

Restaurantology® SaaS Bookings Guide

Sales Ops frameworks and CRM tactics to define, track, and scale software bookings in restaurant tech

Restaurantology is a Go-to-Market (GTM) activation partner for B2B companies selling to restaurants. We help teams grow faster and sell smarter with real-time market intelligence and operational precision.

Learn more at restaurantology.io.

Copyright © 2025 by Restaurantology, LLC All rights reserved.

This playbook is for educational use only. You're encouraged to adopt and adapt the parts that help your team. Just don't resell, repackage, or redistribute it without permission. If you quote or share it (thank you), please credit Restaurantology and include a link.

Written by: Grant Gadoci

Questions? Contact hello@restaurantology.io.

Table of Contents

Preface	4
Intro & expectation setting	5
The perks and problems with SaaS bookings	7
Part 1 - Products, not promises	12
Part 2 - Consistency in counts	16
Part 3 - Booking boundaries	20
Part 4 - Set in stone	25
Final thoughts	30
Appendix	32
Additional resources	38
Note from the author	39

Preface

In the restaurant industry, change isn't just frequent. It's foundational. For B2B teams selling into this space, that volatility complicates every part of the Go-to-Market (GTM) motion.

Restaurantology was built to solve a clear problem: most teams selling to restaurants are flying blind. Our platform delivers real-time market intelligence to help them grow faster and sell smarter.

Since 2019, we've expanded beyond data to support dozens of restaurant tech companies through GTM consulting and fractional Sales Ops. These services help teams improve pipeline accuracy, optimize CRM architecture, and scale smarter.

At the core of our work is a single mission to fix the two biggest CRM gaps in restaurant tech: unreliable data, and systems that weren't built for this industry.

This playbook distills what we've learned. The tactics are proven. The insights are practical. The goal is clarity, not complexity.

Take what helps. Skip what doesn't. Use it to sharpen your CRM, align your GTM team, and book smarter.

Intro & expectation setting

Hello and welcome. I'm Grant Gadoci, and I help B2B teams build GTM strategies that grow restaurant-facing businesses faster and more predictably. If you're responsible for operationalizing the systems and teams that sell to restaurants, you're in the right place.

Why this exists

Eight years ago, I co-founded Restaurantology to help sales and marketing professionals become true restaurant industry experts. Our mission: give revenue teams better data and sharper systems so they can uncover opportunities, drive pipeline, and scale with purpose.

This playbook was built for B2B professionals managing or refining SaaS bookings systems, especially those targeting restaurants.

By the end, you'll have a four-part solution framework that defines success and aligns your GTM motion with the way bookings actually work.

Please Note: This isn't a demand gen guide. It's not about how to book more. It's about how to track, manage, and operationalize the bookings you already have.

Who this is for

The core audience includes anyone with admin-level Salesforce access, be it Sales Ops, RevOps, or Analysts able to implement changes confidently. But the reach goes broader. If you're in restaurant tech and care about standardized definitions, scalable architecture, or CRM alignment, this is for you.

Founders, executives, and GTM leaders will find value too, especially when reviewing, benchmarking, or refining an existing plan.

At its core, this is for anyone helping restaurant tech teams—whether quota-carriers, CSMs, or execs—learn how to talk about the industry, operate inside it, and serve it well.

What we're solving

The goal of this guide is simple: shorten your learning curve in building CRM bookings systems that actually support business outcomes.

We'll do that by focusing on two key objectives:

- 1. Define what SaaS bookings are, why they matter, and why they're uniquely tricky in restaurant tech.
- 2. Unpack the four most common bookings errors, then walk through real solutions that improve accuracy and scale.

Last thing before we dive in

If we haven't connected, feel free to reach out. I post regularly on LinkedIn and publish insights from over two decades spent building, fixing, and optimizing the revenue processes behind restaurant tech. You can also reach me at grant@restaurantology.io.

Let's get started.

The perks and problems with SaaS bookings

Accurate bookings are the foundation of revenue strategy in SaaS. For restaurant tech companies, getting bookings right isn't optional. It's essential for forecasting, prioritizing pipeline, and scaling with confidence.

The restaurant industry adds complexity: volatile operators, evolving tech stacks, and unpredictable expansion patterns. Layer on inconsistent CRM habits or manual processes, and even well-intentioned teams start flying blind.

Tracking bookings isn't just about numbers. It's about establishing a data-driven foundation for decisions across your business.

Before solving for bookings, your team needs shared definitions. That includes how bookings are calculated, what counts as committed revenue, and how to structure deals consistently across reps and segments.

This guide starts by breaking down what MRR bookings are and why they matter. Then we'll look at common friction points, especially in Salesforce, and how to build a system that balances short-term clarity with long-term scale.

What are MRR bookings, and why do they matter?

MRR bookings represent the total value of recurring contracts signed in a given period. For restaurant tech companies, this includes new logos, renewals, and upsells across products and services.

