ALLIED BizTech SOLUTIONS

FIELD GUIDE · 2026

Rebuild NetSuite yourself.
The plan we follow.

For mid-market teams on 50 to 500+ NetSuite users who want to own their orders, inventory, procurement and AR/AP, keep the GL their auditors know, and stop paying per user and per module.

- → Is it worth it, at your size
- → What to rebuild and what to drop
- → The architecture, in three layers
- → Migration and go-live checklists
- → The data model
- → A week-by-week build plan
- → The ledger, the close, the auditors
- → Where teams get stuck

NetSuite charges three ways. *A rebuild charges one.*

NetSuite bills a base subscription by edition, a fee per user, and an add-on per module: Advanced Inventory, Multi-currency, Fixed Assets, SuiteAnalytics, OneWorld. Each grows on its own. A rebuild replaces all three with Postgres hosting, the GL software your auditors already accept, and maintenance.

SIZE	NETSUITE A YEAR	REBUILT: YEAR ONE / AFTER	SAVED IN YEAR ONE
25 users · ERP base only	~$80,000	~$77,100 / $24,000	~$2,900
50 users · ERP + 3 modules	~$160,000	~$83,100 / $30,000	~$76,900
100 users · ERP + 5 modules	~$280,000	~$101,100 / $48,000	~$178,900
250 users · OneWorld, multi-subsidiary	~$700,000	~$201,600 / $86,000	~$498,400
500+ users · global, all modules	~$1,800,000+	~$418,700 / $200,000	~$1,381,300

NetSuite does not publish its prices: ranges as of May 2026, from third-party buyer reports and reseller proposals; confirm against your own quote. "Rebuilt" runs the ERP on Postgres with QuickBooks Enterprise or Sage Intacct as the GL, plus our fixed build price: $53,100 up to 100 users on one entity, $115,600 (the top of the $76,600–$115,600 tier) to 250, $218,700 (the top of $140,600–$218,700) above. If your own team builds it, put your engineering cost in that place.

THE RULE OF THUMB

At 25 users, year one comes out ~$2,900 ahead and $56K a year comes back after. **At 100 users, the rebuild saves ~$178K in year one and ~$232K a year after.** From 250 users, year one alone saves more than $450K.

Who you need

- **A senior full-stack engineer** at home with Postgres, double-entry accounting and the QuickBooks Enterprise or Sage Intacct API. Our team ships this in 14–18 weeks; one engineer between other priorities should plan for far longer.
- **Your controller**, a few hours a week: the account mapping, the close calendar, the pilot business unit, the sign-off.
- **Your auditor or CPA**, early: a walkthrough of the journal feed, the audit log and the controls before the pilot ends.
- **An owner after launch**: hosting, the posting queue, upgrades, and on-call at month end.

Rebuild the 15–25% you use. *Drop the rest.*

A SaaS, services or D2C business at $20M–$500M revenue uses 15–25% of NetSuite every day. That share is the project. Manufacturing and regulated industries need careful scoping first.

REBUILD ONE FOR ONE

- **Order management**: sales and purchase orders, multi-line, approvals, status tracking
- **Inventory**: multi-warehouse, serialized, lot, valuation
- **Procurement**: POs, vendors, receiving, approvals
- **AR/AP automation**: invoicing, collections, vendor payments, approval workflows
- **Customer and vendor master** with your tax and credit terms and contact hierarchy
- **Journal entries** to your GL on QuickBooks Enterprise or Sage Intacct
- **Multi-currency**: daily FX rates, realized and unrealized gains and losses
- **Custom fields and forms**, typically 10–25 per object
- **Reporting**: P&L, balance sheet, cash flow, trial balance, AR/AP aging, margin

REBUILD, SIMPLER

- **SuiteAnalytics**: Metabase or Looker on your warehouse
- **SuiteScript, SuiteFlow**: TypeScript or Python
- **Multi-subsidiary** for 1–3 entities
- **SuiteApps**: your top 3–5

