

SALESFORCE OPERATIONS • FIELD GUIDE

Fixing Salesforce technical debt with *AI*.

Finding, pricing, and paying down the invisible debt inside your Salesforce org. In ninety days, without breaking production.

**Neil Sarkar**

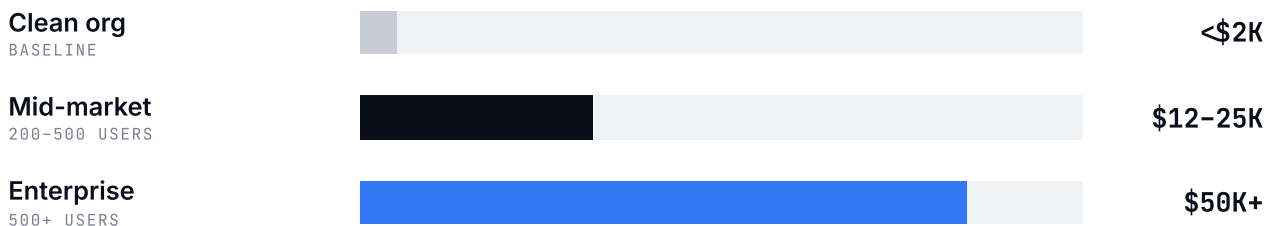
CTO & CO-FOUNDER, CLIENTELL

10 CHAPTERS • 30 MIN READ
WORKSHEETS • AI PROMPTS • SOQL

EXHIBIT • BY THE NUMBERS

USD PER MONTH

The median monthly cost of unmanaged technical debt.



SOURCE • 100+ CLIENTELL ORG AUDITS

10

CHAPTERS

05

WORKSHEETS

15

AI PROMPTS

15

SOQL QUERIES

INDEPENDENTLY AUDITED

SOC 2 • HIPAA • ISO 27001 • GDPR • SALESFORCE PARTNER

GETCLIENTELL.COM
© 2026 CLIENTELL AI

The weight you're *carrying*.

67%

OF SALESFORCE ADMINS SAY
TECHNICAL DEBT IS THEIR
NUMBER-ONE DAILY CHALLENGE,
MORE THAN REPORTS, USER
REQUESTS, OR FLOWS.

Not reports. Not user requests. Not even Flows.
Technical debt: the invisible, compounding mess
that makes every "simple change" take three times
longer than it should.

Here is what the average Salesforce org looks like right now:

- **200+ custom fields** per object: half unused, nobody knows which half.
- **30+ automations** on the Opportunity object: some conflicting, some redundant, some nobody remembers building.
- A **permissions model** that was "temporary" three years ago and is now load-bearing infrastructure.
- **15+ Process Builders** still active alongside the Flows that were supposed to replace them.
- **30,000+ stale reports** that nobody has opened in over a year.

Technical debt isn't just annoying. It costs real money. Slower deployments. Broken automations at 2 AM. Bad data poisoning pipeline forecasts. Admin burnout and turnover. Projects that should take a week taking a month.

And the worst part? Leadership doesn't see it. They see a CRM that "works." They see an admin who seems busy but can't explain why a "simple" field addition took two sprints.

*"This guide makes the invisible **visible**. For admins, the frameworks, queries, and playbooks to audit and fix it. For leadership, why "simple changes" aren't simple, and what it is costing you.*

\$200K

Average CRM spend running at an estimated 60% efficiency because of accumulated shortcuts.

70%

Of admin time in high-debt orgs spent on maintenance, not building the future.

40–80^h

Manual audit time replaced by a two-minute agent scan of your whole org.

What's *inside*.

TEN CHAPTERS
FOUR PARTS
ONE 90-DAY PLAN

I • 01	What Is Salesforce Technical Debt Definition, the five types, and why leadership should care.	04
I • 02	The True Cost Calculator Stop guessing. A five-category worksheet to quantify the drain.	06
II • 03	The Audit: Where to Look SOQL queries, the Tooling API, and AI prompts for every layer.	08
II • 04	Automation Debt: The Silent Killer Process Builders, Workflow Rules, and the order-of-execution trap.	10
II • 05	Field & Metadata Debt The 200-field problem, and how to delete safely without breaking anything.	12
III • 06	Permission Debt: The Security Time Bomb The profile-to-permission-set migration nobody has finished.	15
III • 07	Data Debt: The Foundation Problem Duplicates, inconsistency, completeness, staleness. Fix the substrate.	17
III • 08	The AI-Powered Cleanup Playbook Tools, prompts, and an hour-by-hour schedule per debt type.	19
IV • 09	The 90-Day Debt Reduction Plan Audit, quick wins, structural fixes, then a maintenance cadence.	21
IV • 10	Making the Business Case Three narratives that work, and the one-page executive brief.	23

WORKSHEETS

5

AI PROMPTS

15

SOQL QUERIES

15

• DEFINITIONS • 5 TYPES • LEADERSHIP

What is Salesforce *technical debt*?