Tracking MRR bookings is essential because it:

- Forecasts future revenue: Bookings are your pipeline's most reliable indicator.
 Accurate data powers better financial planning and resource allocation.
- <u>Drives operational efficiency</u>: Teams can't prioritize effectively if they don't trust the data. Bookings insights help align sales coverage, onboarding, and support.

- <u>Clarifies LTV and CAC</u>: Without clean bookings data, lifetime value and customer acquisition cost models fall apart.
- <u>Builds stakeholder confidence</u>: Whether you're reporting to investors or aligning cross-functional teams, clean bookings reporting reflects GTM maturity.

Bookings data becomes your company's early-warning system. When done right, it tells you what's working, what's not, and where to focus.

Why bookings are tricky in restaurant tech

Even with clear definitions, bookings can be hard to manage, especially in restaurant tech. Two common issues make this harder than it needs to be: internal misalignment and resource gaps.

[01] Conflicting priorities across teams

Sales wants to front-load deals to hit quarterly goals. Finance wants clean revenue recognition. Leadership wants peaks to show growth. Everyone has a different reason to shape or report bookings in a certain way.

When those priorities clash, bookings get distorted. That creates short-term wins at the cost of long-term trust in the data.

[02] Gaps in RevOps capacity

Many restaurant tech companies operate with lean teams. Without dedicated RevOps support or proper Salesforce structure, teams fall back on shortcuts: manual entries, mismatched quantities, or unclear pricing logic.

That leads to unreliable reporting, broken dashboards, and missed targets. Worse, it scales poorly. The further you grow, the harder it becomes to fix.

Where most companies get MRR bookings wrong

Even with clean definitions and good intent, many restaurant tech companies still mismanage MRR bookings. The problem isn't always strategy. It's execution.

Inconsistent CRM usage, manual workarounds, and misaligned incentives create a fragile system. The result: inaccurate forecasting, reporting that can't be trusted, and GTM teams operating on broken data.

These four breakdowns come up most often:

[01] Missing products

Entering booking values manually—without using Opportunity
Products—breaks everything downstream. Without product-level structure,
revenue becomes hard to track, segment, or forecast. Line items aren't
optional. They're the foundation of bookings accuracy.

[02] Unclear quantities and pricing

One team uses quantities to count months. Another uses them to count locations. One AE enters ARR, another logs MRR. These inconsistencies corrupt financial reports and forecasting logic. Defining standard inputs at the org level keeps your data usable and your teams aligned.

[03] Misleading contracted revenue

Pilots, phased rollouts, and multi-brand deals often get booked in ways that favor commissions, optics, or cashflow at the cost of clarity. Front-loading everything might help this quarter's number, but it distorts what was actually signed and when. Establishing "booking rules" helps preserve both realism and repeatability.

[04] Post-hoc changes

Changing quantities, prices, or deleting churned units after a deal is marked Closed-Won erodes trust in the system. You lose historical accuracy, sabotage trend reporting, and create chaos for finance. Closed-Won means locked. If changes are needed, debook and document.

The cost of getting this wrong isn't just messy reports. It's strategic drift. If bookings can't be trusted, the business can't plan.

What a solution to MRR bookings might look like

Fixing MRR bookings starts with structure. The most effective path is a tailored CRM setup—especially at the Opportunity level—that captures what matters, enforces consistency, and scales as you grow.

That means building:

- Custom fields to track key metrics like MRR and contract terms
- Validation rules to prevent sloppy or inconsistent inputs
- Flows that automate updates and reduce manual errors

The goal is clarity without complexity. Start with the reporting you need, then work backward to define the fields and logic required to support it.

Building Opportunities for purpose-driven reporting

Every system should answer one question: can we trust the data? That starts with Opportunities built for reporting, not just for sales process.

By structuring Opportunities around clear, required inputs—like product-level line items, quantities, terms, and price types—you get data that drives smarter decisions across the org.

Validation rules and Flows make the process stick, triggering updates when changes happen and keeping your CRM consistent without relying on manual policing.

Key benefits:

- Improved data accuracy: Custom fields and validation rules ensure consistent, structured inputs across every deal.
- <u>Streamlined reporting</u>: Automated updates and reliable logic enable clean dashboards, easier audits, and better forecasting.
- <u>Scalability</u>: As your pipeline grows, your CRM can handle volume without losing structure or visibility.

 <u>Cross-team alignment</u>: When bookings live in one source of truth, sales, finance, and leadership stay on the same page.

Our path forward

This playbook breaks down the system into four parts. Each addresses a critical MRR challenge and offers practical solutions you can implement immediately:

- <u>Products, not promises</u> Line items (Opportunity Products) are non-negotiable. Manual revenue entries lead to bad forecasts and missing insights.
- Consistency in counts Define quantities and prices clearly. Standard inputs mean usable reports and less team confusion.
- <u>Booking boundaries</u> Handle pilots, partial rollouts, and complex deals with rules that preserve realism and reporting trust.
- <u>Set in stone</u> Closed-Won is sacred. Lock it. Use debooking, not edits, to protect historical accuracy.

