formula-based parameters use a simple Python expression evaluated on the readings — for instance a mean across sample readings above a threshold, or a reading value that must be in a set of allowed grades. That covers 'within tolerance', 'must equal', and 'passes this rule'.

How is inspection linked to purchase receipts, stock entries and deliveries?
Each Quality Inspection carries a reference type and reference name — a dynamic link pointing at the exact transaction it governs: a Purchase Receipt, Purchase Invoice, Subcontracting Receipt, Stock Entry, Delivery Note, Sales Invoice or Job Card. The transaction line carries the inspection back, so the link is two-way and per row. That's why enforcement can be precise about which check covers which quantity of stock.

Do we have to inspect every item and every unit?
No. You choose which items are gated by setting the inspection-required flags on the Item master (before purchase and/or before delivery); unflagged items pass through without an inspection. On each inspection you set the sample size and record the sample readings — ERPNext allows up to ten readings per parameter for value-based sampling — and there's a manual-inspection option for genuine tolerance calls a person has to make. So you can be strict where it matters and light-touch elsewhere.

Talk to a real ERPNext expert.

Call or WhatsApp +91 62358 66111 · info@acube.co · acubeinnovations.com