Every shortcut, workaround, and "temporary" fix that makes your org harder to change tomorrow than it is today. Like credit-card debt. It compounds. Unlike credit-card debt, most executives cannot see it on the balance sheet.

The five types of Salesforce technical debt.

I – V

FIVE CATEGORIES. EVERY
AUDIT YOU RUN IN PART II
MAPS BACK TO ONE OF THEM.

I • AUTOMATION DEBT

Process Builders that should have been Flows two years ago. Workflow Rules firing alongside Flows, creating conflicts. The average Enterprise org runs **three to four automation tools simultaneously** on key objects.

II • FIELD DEBT

Fields built for a project that ended. Picklist values untouched since 2019. The average org has **200+ custom fields on Opportunity alone**, and 30–50% are unused.

III • PERMISSION DEBT

Profiles with *Modify All Data* "just in case." Only **34% of orgs** have started the profile-to-permission-set migration Salesforce is forcing.

IV • DATA DEBT

Duplicates, inconsistent formatting, incomplete records, stale 2022 data still in pipeline reports. **51.2%** of admins say deduplication is a top struggle.

V • DOCUMENTATION DEBT

No record of why automations exist. When the admin who built it leaves, the knowledge walks out the door. **78%** of orgs have no formal documentation process.

EXHIBIT 1.1 • THE COMPOUND-INTEREST ANALOGY, IN HOURS

One shortcut, compounded over four years.

YEAR	"QUICK FIX" APPROACH	"DO IT RIGHT" APPROACH
Y1	10 min · to build	45 min · to build
Y2	2 hrs · to modify (conflicts)	15 min · to modify
Y3	8 hrs · to rearchitect	15 min · to modify
Y4	"We need a consultant" · \$15K	15 min · to modify
Total	\$15K + 10 hrs	1.5 hrs

TRANSLATION FOR LEADERSHIP

"We're paying for a \$200,000/year CRM that runs at 60% efficiency because of accumulated shortcuts. Every 'simple change' requires navigating legacy configurations nobody fully understands. We can fix it, but it needs dedicated time, not just more feature requests piled on top."

• WORKSHEET • 5 CATEGORIES • THE TOTAL

The true cost *calculator.*

The biggest reason technical debt persists is that nobody quantifies it. "Things are slow" doesn't get budget. "\$47,000 a year in wasted admin hours" does.

Stop guessing. Start measuring.

Five categories drive the monthly cost of technical debt. Use real numbers. Even rough estimates beat none. Add the five together for your monthly figure.

WORKSHEET 2.1 • MONTHLY TECHNICAL-DEBT COST

COST CATEGORY	WHAT DRIVES IT	INDUSTRY BENCHMARK	YOUR \$ / MO
1 • Deployment time	Excess hours per deploy spent on conflicts & workarounds	3–5× longer than a clean org	\$_____
2 • Automation failures	Fix hours + after-hours emergencies + revenue impact	4× more failures with 3+ tools per object	\$_____
3 • Data quality	Bad-data revenue loss + manual cleanup hours	12% of revenue on average (Gartner)	\$_____
4 • Admin & opportunity cost	Overtime + the value of projects you can't ship	60–70% of time on maintenance, not building	\$_____
5 • Risk & compliance	Breach / audit-failure exposure × probability	SOX / HIPAA / GDPR findings	\$_____

Your monthly technical-debt cost

\$_____

\$12K–25K

Median monthly cost for a mid-market org (200–500 users), across 100+ Clientell audits.

\$50K+

Typical monthly cost for enterprise orgs (500+ users). Most leadership teams have no idea.

FOR LEADERSHIP

If someone hands you this worksheet filled out, take it seriously. They just quantified something that has been invisibly draining your ROI. The fix is almost always cheaper than the cost of doing nothing.

• SQL • TOOLING API • AI PROMPTS

The audit: where to *look*.

Don't try to fix everything at once. First, know what you're dealing with. Think of it as a medical checkup: before treatment, you need a diagnosis.

The six layers to audit, and the exact query for each.

01 Unused fields

Zero- or low-population custom fields cluttering every layout.

02 Conflicting automations

Multiple tools firing on one object in unpredictable order.

03 Redundant Process Builders

Legacy by definition since Salesforce retired them in 2023.

04 Over-permissioned profiles

Anyone but admins with Modify / View All Data.

05 Orphaned metadata

Fields off all layouts, inactive Flows, 0%-coverage Apex.

06 Stale reports

30,000+ reports nobody has opened in over a year.