Conclusion

There's no shortcut to clean bookings data. But there is a system.

By customizing your CRM with the right fields, automation, and guardrails, you can solve the most common breakdowns in MRR tracking. You'll get better forecasts, cleaner dashboards, and stronger alignment across GTM, finance, and leadership.

Don't chase perfection, chase trust. A bookings system you can rely on, scale with, and use to make smarter decisions.

The next four sections walk through the core components of that system. Each part covers one critical failure point, shows what it looks like in practice, and outlines how to fix it.

Let's start with the most common and most overlooked issue: products that never make it into the system.

Part 1 - Products, not promises

Accurate bookings start with accurate inputs. For restaurant tech companies, relying on manual entry—or skipping Opportunity Products altogether—undermines your ability to scale. Line items aren't just a Salesforce feature. They're the foundation of trustworthy, reportable, and auditable revenue data.

When every deal is rooted in structured product-level data, your team gains the visibility needed to forecast reliably, align GTM functions, and grow with confidence.

Defining the problem

Too many SaaS teams still treat the Opportunity > Amount field as a catch-all. They manually type in contract values and skip the product-level detail entirely.

This creates four common breakdowns:

- Inaccurate revenue tracking: Without linking revenue to products, you can't analyze performance or understand what's actually driving growth.
- 2. <u>Inconsistent reporting</u>: Manual inputs are error-prone. Even small mistakes compound across deals and quarters.
- 3. <u>Limited scalability</u>: As volume grows, the lack of structure slows reporting and kills trust in your dashboards.
- **4.** <u>Poor forecasting</u>: Without line items, forecasting lacks the granularity to detect patterns or adjust coverage effectively.

Opportunity Products solve all of this. When implemented correctly, they give you precision, scale, and confidence in your revenue data.

What about HubSpot? HubSpot supports product association too, but many teams still skip it. Whether you use Salesforce or HubSpot, the rule is the same: if you're serious about bookings, you need product-level data.

Aligning our vocabulary

Before you implement, align your team on the fields that matter. Standardized product data helps reporting scale and keeps internal conversations grounded in facts, not assumptions.

We recommend starting with:

- 1. <u>Product Type</u>: Core product vs. Add-on.
- 2. Revenue Type: Recurring vs. One-time.
- 3. Product Family: Logical groupings like platform, services, or business unit.

These fields should be enforced, not optional. They create the structure needed to answer questions across finance, GTM, and CS.

Implementing a solution

Fixing product-level gaps takes both system structure and team discipline. Here's how to start:

[01] Add custom Opportunity-level fields

Even with structured line items, reps and execs need summary data. Add custom fields to the Opportunity object to reflect what's been booked.

- MRR → Monthly Recurring Revenue
- NRR → Non-Recurring Revenue
- ARR \rightarrow MRR x 12
- ACV → ARR + NRR
- Weighted MRR → MRR x Opportunity Probability

Automate these fields using Flows. When products are added or changed, your system should update totals automatically.

[02] Enforce validation rules for Closed Won

Make it impossible to close a deal without structured product data. Suggested rules:

- A rollup to confirm that at least one product has a quantity greater than zero
- A validation to block closure if MRR equals zero (if relevant to your model)

These safeguards ensure that every Closed Won deal reflects real, auditable value.

Best practices

[01] Booking without line items

Not using Opportunity Products means sacrificing forecasting accuracy and product-level insights.

Example: A deal for 10 locations is logged with \$1,500 MRR but no breakdown of what was sold.

Bad practice: Booking total MRR in the Opportunity Amount without using Products

Good practice: Create Opportunity Products for each SKU with quantity, price, and revenue type defined

[02] Collapsing product bundles

Overly broad bundles make it impossible to understand what was actually sold—and what drives churn or expansion later.

Example: A legacy "Pro Plan" bundle includes three SKUs, but only one is still supported.

Bad practice: Booking the bundle as a single product with no detail **Good practice:** Break bundles into individual SKUs to support product-level churn and upsell analysis

[03] Missing revenue type classification

Without classifying revenue as Recurring or One-time, MRR/NRR calculations break down.

Example: A one-time implementation fee is booked as if it contributes to recurring revenue.

Bad practice: Logging all revenue without a clear type

Good practice: Use a Revenue Type field on Products and surface it on the

Opportunity Product to segment financial models

Industry Insight: In restaurant tech, **location count is your GTM foundation**. Yet too many sellers track revenue alone, missing the unit-level detail that powers expansion modeling, brand segmentation, and whitespace targeting. If you're not capturing Quantity at the line-item level, your CRM can't answer the most important question: "Where do we go next?"

Where do we go from here?

Structured product data is the backbone of reliable bookings management. When you pair it with automation and validation, you create a CRM system that's scalable, trustworthy, and actionable.

Before moving on, remember...