DROP ON PURPOSE

- **SuiteCommerce**: Shopify, Saleor or Medusa.js
- **Manufacturing**: Fishbowl or Katana
- **5+ entities**: stay on NetSuite
- **The SuiteApp long tail**
- **NetSuite training** and certifications

ADD WHAT PER-USER PRICING DISCOURAGES

- Users at no fee; cost scales with bytes
- Direct SQL, no SuiteAnalytics seat
- Workflow in version-controlled code

Three layers. *The audited GL stays.*

NetSuite is an ERP data model, workflow, accounting, inventory and reporting on commodity infrastructure. Split it the way auditors already see it: an ERP that runs the business on Postgres, feeding journal entries to a standard GL. Only the NetSuite layer goes.

LAYER 1 · PEOPLE USE

The ERP workspace

- Sales orders and invoices
- Inventory by warehouse, serial and lot
- POs, vendors and receiving
- Customer and vendor records
- Revenue, margin, AR/AP aging
- Multi-entity, 1–3 entities

LAYER 2 · DATA + WORKFLOW

Postgres

- Order, inventory, customer and vendor tables
- Workflow: approvals, notifications
- Multi-currency, daily FX rates
- Audit log on every change
- Custom fields and objects
- Period-end close workflows

LAYER 3 · ACCOUNTING

The GL

- **QuickBooks Enterprise** below $50M revenue
- **Sage Intacct** above $50M
- Or the GL you already run
- Journals posted by API
- Trial balance and close support

Decisions to make in week one

DECISION	OUR DEFAULT	WHY
The GL	QuickBooks Enterprise below $50M revenue, Sage Intacct above, or the GL you run today	Auditors keep software they already accept
Posting grain	One journal per source document, by API	Every GL line traces to its invoice, bill or receipt
App stack	Next.js on Postgres, in your cloud account	Standard parts; every record in a database you own
Sign-in and roles	Your SSO; roles drawn from a segregation-of-duties matrix	Controls exist from day one, not after the audit
Currency	One functional currency per entity, daily FX rates	Revaluation and intercompany stay mechanical
Reporting	Metabase on a warehouse copy	Finance gets SQL without a seat fee

Compliance: build SOC 2-ready from the first commit. The audit log, permission controls and segregation of duties belong in the data model; write the control documentation as you build, not after.

Model the ledger first. *The rest hangs off it.*

Model these before writing any screen. Keep NetSuite's internal ID on every imported row, so the import can run twice and any balance traces back to its source.

TABLE	KEY COLUMNS
`entities`, `periods`	id, name, functional_currency, gl_company_id; periods: entity_id, starts, ends, status (open, closing, closed), closed_by, closed_at
`accounts`	id, entity_id, number, name, type, gl_account_id, active; the mapping from items and tax codes in `posting_rules`
`customers`, `vendors`	id, netsuite_id, name, parent_id, terms, credit_limit, tax_code, currency, custom (jsonb)
`items`	id, netsuite_id, sku, type (inventory, service, non-inventory), costing method, tracking (lot, serial), income, COGS and asset accounts
`documents`	id, netsuite_id, type (sales order, invoice, PO, receipt, bill, payment), entity_id, party_id, status, currency, fx_rate, date, period_id; lines in `document_lines`
`stock_moves`	item_id, location_id, lot or serial, quantity, unit_cost, document_id, at
`approvals`	document_id, step, approver_id, decision, comment, at
`journals`, `journal_lines`	id, entity_id, period_id, document_id, status (pending, posted, failed), gl_journal_id, posted_at; lines: account_id, debit, credit, currency, fx_rate, department, class, location
`fx_rates`	date, from_currency, to_currency, rate, source
`rev_schedules`	contract_id, obligation, amount, starts, ends, recognized_to_date; one journal per monthly run
`audit_log`	table, row_id, field, old_value, new_value, user_id, at

TWO RULES THAT SAVE A REBUILD

Posted means permanent: correct a posted journal with a reversing entry, never an edit, and let nothing post into a closed period. **Every journal line carries its source document** and the GL's journal ID: that is how an auditor gets from the trial balance to the invoice and back.