SQL · FIND UNUSED FIELDS

```
SELECT COUNT(Custom_Field_1__c), COUNT(Custom_Field_2__c)
FROM Opportunity
WHERE CreatedDate = LAST_N_DAYS:180
-- COUNT(field) returns non-null values · check 25 fields per query
```

AI PROMPT

I have [N] custom fields on [OBJECT] with these population rates over 6 months:
[PASTE DATA]

Categorize each as ACTIVE / LOW USAGE / LIKELY UNUSED / CRITICAL REGARDLESS.
For each LIKELY UNUSED field: safe to delete, archive first, or investigate,
and the exact dependency check to run before deleting anything.



PRO TIP

Never delete a report or field without checking dependencies first: dashboards, subscriptions, Flow references. Always archive to a "Deprecated" folder before deleting. It's insurance.

• PROCESS BUILDERS • WORKFLOW RULES • ORDER OF EXECUTION

Automation debt: the *silent killer*.

If fields are the atoms of your org, automations are the nervous system. In most orgs that nervous system has been rewired so many times nobody knows how a signal gets from A to B.

One field change. Seven things fire.

A rep moves an Opportunity to **Closed Won**. Here's the cascade we see in nearly every audit:

- 01 A before-save record-triggered **Flow** updates the Close Date.
- 02 A **Workflow Rule** emails the account team.
- 03 A **Process Builder** updates a rollup on the Account.
- 04 An after-save **Flow** creates an onboarding Task.
- 05 An **Apex trigger** calculates commission.
- 06 The Account update (from #3) fires **another Flow** that touches the Contact.
- 07 A **validation rule** on the Contact blocks the update because a required field is missing.

✗ **Result:** "An unhandled fault has occurred." The rep pings you on Slack. You spend **90 minutes** tracing the cascade. That's automation debt.

PROCESS BUILDERS

Retired in 2023, with zero new investment. They run in the same transaction as Flows but in a different order, can't be version-controlled, and are invisible to many debug tools. 10+ active? Migrate them.

WORKFLOW RULES

Deprecated since 2022, still running in nearly every org. Email alerts that should be Flows, field updates that collide with Flow updates, outbound messages to integrations replaced years ago. Triage and deactivate.

FOR LEADERSHIP

Your team is running the equivalent of Windows XP alongside Windows 11. Every month you wait to migrate adds risk and cost. You're one Salesforce release away from a deprecated Process Builder breaking in production with no warning.

• FIELDS • PICKLISTS • METADATA SPRAWL

Field & metadata debt: the *200-field* problem.

Every field starts with good intentions. A VP needs one for X. A consultant builds thirty for a project that gets shelved. Nobody ever deletes them, and you end up with objects 300 fields deep.

What field debt actually looks like.

Page layouts that scroll forever. Users who don't know which "Revenue" field to use: *Annual?* *Total?* *Revenue FINAL v2?* Here's the diagnosis.

SYMPTOM	IMPACT
200+ custom fields on Opportunity	Confusion, slow page loads, cluttered layouts
30–50% of fields under 5% populated	Wasted real estate, misleading reports
Duplicate fields, same data different names	Inconsistent reporting, conflicting data
Fields with no description	New admins can't understand the data model
Picklist values untouched 2+ years	Bad data and "Other" text-field workarounds
Fields from ended consultant projects	Nobody knows if they're safe to delete

METADATA SPRAWLS TOO

Record Types Unused but locked by profiles	Page Layouts v2 · FINAL · NEW · COPY	Custom Objects POCs that went nowhere	Custom Tabs Unclicked for years
--	--	---	---



PRO TIP · THE SAFE DELETE

Never delete a field the day you decide it's unused. Move it off all page layouts first. Wait 2–4 weeks. If nobody notices, it's safe. If someone panics, add it back to the layout. No data lost.

HALFWAY

You're halfway *through.*

If you're mentally counting the custom fields on your Opportunity object right now, you're exactly who this guide is for. The next five chapters cover permissions, data quality, the specific AI tools and playbooks to fix everything, a 90-day plan, and how to sell it to leadership.

• PROFILES • PERMISSION SETS • COMPLIANCE

Permission debt: the security *time bomb*.

The most dangerous type of technical debt, because it's not just inefficient. Someone needed access urgently. The quick fix was a broad permission. Nobody removed it. Repeat 200 times over five years.

What we find in nearly every audit.

72%

of orgs have a non-admin profile with Modify All Data

23

average permission sets, with significant overlap

34%

have started the profile-to-permission-set migration

89%

have no regular cadence for permission reviews

SOQL • HIGHEST-RISK USERS

```
SELECT Id, Name, Profile.Name, LastLoginDate
FROM User
WHERE IsActive = true AND Profile.PermissionsModifyAllData = true-- anyone but System
Administrators here = permission debt
```

THE PROFILE → PERMISSION-SET MIGRATION

01 Discovery

1 WEEK

Export profiles, map which permissions are actually used.

02 Design

1-2 WEEKS

Permission-set groups by job function; an access matrix.