- <u>Use Opportunity Products</u>: Every deal. No exceptions.
- Automate field updates: Let Flow handle MRR, ARR, and ACV so reps don't have to.
- <u>Block incomplete deals</u>: Validation rules should stop any Closed Won that lacks product detail.

Next, we'll tackle the second most common source of bookings chaos: inconsistent use of Quantity and Price.

Part 2 - Consistency in counts

Adopting Opportunity Products is a strong start. But without consistency in how your team uses **Quantity** and **Sales Price**, the data falls apart fast. These inconsistencies might seem minor, but they erode trust in your CRM, distort reporting, and break forecasting logic.

Standardizing Quantity and Price inputs isn't just about data hygiene. It's about making bookings numbers actually mean something.

Defining the problem

Most inconsistencies fall into familiar patterns:

- Using Quantity to represent time (e.g., 12 = months in a year) instead of actual unit count
- Averaging Sales Price across locations rather than capturing true per-unit pricing

These shortcuts cause bigger problems:

- Skewed financial reporting: Misaligned inputs corrupt metrics like MRR, ARR, and ACV
- Inconsistent interpretation: Teams disagree on what the data means because it's not entered the same way
- <u>Misaligned incentives</u>: Reps inflate Quantity or fudge pricing to hit quota, not accuracy

Fixing this means codifying what each field should represent, and enforcing it.

Aligning our vocabulary

Lock in definitions. Then train around them.

- Quantity: Always represents how many units are being sold—restaurants, rooftops, or locations.
- 2. Sales Price: Always reflects MRR per unit. Period.

These rules should live in your Sales Handbook and be reinforced in onboarding, training, and deal desk reviews.

Implementing a solution

Consistency won't hold without systems to back it up. Here's where to start:

[01] Documentation and training

Spell out the rules. Include real examples:

- Single-location deals
- Price breaks across location tiers
- Add-ons priced separately

Update documentation regularly and use training to reinforce edge cases and new SKUs.

[02] Validation and audits

Use reports to flag discrepancies and hold the line on data quality:

- Audit whether Quantity x Sales Price = MRR
- Compare regional deal structures to catch inconsistent inputs
- Review rep-by-rep data patterns quarterly

Automation can surface the issues, but human oversight is still key.

Best practices

[01] Booking locations with different prices

Collapsing pricing into one line item makes it impossible to track variation across a deal.

Example: A five-location deal includes three locations at \$100/month and two at \$75/month.

Bad practice:

- Quantity = 1, Price = \$450 for five locations
- Averaging Price = \$90 with Quantity = 5

Good practice: Break it out into distinct lines:

- Quantity = 3 at \$100/month
- Quantity = 2 at \$75/month

This keeps both location count and per-unit pricing accurate.

[02] Booking add-ons

Without separating add-ons, you blur what was actually sold and lose expansion visibility.

Example: A location adds a paid module that increases MRR by \$50.

Bad practice:

- Roll the \$50 into the core product's Sales Price
- Leave the add-on undocumented

Good practice:

- Create separate Opportunity Products for add-ons with defined Quantity and Price.
- Skip internal-only features or configuration items that carry no financial weight.

[03] Booking seasonal locations

Treating seasonal restaurants as full-year subscribers skews revenue models and misleads planning.

Example: A location that operates 8 months per year is booked as if active year-round.

Bad practice:

- Booking full MRR resulting in misleading anticipated ARR
- Churning and rebooking each year

Good practice:

Book based on operational months:

• \$100/month for 8 months = \$67 MRR. This keeps ARR and forecast models realistic.

Industry Insight: Most revenue teams rely on CSV exports and analyst deep-dives to assess whitespace and upmarket traction. But with the right Opportunity structure, Quantity becomes the fastest signal of how much room is left to grow. A Quantity of 1 on a mid-market deal doesn't mean "small business" but rather signals a wedge. When logged to the right account, it becomes a launchpad for expansion, whitespace mapping, and smarter GTM segmentation. No spreadsheets required.

Where do we go from here?

Consistency in Quantity and Price is non-negotiable. If you don't trust the inputs, you can't trust the outputs.

Before moving on, remember...

- Quantities reflect locations: Use them to count restaurants, not time
- Sales Price captures MRR: Enter true monthly value per unit
- Break out pricing differences: Separate line items maintain accuracy and prevent confusion

Next, we'll explore how to create clear guardrails for complex deals like pilots, rollouts, and multi-brand accounts so bookings reflect business reality, not just quota math.

Part 3 - Booking boundaries

Complex deals are common in restaurant tech: pilots, phased rollouts, multi-brand contracts. Without clear rules for how and when to book them, companies inflate revenue, create confusion, and undermine trust.

You don't need more CRM features. You need guardrails. Bookings should reflect reality, not just quota math.