Up to 18 weeks, *pilot included.*

This is the order we build in. Each step ends with something your controller can test, and the last six weeks run one business unit on it, through a real month-end close, before anyone else moves.

WEEKS	BUILD	DONE WHEN
1–2	**Audit and model.** List every record type, custom field, saved search, SuiteScript and SuiteFlow in use. The tables, SSO and roles. Export through the SuiteCloud APIs into staging.	Counts in staging match NetSuite
3–4	**The ledger feed.** Accounts mapped, posting rules, the journal API to QuickBooks Enterprise or Sage Intacct, the posting queue, the audit log.	A test invoice posts a balanced journal and drills back
5–6	**Order to cash.** Customers, sales orders, invoices, collections, multi-currency, revenue schedules (ASC 606).	AR aging matches NetSuite
7–8	**Procure to pay.** Vendors, POs, receiving, bills, matching, payment approvals.	AP aging matches NetSuite
9–10	**Inventory.** Locations, lots, serials, stock moves, costing, reorder points.	Inventory valuation matches NetSuite
11–12	**Close and reporting.** The close checklist, FX revaluation, reconciliations, P&L, balance sheet, cash flow, dashboards. SuiteScripts translated. A full migration rehearsal.	A month closes in staging and ties to NetSuite
13–18	**Pilot and cutover.** One business unit for six weeks, through a month-end in both systems; the auditor walkthrough; a delta export; then everyone moves.	Your CFO and auditor sign off

AFTER CUTOVER

Keep NetSuite **read-only until your renewal date**, then cancel with a full export kept. Last year's history stays where the auditors can find it, and nobody is in a hurry.

Auditors test the GL. *Feed it cleanly.*

Orders, inventory and approvals are ordinary software. What decides whether finance trusts the rebuild is the ledger: every business event lands in the GL as a balanced journal, every period closes on schedule, and the auditor can trace any balance to its source. Auditors don't care what feeds the GL; they care that the GL is auditable.

1. **Map before you post.** Every item, tax code, customer class and expense type maps to a GL account in the posting-rules table, not in code branches. The controller signs the mapping before the first journal posts.
2. **Post balanced, and only once.** Check that debits equal credits before sending. Use the source document's ID as the idempotency key, so a retry never posts twice. Failed postings land in a queue finance can see, with the reason.
3. **Run the close as a workflow.** Cut-off for shipments and receipts, accruals, revenue schedules, FX revaluation, then reconcile AR, AP and inventory to their control accounts. Lock the period in the ERP and the GL together.
4. **Put revenue on schedules (ASC 606).** One deferred-revenue schedule per performance obligation; a contract change starts a new one; the monthly run posts the journals. Design it to your contracts, not to a generic module.
5. **Build the controls the auditor tests.** An audit log on every change: who, what, when, the old value and the new. Roles and permissions. Segregation of duties: whoever sets up a vendor cannot approve its payment, and whoever raises a PO cannot approve it.

Close one month in both systems

During the pilot, close the same month in NetSuite and in the rebuild. The trial balances must tie, account by account, before anyone else moves. It is the strongest evidence you can hand an auditor.

Walk the auditor through it early

Show the posting rules, the reconciliations, the audit log and the duties matrix before the pilot ends. Your CFO and auditor sign off as part of pilot acceptance, not after go-live.

SOX, GAAP and IFRS

The tests read three things: an audit trail on every transaction, permission controls and segregation of duties. Build all three in from the first week and the compliance posture matches NetSuite's or exceeds it.