03 Pilot

2 WEEKS

Migrate your simplest profile; run old & new side by side.

04 Rollout

4-8 WEEKS

One profile at a time, 48-hour validation, rollback per step.

FOR LEADERSHIP

Permission debt is a compliance risk. If an auditor asks "who has access to what?" and gets a shrug, that's a finding. The migration to permission sets is inevitable. Salesforce is forcing it. The only question is whether you do it on your timeline or theirs.

• DUPLICATES • FORMATTING • COMPLETENESS • STALENESS

Data debt: the *foundation* problem.

You can have perfect automations, clean metadata, and tight permissions. But if the data underneath is a mess, none of it matters. Bad data in, bad decisions out.

The four types of data debt.

01 • DUPLICATES

"Acme Inc", "Acme, Inc.", "ACME", "Acme Corporation": four records, four owners, four pipelines. Your total-pipeline report is 20–30% inflated.

02 • INCONSISTENT FORMATTING

"United States" vs "US" vs "USA" vs "U.S." Every inconsistency breaks every filter, report, and automation that relies on an exact match.

03 • INCOMPLETE RECORDS

Leads without email. Accounts without industry. Every gap is a missed insight, a broken trigger, or a rep looking up what should already be there.

04 • STALE DATA

Contacts who left two years ago. Deals stuck in "Negotiation" since Q2 2024. A report showing \$5M of "active" deals that are dead is worse than no report.

51%

of admins say dedup is a top struggle

30%

of CRM data goes stale every year

25%

duplicate rate across Accounts & Contacts

91%

of CRM data is incomplete in some field

FOR LEADERSHIP

Bad data isn't an admin problem. It's a revenue problem. When your pipeline says \$10M but 20% is duplicate or stale, you're deciding on a number that's off by \$2M. Cleanup costs a fraction of bad forecasting.

• TOOLS • PROMPTS • STEP-BY-STEP

The AI-powered cleanup *playbook*.

You've identified the debt. Now fix it. This is your toolbox: the tools, the prompts, and the approaches that turn a 40-hour audit into a two-minute scan.

The toolbox: what each is for.

General AI ChatGPT, Claude, Gemini. Analysis, planning, queries, migration docs. Can't touch your org.	Specialized Hubbl, Sweep, Elements.cloud, OrgCheck. Benchmarking & dependency analysis.	Native Optimizer, Field Trip, Duplicate Rules. Free first pass, limited depth.	Clientell agents Connect to the org, scan everything, return a costed plan.
---	---	---	---

EXHIBIT 8.1 • THE SAME AUDIT, TWO WAYS

THE MANUAL APPROACH	40-80 HRS	WITH CLIENTELL AI	2 MIN
Export SOQL results & metadata → paste into ChatGPT or Claude → get a categorized plan → validate against org knowledge → execute in a sandbox → test with real users → monitor for 48-72 hours.		Connect to your org → the agent scans every field, automation, permission and dependency → returns a prioritized, dollar-costed remediation plan. No exports. No prompt engineering. No guessing about dependencies.	

THE DIFFERENCE

General-purpose AI analyzes what you give it. Our agents see everything, including the standard-field dependencies the Tooling API doesn't return and the cascading conflicts that span three or more objects. Always validate in a sandbox; AI is excellent at analysis, not infallible.

• AUDIT • QUICK WINS • STRUCTURE • CADENCE

The 90-day debt-reduction *plan.*

Month one isn't about fixing anything. It's about understanding what you're dealing with and building the case. Then quick wins, then structure. Then a cadence so it never drifts back.

The plan, month by month.

01

Month 1

AUDIT & PRIORITIZE

- ☐ Run Optimizer; count automations & fields per key object
- ☐ Field population + permission audit queries
- ☐ Fill out the Cost Calculator worksheet
- ☐ Present findings; secure dedicated time

Deliverable: prioritized debt list with \$ costs + buy-in.

02

Month 2

QUICK WINS

- ☐ Deactivate redundant Workflow Rules & Process Builders
- ☐ Hide unused fields from layouts; archive stale reports
- ☐ Strip Modify All Data from non-admin profiles
- ☐ Turn on Matching/Duplicate Rules; merge worst clusters

Deliverable: 15–25 wins; fewer failures, faster deploys.

03

Month 3

STRUCTURAL

- ☐ Migrate top 3–5 critical Process Builders to Flows
- ☐ Consolidate automations on your busiest object
- ☐ Pilot the profile → permission-set migration
- ☐ Build a data-quality score + dashboard

Deliverable: simpler automation, pilot live, quality monitored.