Defining the problem

Ambiguity around bookings creates mess fast. Common patterns include:

- <u>Delaying pilots</u>: Teams hold back Closed Won status until revenue starts, even if a signed pilot is live
- Booking too early: Full rollouts are booked upfront before pilots conclude or revenue is certain
- Combining brands: Unrelated multi-brand deals are lumped into one Opportunity, killing reporting clarity

These habits lead to:

- Overstated revenue: MRR gets inflated by deals that aren't live or guaranteed
- Lost trust: Finance and execs lose confidence in what's actually real
- <u>Hidden risk</u>: Pilots fail quietly and phased rollouts drag, but none of that shows up in your CRM

The solution: enforce four clear "gates of truth" for every Closed Won booking.

Aligning our vocabulary

These gates define what qualifies as a valid Closed Won. Use them to audit deals, coach reps, and structure your CRM.

[GATE 01] Contract-to-Opportunity mapping

Each contract should tie to one Opportunity. Avoid combining separate contracts, especially if they differ in scope, terms, or timing.

Exception: Add-on growth (e.g., NROs) can be rolled into an existing Opportunity if properly flagged (e.g., Type = Organic Growth).

[GATE 02] Imminent revenue only

Bookings should reflect revenue that starts soon, ideally within 60 days. If implementation is phased, note that clearly. Don't book aspirational revenue.

[GATE 03] Structure by brand hierarchy

Multi-brand deals must follow your Salesforce account hierarchy:

- Split brands that don't share a parent (e.g., Subway and McDonald's)
- Combine brands under a single corporate owner (e.g., Taco Bell and Pizza Hut under Yum! Brands)

This keeps reporting clean and scalable.

[GATE 04] Revenue motion defines Opportunity Type

Use Opportunity > Type to reflect the work behind the win:

- Acquisition: New business, pilots, rollouts, or new franchisees
- Expansion: Upsells, cross-sells, or NROs within existing relationships

Labeling matters. Misclassifying a full-cycle franchisee deal as "Expansion" distorts both effort and forecast accuracy.

Implementing a solution

Gates are only useful if enforced. Here's how to make them stick:

[01] Documentation and training

Include pilot, rollout, and multi-brand rules in your Sales Handbook. Clarify how to tag deals and distinguish motions. Use real examples to train AEs and RevOps.

[02] Validation and audits

Enforce with system logic and scheduled reviews:

- Require attached contracts for Closed Won (exceptions allowed for specific motions)
- Flag pilot deals and monitor conversion to paid
- Use negative quantities to debook pilot units as they convert, preserving count integrity

Best practices

[01] Booking pilots and rollouts

Companies often blur the line between pilot phases and full-scale rollouts, inflating bookings and distorting pipeline metrics.

Example: A brand agrees to pilot five locations, with potential to expand to 50 if successful.

Bad practice:

- Booking all 50 locations up front under one Opportunity
- Booking nothing because the trial is free

Good practice:

- Create a Pilot Opportunity for 5 locations (Type = Pilot), being sure to document price
- Float an open Opportunity for the remainder (Type = Rollout) with the remaining locations and prices

[02] Managing trial-to-paid conversions

Unstructured transitions from trial to paid accounts can break forecasting and misrepresent customer intent.

Example: A five-location trial converts into a 20-location paid contract two months later.

Bad practice:

• Let the original trial Opportunity linger or overwrite it with paid deal info

Good practice:

- Close the trial as-is
- Open a new Opportunity for the 20 paid locations
- Use quantity math to reflect net-new growth: -5 (trial) +20 (paid)

[03] Handling phased rollouts

Many teams overestimate revenue by booking phased implementations all at once.

Example: A 100-location brand signs a rollout contract but implementation is staggered over 6 months.

Bad practice:

Booking all 100 locations immediately

Good practice:

- Create Opportunities tied to each rollout phase
- Add a custom field for expected go-live dates
- Book revenue as contracts become actionable and time-bound

[04] Split-booking multi-brand deals

When a single parent company owns multiple brands, bookings can get messy fast.

Example: A deal spans two brands under different parent entities.

Bad practice:

Logging both brands under a single Opportunity tied to one Account

Good practice:

- Separate Opportunities for each brand if no shared parent exists
- Use a unified Opportunity only if brands roll up to the same corporate (e.g., Yum! Brands)

[05] Defining revenue motions

Labeling Opportunities incorrectly distorts reporting and go-to-market metrics.

Example: A franchisee buys for the first time, but their parent brand already uses your product.

Bad practice:

Marking the Opportunity as Expansion

Good practice:

- Use Opportunity > Type to reflect the motion: New Business for first-time franchisee
- Coach reps to split distinct motions (e.g., renewal + upsell) into separate
 Opportunities

Industry Insight: In multi-brand restaurant groups, bookings tied to a single parent company often span distinct concepts, teams, and timelines. Without booking hygiene—like splitting unrelated brands into separate Opportunities—you risk muddled reporting and delayed rollouts. A deal with Inspire Brands isn't one brand. It's six.

Where do we go from here?

Establishing booking boundaries brings structure to complexity. The four gates of truth give your team a shared framework for handling pilots, phased rollouts, and multi-brand contracts—all while preserving reporting accuracy.

Before moving on, remember...

