SCOTCH GAME

Offer & Pricing Architecture
A multi-unit playbook to win share and profit

Price doesn't win alone. In multi-unit businesses, you gain **market share** and **profitability** when the **offer system**—what you sell, how you package it, how you price and present it, and how/when/where you make it available—matches the jobs customers hire you for and the moments they're ready to buy. Pricing then **amplifies** that fit.

Your portfolio is the customer's map of value

Across hundreds or thousands of locations and digital touchpoints, your portfolio should be designed around **Occasions** which are the **context × time window × channel/fulfillment** where the purchase happens (on-the-go vs. planned; self-serve app vs. assisted; off-peak vs. peak).

If the portfolio doesn't reflect these, price is just a tax. When it does, price is an accelerant.

Ten cross-unit principles

1) Define the Offer System, not just items

Think beyond "products." An offer is a **package**: core + options/add-ons, access level (standard/premium), service level (SLA/turnaround), commitment term, convenience features (delivery, pickup, concierge), and payment model (one-time, subscription, usage, financing).

2) Build a Tiered Value Architecture (Good/Better/Best)

Give customers an **earned** ladder—clear trade-offs in benefits, convenience, and support—so value-seekers and premium buyers both see themselves. Tie each tier to **COGS/GP targets** and clear operating requirements.

3) Version by context (moment × channel × fulfillment)

Don't ship one version to all contexts. Configure variants for on-the-go vs. planned use, staffed vs. self-serve, peak vs. off-peak. This grows **reach** (new moments) and **frequency** (more relevance).

4) Use Bundles & Add-ons to move mix

Engineer attach paths (“starter + must-have add-on,” “family + convenience add-on”) that improve **GM$ per transaction** and simplify choices. Bundles should trade up, not just discount.

5) Install a Price-Pack Architecture (PPA)

Decide what the “pack” is in your world (units, time, features, users). Then set **price corridors** by market/segment and **fences** (time, channel, audience) so you can flex value locally without eroding the brand.

6) Govern Promotions to create reach, not addiction

Promos are for **trial and CEP coverage**, not replacing everyday value. Use fenced offers and keep **always-on value lines** to protect price image.

7) Make Value Communication do the heavy lifting

Name, layout, and sequence matter (in-store signage, app UI, web pages, sales scripts). Present the ladder and bundles so the better choice feels obvious. This is where you “earn” the price.

8) Protect Perception Guardrails

Track price image, fairness, and NPS alongside economics. Win share **without** crossing thresholds that trigger churn or backlash.

9) Simplify for Throughput & Reliability

Portfolio discipline (SKU/service catalog control, retire-to-replace policy) reduces training load, errors, queues, and cost—freeing capacity to serve more customers at higher quality.

10) Pull Offer & Pricing upstream into development

Stage-gate every new concept against the architecture: Which **job** is it for? Which **entry moment** does it unlock? Which **occasion** does it win? Does it **fit the ladder** and corridor economics? Can ops deliver reliably at scale?

Measurement that travels across units

- **Economics:** traffic, GM$ / transaction, contribution margin, mix/attach, utilization of tiers/versions, promo dependency.
- **Demand:** traffic vs. ticket (or volume vs. ARPU), repeat rate, cohort retention.
- **Channels/occasions:** adoption by channel, off-peak fill, context penetration.
- **Perception:** price image, fairness, NPS, complaints.
- **Discipline:** corridor compliance, promo fence adherence, retire-to-replace ratio.

90-day operating plan (portable to any multi-unit)

Weeks 1–6 — Diagnose & design

- Map JTBD, CEPs, and key occasions; quantify where you **don't** show up.
- Analyze elasticity, mix/attach, corridor compliance, off-peak performance, promo dependency.
- Draft the tiered value architecture, context versions, and first wave of bundles/add-ons.

Weeks 6–8 — Pilot

- Launch **3–5 pilots**: one versioning change, one bundle/attach program, one corridor/fence reset, one value-comms revamp.
- Set guardrails: margin, perception, NPS, and operational reliability.

Weeks 9–12 — Scale

- Read the scorecards (incrementality over holdouts). Scale what proves out. Retire what doesn't. Publish the **Offer & Pricing Playbook** and cadences (weekly ops, monthly portfolio, quarterly strategy).

Common failure modes (and fixes)

- **One-size pricing:** ignores local demand curves → **corridors + guardrails**.
- **Promo dependency:** trains discount behavior → **fences + always-on value**.
- **Complexity creep:** too many choices → **retire-to-replace** and enforce catalog rules.
- **Premium without proof:** price ladders without visible value → **value comms** and service level clarity.
- **Isolated innovation:** new items that don't fit the ladder → **stage-gate against the architecture**.

Bottom line: Multi-unit winners treat the portfolio as a **system**—offers, versions, bundles, price packs, and communication—built around jobs, entry moments, and occasions. Pricing amplifies a portfolio that fits; it cannot repair one that doesn't.