AFTER 90 DAYS • THE MAINTENANCE CADENCE

WEEKLY • 15 MIN

Error logs; stuck deployments

MONTHLY • 2 HR

Field population; merge dupes; quality dashboard

QUARTERLY • ½ DAY

Permission & license audit; refresh the calculator

PER RELEASE • 4 HR

Scan release notes for deprecations; test in sandbox

• MONEY • RISK • SPEED • OBJECTIONS

Making the business *case*.

The hardest part of fixing technical debt isn't the technical work. It's getting permission to do it. Translate the problem into the language leadership understands: money, risk, and speed.

Three narratives that get it funded.

THE MONEY STORY

"\$X/month on workarounds and overtime. For X hours over 90 days, we cut it 40–60%."

CFOs think in ROI. Give them a number they can model.

THE RISK STORY

"X users can modify all data. We run tools Salesforce is sunsetting. An auditor would find issues today."

Execs who won't fund improvements will fund risk mitigation.

THE SPEED STORY

"Every project runs 30–50% slower. Fix the foundation and we ship faster on everything after."

Leadership cares about velocity. Tie cleanup to the roadmap.

HANDLING THE THREE OBJECTIONS

"We don't have time for this."

We don't have time NOT to. Every week costs hours of workarounds; the cleanup pays for itself in weeks.

"Can't we just hire a consultant?"

\$15–30K for this scope. Do 80% in-house with AI; bring in help for the 20% that needs an expert.

"What if we break something?"

Every change is sandbox-tested with a rollback per step. The bigger risk is a deprecated Process Builder breaking with no warning.

FOR ADMINS

You're not asking for a favor. You're proposing a project with clear ROI. Bring numbers, not complaints. The admin who quantifies the problem and proposes the solution is the one who gets promoted to Architect.

• QUERIES • PROMPTS • COPY-PASTE READY

The *toolkit*.

Every SOQL query and AI prompt referenced in this guide, organized by chapter. Copy, paste, and run against your own org.

SOQL query library.

Run in Developer Console, Workbench, or any Tooling API client.

CH 03 • THE AUDIT

SOQL • FIND FIELDS WITH ZERO POPULATION

```
SELECT COUNT(Custom_Field_1__c), COUNT(Custom_Field_2__c), COUNT(Custom_Field_3__c)
FROM Opportunity
WHERE CreatedDate = LAST_N_DAYS:180
```

TOOLING API • GET ALL CUSTOM FIELDS ON AN OBJECT

```
SELECT DeveloperName, Description, CreatedDate, LastModifiedDate
FROM CustomField
WHERE TableEnumOrId = 'Opportunity'
ORDER BY CreatedDate ASC
```

TOOLING API • LIST ALL AUTOMATIONS ON AN OBJECT

```
-- Flows
SELECT Definition.DeveloperName, ProcessType, Status, TriggerType
FROM FlowDefinitionView
WHERE ProcessType IN ('AutoLaunchedFlow','RecordTriggerFlow','Workflow')
AND Status = 'Active'

-- Workflow Rules
SELECT Name, TableEnumOrId
FROM WorkflowRule
WHERE TableEnumOrId = 'Opportunity'

-- Process Builders (legacy)
SELECT Definition.DeveloperName, ProcessType
FROM FlowDefinitionView
WHERE ProcessType = 'Workflow'
AND Status = 'Active'
```

SOQL • USERS WITH MODIFY ALL DATA

```
SELECT Id, Name, Profile.Name, LastLoginDate
FROM User
WHERE IsActive = true AND Profile.PermissionsModifyAllData = true
```

TOOLING API • PERMISSION SETS GRANTING DANGEROUS PERMISSIONS

```
SELECT Label, PermissionsModifyAllData, PermissionsViewAllData,
PermissionsManageUsers, PermissionsAuthorApex
FROM PermissionSet
WHERE PermissionsModifyAllData = true OR PermissionsViewAllData = true
```

SOQL • REPORTS NOBODY USES

```
SELECT Name, DeveloperName, FolderName, LastRunDate, LastViewedDate
FROM Report
WHERE LastViewedDate < LAST_N_DAYS:365
ORDER BY LastViewedDate ASC
```

CH 04 • AUTOMATION

TOOLING API • ACTIVE WORKFLOW RULES BY OBJECT

```
SELECT Name, TableEnumOrId
FROM WorkflowRule
WHERE ManageableState = 'unmanaged'
```

CH 05 • FIELDS

TOOLING API • CUSTOM FIELDS WITH METADATA

```
SELECT DeveloperName, Description, NamespacePrefix, CreatedDate,
       LastModifiedDate, Metadata
FROM CustomField
WHERE TableEnumOrId = 'Opportunity'
AND NamespacePrefix = null ORDER BY CreatedDate ASC
```

SQL • MEASURE FIELD POPULATION

```
SELECT
  COUNT(Id) Total_Records,
  COUNT(Field_1__c) Field_1_Populated,
  COUNT(Field_2__c) Field_2_Populated,
  COUNT(Field_3__c) Field_3_Populated
FROM Opportunity
WHERE CreatedDate = LAST_N_DAYS:365
```