- Gate 1: One contract, one Opportunity
- Gate 2: Only book revenue that's near-term and confirmed
- Gate 3: Structure deals based on brand hierarchy
- Gate 4: Let the sales motion—not the account—drive Opportunity Type

In the final section, we'll cover how to lock down your CRM so bookings can't be changed after they're closed, preserving historical accuracy and avoiding silent revisions.

Part 4 - Set in stone

Once a booking is marked Closed Won, it becomes a historical artifact. It should not be edited, reshaped, or undone. Bookings represent a record of commitments made at a specific point in time. Altering them after the fact breaks the chain of trust that reporting, forecasting, and planning rely on.

The integrity of your bookings system lives or dies by how you treat Closed Won deals.

Defining the problem

Post-hoc changes are the most damaging form of data corruption in CRM. They happen in several ways:

- <u>Reverting wins to losses</u>: If a deal fails in implementation, reps undo the Closed Won stage
- Editing line items: Quantities or products are removed after churn or change
- Altering pricing: Reps adjust original pricing instead of recording it as a renewal update
- Inventing post-Won stages: Custom statuses like "Implementation" or "Churned" introduce chaos

These edits lead to:

- Lost historical accuracy: Past reports become unreliable and trends unusable
- Broken performance metrics: Win rates, ACV, and churn are distorted
- <u>Disconnected systems</u>: Salesforce automations fail when Closed Won logic breaks
- Misaligned teams: Finance, RevOps, and leadership all report different truths

Aligning our vocabulary

To protect your historical record, three core principles apply.

[01] Closed Won is final

Any changes—good or bad—must be captured in a new Opportunity. This keeps the original booking clean and auditable.

Example:

- A customer drops locations post-sale? Don't edit the deal. Update the Subscription.
- A price is renegotiated at renewal? Capture it in the renewal record, not the original win.

[02] Track contract length with Initial Term

Add a required field for Initial Term (in months). It defines the window for valid contract adjustments and helps teams distinguish legitimate changes from net-new business.

[03] Separate contract adjustments from debookings

Borrowing from restaurant logic:

- Adjustments are comps: pricing changes, quantity edits within the contract
- Debookings are voids: full reversals when deals collapse or were booked in error

This framing keeps reporting clean and language consistent across teams.

Implementing a solution

Implementing controls ensures bookings data remains reliable and actionable.

[01] Documentation and training

Update your Sales Handbook with clear policies:

- When to debook vs. adjust
- How to handle renewals, churn, and pricing shifts

How the Initial Term field guides change decisions

Train teams quarterly and during onboarding refreshers.

[02] Validation and audits

Use system logic to enforce guardrails:

- Lock Closed Won fields: Prevent edits to Products, Pricing, and Stages
- Allow exceptions with controls: Give same-day edit access or manager overrides if needed
- Audit edits regularly: Track all field changes using Salesforce reports or audit trail tools

Consider a centralized Deal Desk to validate and finalize complex bookings before closure.

Best practices

[01] Debooking vs. adjustments

It's common to overuse debookings for small pricing or scope changes, but this muddles your data trail and inflates churn.

Example: A customer downgrades a \$500/month package to \$450/month mid-term.

Bad practice:

- Debooking the full deal and rebooking it at the new amount
- Making changes directly in the original Opportunity without context

Good practice:

- Set thresholds (e.g., use debooking only if change exceeds 10% MRR)
- Track smaller shifts via contract adjustments
- Always require clear documentation and approvals for edits
- Use Initial Term to define eligibility and limits for in-term changes

[02] Managing renewals and pricing shifts

Renewals often blur with expansions or pricing updates—lumping them together kills data clarity.

Example: A customer renews and adds new features with a pricing uplift.

Bad practice:

- Reopening or editing the original Opportunity
- Failing to track upsell revenue separately from the renewal

Good practice:

- Use a dedicated Opportunity with Type = Renewal
- Automate creation via Flow so the renewal enters pipeline with accurate terms
- If upsell is substantial, create a second Opportunity with Type = Upsell
- If combined, document both motions clearly in the Opportunity Products

[03] Retiring product bundles

Transitioning away from bundled SKUs can get messy if not handled methodically.

Example: A legacy bundle is replaced by discrete modular SKUs after product restructuring.

Bad practice:

- Removing the bundle without rebooking its components
- Skipping clear attribution of churn vs. expansion

Good practice:

- Use negative quantities to zero out the legacy bundle
- Rebook each new SKU separately with appropriate quantities and pricing
- This preserves clarity in churn, expansion, and renewal reporting

Industry Insight: Implementation failure in restaurant tech often stems from **brand-level misalignment, not product failure**. When bookings are reversed post-close to "correct" this, you lose visibility into that failure mode. Debooking shouldn't hide failure or erase history; it should highlight risk exposure in your sales process.

Where do we go from here?

A Closed Won booking is your system's version of "signed and sealed." It should never be rewritten. Locking down historical records safeguards reporting, protects system integrity, and prevents cross-team confusion.