Move it, launch it, *run it.*

Migration

- ☐ Export through the SuiteCloud SOAP and REST APIs: customers, vendors, items, transactions, custom fields
- ☐ Keep NetSuite's internal ID on every row
- ☐ Cut over at a period end, once NetSuite has closed that month
- ☐ Bring open invoices, bills, POs and sales orders live
- ☐ Bring full history across for reporting and audit
- ☐ Count stock by location, lot and serial at cutover
- ☐ Translate the SuiteScripts still in use; retire the rest
- ☐ Tie AR and AP aging, inventory valuation and the trial balance to NetSuite at the cutover date

Go-live

- ☐ A pilot month closed in both systems, trial balances tied
- ☐ CFO and auditor sign-off at pilot acceptance
- ☐ Users signing in through your SSO, roles checked against the duties matrix
- ☐ The posting queue empty, with alerts on failures
- ☐ Backups proven by a restore
- ☐ Rollback ready: NetSuite read-only, not cancelled

Running it

- ☐ **Always on:** posting-queue and error alerts, security patches
- ☐ **Daily:** FX rates loaded, failed postings cleared
- ☐ **Monthly:** the close checklist, reconciliations, revenue schedules
- ☐ **Quarterly:** an access review against the duties matrix, and the running cost against the NetSuite bill you replaced
- ☐ **Yearly:** the audit, and a restore drill

Who owns it

Name one person before launch. The ERP is where the month closes: someone has to answer when a journal fails to post on the last day of the month.

Five ways a rebuild *stalls.*

1. Switching ERPs, or tuning the one you have

SAP Business One and Microsoft Dynamics 365 are competent, at NetSuite-tier pricing, with 9–18 month migrations and the same per-user economics. NetSuite partners charge $500K–$3M for a better-tuned NetSuite. Either way the bill stays.

2. Bolting apps onto the accounting system

Sage Intacct is excellent accounting at $40K–$80K a year, but it does not run orders, inventory or procurement. QuickBooks Enterprise plus five bolt-on apps works below $30M revenue; above $50M, the integrations get unwieldy.

3. Self-hosting Odoo and calling it done

Odoo Community is free; Odoo Enterprise's hosted pricing is comparable to NetSuite. Production-grade operations, custom modules and support are still real engineering, and the running of it is yours.

4. Rebuilding everything

Manufacturing, consolidation across 5+ entities and the SuiteApp long tail are where schedules slip. Rebuild the 15–25% in daily use; leave the rest to specialist tools.

5. Skipping the pilot close

Six weeks on one business unit, through a real month-end, finds what a test plan misses: the custom field finance reads every Monday, the approval nobody wrote down, the account that does not tie.

And one that is not a mistake: negotiating

NetSuite discounts enterprise contracts 10–25% on multi-year, multi-module commits. Do it if the rebuild is a year away; it captures less than a quarter of what owning the ERP does.

RATHER HAVE IT BUILT?

We build this exact plan. *Fixed price.*

The same scope, architecture and 14–18 weeks, delivered by a senior team that has shipped 250+ products.

$53,100 fixed up to 100 users on a single entity, $76,600–$115,600 for 100–250 users, $140,600–$218,700 above 250, agreed before we start

---

Live in 14–18 weeks, with a six-week pilot on one business unit before cutover

---

Every NetSuite record moved, with full history and internal IDs; SuiteScript translated to TypeScript or Python

---

Your GL stays on QuickBooks Enterprise or Sage Intacct; your CFO and auditor sign off at pilot acceptance

---

90-day warranty, and the code and every record are yours

We can run it for $12,000–$28,000 a year: hosting and maintenance

alliedbiz.tech/rebuild/netsuite-alternative

Allied BizTech · 15+ years · ISO 9001:2015 · 250+ products shipped. NetSuite, QuickBooks and Sage Intacct are trademarks of their owners, named here for identification only; this guide is independent and not endorsed by any of them. NetSuite prices as of May 2026.