TOOLING API • FIELD DEPENDENCIES

```
SELECT MetadataComponentName, MetadataComponentType, RefMetadataComponentName
FROM MetadataComponentDependency
WHERE RefMetadataComponentName = 'Account.Custom_Field__c'
```

CH 06 • PERMISSIONS

SQL • FIND YOUR HIGHEST-RISK USERS

```
-- Users with Modify All Data (should be admins ONLY)
SELECT Id, Name, Profile.Name, UserRole.Name, LastLoginDate
FROM User
WHERE IsActive = true AND Profile.PermissionsModifyAllData = true ORDER BY Profile.Name

-- Users with View All Data
SELECT Id, Name, Profile.Name
FROM User
WHERE IsActive = true AND Profile.PermissionsViewAllData = true
```

SQL • IDENTIFY OVER-PERMISSIONED PROFILES

```
-- Object permissions: who has Modify All Records?
SELECT Parent.Profile.Name, SubjectType,
       PermissionsModifyAllRecords, PermissionsViewAllRecords
FROM ObjectPermissions
WHERE PermissionsModifyAllRecords = true AND Parent.IsOwnedByProfile = true
```

TOOLING API • FIND PERMISSION-SET OVERLAP

```
-- Permission sets and their assignments
SELECT PermissionSet.Label, Assignee.Name, Assignee.Profile.Name
FROM PermissionSetAssignment
WHERE PermissionSet.IsOwnedByProfile = false ORDER BY Assignee.Name
```

CH 07 • DATA

SOQL • MEASURE RECORD COMPLETENESS

```
-- Field completeness on Leads
SELECT
  COUNT(Id) Total,
  COUNT(Email) Has_Email,
  COUNT(Phone) Has_Phone,
  COUNT(Company) Has_Company,
  COUNT(Industry) Has_Industry,
  COUNT(LeadSource) Has_Source
FROM Lead
WHERE IsConverted = false AND CreatedDate = LAST_N_DAYS:365
```

SOQL • FIND STALE RECORDS

```
-- Opportunities stuck in mid-funnel
SELECT Id, Name, StageName, LastModifiedDate, Owner.Name
FROM Opportunity
WHERE IsClosed = false AND LastModifiedDate < LAST_N_DAYS:90
ORDER BY LastModifiedDate ASC

-- Contacts with bounced emails
SELECT Id, Name, Email, Account.Name
FROM Contact
WHERE HasOptedOutOfEmail = true OR Email = null
```

AI prompt library.

Paste into Clientell AI. Replace [bracketed] values with your data.

CH 03 • THE AUDIT

CATEGORIZE UNUSED FIELDS

AI PROMPT

I have [NUMBER] custom fields on [OBJECT]. Here are the field names and their population rates over the last 6 months:

[PASTE YOUR DATA]

Categorize each field as:

1. ACTIVE (>50% populated, used regularly)
2. LOW USAGE (10-50% populated, may be legacy)
3. LIKELY UNUSED (<10% populated, candidate for removal)
4. CRITICAL REGARDLESS (even low usage is expected -- e.g., edge case fields)

For each LIKELY UNUSED field, suggest: safe to delete, archive first, or investigate further. Include the dependency check I should run before deleting anything.

DETECT CONFLICTING AUTOMATIONS

AI PROMPT

Here are all active automations on our [OBJECT] object:

Flows: [LIST NAMES AND TRIGGER TYPES]

Process Builders: [LIST NAMES]

Workflow Rules: [LIST NAMES]

Apex Triggers: [LIST NAMES]

For each pair that could conflict:

1. Identify the potential conflict (field updates, DML, order of execution)
2. Rate the risk (HIGH/MEDIUM/LOW)
3. Recommend which should be consolidated or removed
4. Flag any that should NEVER run together

Also map out the likely order of execution for a record update on this object.

AUDIT REDUNDANT PROCESS BUILDERS & WORKFLOW RULES

AI PROMPT

I have [NUMBER] active Process Builders in my org. Here they are:

[LIST EACH WITH: Name, Object, What it does (brief)]

For each one, provide:

1. Can this be directly migrated to a Flow? (YES/NO/PARTIAL)
2. What's the migration complexity? (SIMPLE/MODERATE/COMPLEX)
3. Are there any gotchas in the migration (invocable actions, cross-object updates, etc.)?
4. Priority to migrate (based on object importance and conflict risk)

Create a migration plan ordered by: quick wins first, then critical-path items, then nice-to-haves.