For detailed definitions of Opportunity Types, Lead Sources, and recommended custom fields, refer to the <u>Appendix</u>. These resources provide the building blocks for implementing the best practices outlined here and ensure alignment across your CRM and GTM teams.

Before moving on, remember...

- <u>Closed Won is locked</u>: No edits. Use new Opportunities to reflect changes
- Debookings ≠ adjustments: Track each distinctly with clear logic
- Renewals and pricing shifts deserve structure: Automate where you can, and enforce consistent Opportunity usage

Next, we'll wrap up this guide with final thoughts on scaling your bookings system, from accuracy to alignment, from one deal to thousands.

Final thoughts

Let's bring it full circle.

You now have a proven framework for managing SaaS bookings with clarity and control designed specifically for the complexity of restaurant tech. From clean inputs to preserved historical records, each step is about removing ambiguity and building trust in your data.

Here's what we covered:

- Require line items: Use Opportunity Products to capture product-level detail and power accurate reporting.
- Align Quantity and Price: Standardize inputs so your MRR, ARR, and ACV actually reflect what was sold.
- 3. <u>Set booking boundaries</u>: Handle pilots, rollouts, and multi-brand deals with structure and discipline.
- Lock in history: Treat Closed Won like a ledger—unalterable, auditable, and trusted.

This guide isn't about redefining bookings. It's about operationalizing them so they scale as you do.

The power of accurate bookings

Your bookings system is more than a financial tool, it's a source of truth across your GTM engine.

In restaurant tech, where complexity is built into every deal, structured bookings do more than forecast revenue. They unify how sales, finance, and leadership measure traction, allocate resources, and evaluate go-to-market performance.

Accurate bookings unlock:

- Faster expansion planning
- Clean renewal baselines
- Better segment and whitespace analysis
- Trust across every downstream system

When this data breaks, everything downstream breaks with it. But when it's built right, it becomes the lever you use to scale without friction.

A living framework

This isn't the final form. It's your foundation.

Revisit it often. As products evolve, teams grow, or strategy shifts, your bookings system should keep pace. What matters is that you've committed to structure, alignment, and accountability.

Thanks for letting me walk you through it. You now have the tools to lead with precision, scale with purpose, and turn your bookings data into a strategic advantage.

Here's to what comes next.

Appendix

Opportunity Custom Fields: Recommendations

Туре	Field Label	Field Type	Values / Examples	
Custom	MRR	Currency (2,0)		
Custom	Weighted MRR	Formula (Currency)	MRR_c * Probability	
Custom	NRR	Currency (2,0)		
Custom	ARR	Formula (Currency)	MRR_c * 12	
Custom	ACV	Formula (Currency)	ARR_c + NRR_c	
Custom	Initial Term (in months)	Number (3,1)		
Custom	MRR Won	Formula (Currency)	IF(IsWon = True, MRR_c, 0)	
Custom	Days to Close	Formula IF(IsClosed,(CloseDate -DATEVALUE(CreatedDate)), null)		

Opportunity > Type: Definitions

Туре	Revenue Motion	Definition	
New Business	Acquisition	Net-new logos or new location deals. Any expansion not tied to an existing contract counts as New Business, even if it's a subset of locations within a current customer.	
Pilot	Acquisition	A contractually confirmed pilot for a subset of locations. Should always precede a Rollout and include clearly defined conversion criteria.	
Rollout	Acquisition	Revenue expected post-pilot. Rollouts should only exist alongside a corresponding Pilot, and remain at 80% probability until the pilot completes.	
Upsell	Expansion	New modules or add-ons at existing, active locations. These expand account value but do not add new locations.	
Cross-sell	Expansion	A sale of an additional, distinct product to an existing customer. Captures horizontal expansion across your platform.	
Organic Growth (NROs)	Expansion	New locations opened by an existing, fully adopted customer. Follows CAP (critical adoption point).	
Renewal	Retention	Continuation of an existing contract. May include pricing upticks or modified terms.	
Contract Adjustment	Financial	Mid-contract changes to price, scope, or duration. Similar to a restaurant "comp."	
Debooking	Financial	A full reversal of a booking (e.g., failed implementations or booking errors). Similar to a "void."	