INVESTIGATE ORPHANED METADATA

AI PROMPT

Help me identify orphaned metadata in my Salesforce org. Here's what I've extracted:

Custom fields not on any page layout: [LIST]
 Inactive Flows: [LIST WITH LAST MODIFIED DATE]
 Apex classes with 0% coverage: [LIST]
 Reports not viewed in 12+ months: [COUNT]
 Email templates not used in 12+ months: [COUNT]

For each category:

1. What's safe to delete immediately?
2. What needs dependency checking first?
3. What should be archived (not deleted)?
4. What's the process for safe removal?

Give me a prioritized cleanup list starting with the lowest-risk, highest-impact items.

CH 04 • AUTOMATION

DETECT AUTOMATION CONFLICTS

AI PROMPT

Here is every automation on our Opportunity object:

[PASTE YOUR TABLE]

Analyze this for:

1. ORDER OF EXECUTION CONFLICTS: Which automations could fire in a problematic sequence?
2. FIELD COLLISIONS: Which automations update the same field(s)?
3. RECURSION RISK: Which could trigger each other in a loop?
4. REDUNDANCY: Which do overlapping or identical things?
5. LEGACY DEBT: Which use deprecated tools (Process Builder, Workflow Rules)?

For each issue found, rate it RED (will cause failures), YELLOW (may cause unexpected behavior), or GREEN (low risk but should be cleaned up).

Then recommend a target state: what should the automation landscape look like after cleanup?

BUILD A MIGRATION DECISION TREE

AI PROMPT

For each Process Builder, ask:

1. Does a Flow already do something similar? → CONSOLIDATE into the existing Flow
2. Is it simple (1-2 criteria, 1-2 actions)? → MIGRATE using the built-in migration tool
3. Is it complex (multiple criteria groups, cross-object)? → REDESIGN as a new Flow
4. Does it interact with Apex or other automations? → ARCHITECT holistically (don't migrate in isolation)
5. Is it for an object nobody uses anymore? → DELETE IT

PLAN A PROCESS BUILDER MIGRATION

AI PROMPT

I need to migrate this Process Builder to a Flow. Here are the details:

Name: [NAME]
 Object: [OBJECT]
 Criteria Groups: [DESCRIBE EACH]
 Actions per criteria group: [DESCRIBE EACH]
 Does it interact with other automations on this object? [YES/NO -- DESCRIBE]
 Does it call Apex Invocable Actions? [YES/NO]

Provide:

1. A Flow design (trigger type, entry criteria, elements needed)
2. Any gotchas in the migration (things that work differently in Flows vs. PB)
3. Testing plan (what to verify before deactivating the Process Builder)
4. Rollback plan (how to revert if something breaks)
5. Step-by-step migration instructions

TRIAGE WORKFLOW RULES

AI PROMPT

Here are all Workflow Rules in my org: [LIST]

For each, classify as:

- MIGRATE (has active business value, needs to become a Flow)
- DEACTIVATE (no longer needed, safe to turn off after verification)
- INVESTIGATE (unclear purpose, needs someone to confirm before touching)

Provide a safe deactivation checklist for each DEACTIVATE candidate.

CH 05 • FIELDS

FIELD TRIAGE & SAFE-DELETE PLAN

AI PROMPT

Here is a field inventory for our [OBJECT] object. For each field I'm including: name, description (if any), population rate, last modified date, and known dependencies.

[PASTE DATA]

Categorize each field into:

1. KEEP -- actively used, well-populated, has dependencies
2. CONSOLIDATE -- overlaps with another field, merge data and delete one
3. ARCHIVE -- low usage but might have historical value, hide from layouts but don't delete yet
4. DELETE -- zero usage, no dependencies, safe to remove
5. INVESTIGATE -- unclear purpose, needs business owner confirmation

For CONSOLIDATE candidates, identify which fields should be merged and suggest a data migration approach.

For DELETE candidates, provide a safe deletion checklist (dependency check, layout removal, Flow check, report check).

PLAN METADATA CLEANUP

AI PROMPT

Here is a summary of our Salesforce metadata inventory:

Custom Objects: [NUMBER] (list objects with <100 records)

Record Types per object: [LIST OBJECTS WITH 4+ RECORD TYPES]

Page Layouts: [NUMBER] (list any with "v2", "FINAL", "NEW", "OLD", "COPY" in the name)

Custom Tabs: [NUMBER]

Custom Apps: [NUMBER]

For each category, help me identify:

1. What's clearly safe to clean up?
2. What needs dependency checking first?
3. What's the recommended approach (delete, merge, archive)?
4. What's the order of operations (what to clean first to unblock cleaning other things)?

CH 06 • PERMISSIONS

ANALYZE PERMISSIONS

AI PROMPT

Here is our current permission structure:

Profiles in use: [LIST WITH USER COUNT PER PROFILE]

Permission Sets: [LIST WITH DESCRIPTION AND USER COUNT]

Permission Set Groups: [LIST IF ANY]

Known issues:

- [LIST ANY KNOWN OVER-PERMISSIONING]
- [LIST ANY COMPLIANCE REQUIREMENTS]

Please analyze and provide:

1. Which profiles have excessive permissions? (Compare to Salesforce's minimum access principle)
2. Which permission sets overlap significantly? (Candidates for consolidation)
3. Which users have permission combinations that create security risks?
4. A recommended target state: minimum profiles + permission set groups for each job function
5. A migration sequence that won't break access for active users

PLAN THE PERMISSION-SET MIGRATION

AI PROMPT

I need to migrate from profile-based permissions to permission sets. Here are my current profiles:

[LIST EACH PROFILE WITH: Name, User Count, Key Permissions, Objects Accessed]

Our compliance requirements: [SOX/HIPAA/GDPR/NONE]

Salesforce Edition: [EDITION]

Do we use Permission Set Groups yet? [YES/NO]

Create a phased migration plan including:

1. Which profile to migrate first (lowest risk, best pilot)
2. Permission Set Group design for each job function
3. The minimum-access profile template
4. Validation queries to confirm no access was lost
5. Rollback procedure for each step
6. Timeline estimate

CH 07 • DATA

AUDIT DATA QUALITY

AI PROMPT

I need to estimate my Salesforce duplicate rate. Here are my numbers:

Total Accounts: [NUMBER]

Total Contacts: [NUMBER]

Total Leads: [NUMBER]

I've exported a sample of 500 Accounts. Here are potential duplicate clusters (accounts with similar names):

[PASTE SAMPLE -- even 20-30 examples is enough]

Based on this sample:

1. What's my estimated duplicate rate?
2. What matching criteria should I use for dedup (name, domain, phone, address)?
3. What's the recommended approach: Salesforce native Matching Rules, third-party tool, or manual?
4. What's the merge strategy (which record is the "master")?
5. What data do I need to preserve from the duplicate before merging?

GENERATE STANDARDIZATION RULES

AI PROMPT

I need to standardize data in my Salesforce org. Here are the fields with inconsistency issues:

1. Country field: contains "US", "USA", "United States", "U.S.", "America", etc.
2. State field: mix of abbreviations and full names
3. Phone field: formats include (555) 555-1234, 555-555-1234, 5555551234, +1 555 555 1234
4. Industry picklist: users type in "Other" and put the real answer in a notes field

For each field:

1. Define the standard format we should use
2. Write the Salesforce formula or Flow logic to standardize incoming data
3. Provide a Data Loader approach for cleaning existing records
4. Suggest validation rules to prevent future bad data
5. Estimate the effort (hours) for each cleanup task

AI PROMPT

BUILD A DATA-QUALITY SCORE

Help me create a data quality scoring system for our Salesforce org.

Key objects: Account, Contact, Lead, Opportunity

Critical fields per object:

- Account: [LIST YOUR MUST-HAVE FIELDS]
- Contact: [LIST]
- Lead: [LIST]
- Opportunity: [LIST]

Design:

1. A completeness score (% of critical fields populated) per record
2. A formula field or Flow that calculates and stores this score
3. Thresholds (what score = "good", "needs attention", "critical")
4. A dashboard showing data quality trends over time
5. Alerts when quality drops below threshold

THIS GUIDE SHOWS WHAT TO FIX • FINDING EVERY ISSUE TAKES WEEKS

You know what to fix. Let the agent *fix it*.

This guide can't look at *your* org. Our AI can. It scans every field, automation, permission and dependency in about two minutes and returns a prioritized plan ranked by dollar impact: work that takes a human 40–80 hours.

START NOW • FREE

Start your free trial

Point Clientell's AI agents at your own org and watch the debt surface. No credit card required.

START FREE TRIAL →

app.clientell.ai

15 MINUTES • NO PITCH

Book a live org audit

We screen-share your org, run the scan, and walk you through what we find. You keep the results either way.

BOOK THE AUDIT →

cal.com/neils/virtual-coffee-with-neil

Neil Sarkar

CTO & CO-FOUNDER • CLIENTELL •
NEIL@GETCLIENTELL.COM

© 2026 CLIENTELL AI, INC.
SALESFORCE CONSULTING PARTNER



NEIL SARKAR PHOTOGRAPHED 2025

• WHO WROTE THIS

Neil *Sarkar*

CTO & CO-FOUNDER · CLIENTELL

150+ Salesforce implementations.
Six years in the ecosystem. Builds
AI agents that do Salesforce admin
work from plain English.

Before Clientell, scaled Salesforce
practices at two VC-backed
startups. Has personally audited
hundreds of orgs and seen every
flavour of technical debt
imaginable.

150⁺SALESFORCE
IMPLEMENTATIONS**6^{yr}**IN THE
ECOSYSTEM**10K⁺**NEWSLETTER
READERS[LINKEDIN →](#)[X/TWITTER →](#)[NEWSLETTER →](#)