Opportunity > Lead Source: Recommendations

Lead Source	Category	Definition	
MKT - Advertisement	Marketing	Paid advertising campaigns including digital, print, radio, or TV.	
MKT - Conference	Marketing	Leads generated via sponsored presence or speaking at events.	
MKT - Content	Marketing	Inbound leads from webinars, blog posts, whitepapers, videos, etc.	
MKT - Paid Search	Marketing	SEM efforts like Google Ads or Bing Ads.	
MKT - Purchased List	Marketing	Outreach based on third-party lists.	
MKT - Website	Marketing	Direct website traffic including forms, chat, and contact clicks.	
BD - Channel Partners	Partnerships	Partner-led sales or referrals from reseller agreements.	
BD - Consultants	Partnerships	Consultant-referred opportunities based on client relationships.	
SLS - Cold Prospecting	Sales	Net-new outbound outreach including cold calls and cold emails.	
SLS - Conference	Sales	Leads generated through field sales attending events.	
SLS - Exec Network	Sales	Opportunities from executive or board networks.	
SLS - Referral	Sales	Warm handoffs from partners, customers, or internal networks.	
SLS - External Demand Gen	Sales	Leads generated through third-party SDR firms or outsourced efforts.	
SLS - Internal Demand Gen	Sales	Leads generated by internal SDRs working inbound or outbound.	
CX - Existing Customer (Sales-Initiated)	СХ	Sales-driven expansion within current accounts.	
CX - Existing Customer (CX-Initiated)	сх	Opportunities identified by CSMs during account management.	
CX - Existing Customer (Customer-Initiated)	СХ	Customers proactively requesting expansion or add-ons.	
Other	Other	Use sparingly. Review regularly to determine if a new category is needed.	

Note: Include a Lead Source Detail field (free text) and/or use Primary Campaign Source to capture specific context. This supports more granular reporting and attribution.

Automation Template: Calculating MRR and NRR via Opportunity Products

Flow Name: Opportunity Product: Calculate MRR and NRR Totals Flow Type: Record-triggered Flow

- ► Start (Trigger Configuration)
 - **Object**: Opportunity Product
 - Trigger Flow When: A record is created or updated
 - Condition Requirements: None
 - When to Run: Only when a record is updated to meet the condition requirements
 - Optimize the Flow for: Actions and Related Records
 - Include a Run Asynchronously Path: No

Attention: This Flow depends on a custom Revenue Type picklist field on the **Product2** object, with values for **Recurring** and **One-Time**.

To reference this classification in the Flow, create a formula text field on the Opportunity Product called Revenue_Type__c that stores the selected value locally.

Resources Needed

Туре	API Name	Data Type	Object	Value
Constant	mrrRevType	Text		Recurring
Formula	tempCompare	Boolean		{!mrrRevType} =={!tempRevT ype}
Variable	oppProductList	Record (Allow multiple values = TRUE)	Opportunity Product	
Variable	oppProductCo unt	Number		0 decimal, 0 default value
Variable	MRRTotalPrice	Currency		2 decimal, 0 default value
Variable	NRRTotalPrice	Currency		2 decimal, 0 default value
Variable	tempRevType	Text		null

1 Step 1 - Get Records

• Label: getOpportunityProducts

• **Object**: Opportunity Product

• Filter: OpportunityId Equals \$Record > Opportunity ID

• Store: All records

• Fields to store: ID, TotalPrice, Revenue_Type__c

• Output Variable: oppProductList

2 Step 2 - Loop: iterateThroughProducts

Collection: oppProductListDirection: First item to last

Inside Loop:

Assignment 1: oppProductCount += 1

- Assignment 2: tempRevType = loopProductLineItems > Revenue Type
- Decision:
 - If tempCompare == TRUE:
 - → MRRTotalPrice += loopProductLineItems > TotalPrice
 - o Else:
 - → NRRTotalPrice += loopProductLineItems > TotalPrice

3 Step 3 - Decision: checkProductCount

• **Condition:** oppProductCount > 0

4 Step 4 - Update Records

Label: updateMRRAndNRR

• Object: Opportunity

Filter: Id Equals \$Record > Opportunity ID

Set Fields:

MRR_c = MRRTotalPrice

NRR_c = NRRTotalPrice

Note: Clone this Flow and save as Opportunity Product: Adjust MRR and NRR Totals on Deletion to recalculate totals when a product is removed.

In the cloned Flow:

- 1. Set the trigger to run when a record is deleted
- 2. Update the Get Records filter to **exclude** the deleted line item:

Id Does Not Equal \$Record > Line Item ID

Additional resources

Looking to go deeper? Explore more from Restaurantology:

- SaaS Churn Guide: Sales Ops tactics and CRM strategies to help track and manage restaurant tech attrition at scale → <u>Download here</u>
- KNOW THE MARKET: A 5-step go-to-market guide for companies selling to restaurants → <u>Watch on YouTube</u>
- MarketMindsTM Blog \rightarrow Read here
- MarketMinds[™] Podcast → <u>Listen here</u>

Note from the author

Thank you for reading Restaurantology's SaaS Bookings Guide.

This playbook is the product of years spent deep in the details helping restaurant tech companies align their CRM, sharpen their GTM, and build bookings systems that actually scale.

If this helped you or your team, I hope you'll share it. The more we standardize how bookings are tracked, the faster this ecosystem matures.

And if you see an opportunity to improve it—or want to talk shop about market intelligence, data-driven GTM strategy, or RevOps infrastructure—I'd love to hear from you.

All the best,



Grant Gadoci

CEO @ Restaurantology | GTM, RevOps, and Revenue
R&D for B2B restaurant tech companies
Email | LinkedIn | Schedule time to chat