

## Business Strategy / Operational Systems

Develop systems for onboarding, evaluating, and offboarding vendors, contractors, and strategic partners.

Difficulty: Intermediate → Advanced

Model: GPT-4 / Claude / Gemini

Use Case: Vendor Management, Partner Relationships, Contractor Oversight

Updated: May 2026

Why This Prompt Exists

Most businesses have ad hoc vendor relationships — no onboarding, no evaluation, no offboarding.

You get:

- vendors who don't meet expectations (no accountability)
- no way to compare vendors (no data)
- security and compliance risks
- difficult transitions when changing vendors
- vendor lock-in because offboarding is messy

But vendor management is not bureaucracy.

It is risk management and quality assurance.

- Onboarding: vetting, contracts, access setup
- Evaluation: performance metrics, regular reviews
- Communication: points of contact, escalation paths
- Offboarding: access removal, data transfer, final payment

Without a system, vendor relationships are chaotic.

This framework forces AI to build a vendor management system.

The Prompt

Assume the role of a vendor management specialist who creates oversight systems.

Your task is to create a vendor management system.

Generate:

1. VENDOR ONBOARDING CHECKLIST
  - Vetting criteria
  - Contract requirements
  - NDAs and security reviews
  - Access setup
  - Point of contact assignment
  
2. PERFORMANCE METRICS (per vendor type)
  - Quality metrics
  - Timeliness metrics
  - Communication metrics
  - Cost metrics
  
3. EVALUATION CADENCE
  - Quarterly business reviews
  - Scorecard process
  - Issue escalation path

#### 4. COMMUNICATION PROTOCOL

- Primary contacts
- Meeting cadence
- Status reporting requirements

#### 5. OFFBOARDING CHECKLIST

- Access removal
- Data transfer
- Final payment
- Exit interview
- Contract termination

#### 6. VENDOR PORTFOLIO REVIEW

- Annual vendor rationalization
- Consolidation opportunities

#### INPUTS:

Number of Vendors/Contractors:

[INSERT NUMBER]

Vendor Types:

[E.G., "Software vendors, freelancers, agencies, suppliers"]

Current Pain Points:

[LIST OR "UNKNOWN"]

Compliance Requirements:

[LIST OR "NONE"]

Critical Vendors (can't operate without them):

[LIST]

RULES:

- Onboarding: vet before they start (prevention)
- Metrics: what gets measured gets managed
- Evaluation: quarterly for critical vendors
- Offboarding: have a plan before you need it
- Documentation: keep vendor files organized
- Regular vendor review: consolidate where possible

How To Use It

- Vet vendors before onboarding (prevention is cheaper than fixing).
- Quarterly reviews for critical vendors (stay ahead of issues).
- Document everything (protects both parties).
- Offboarding checklist prevents access and data leaks.
- Annual vendor review: consolidate where possible.

Example Input

**Number of Vendors/Contractors:** 15 (active)

**Vendor Types:** Software vendors (5), freelance designers (4), content writers (3), virtual assistants (2), agency partner (1)

**Current Pain Points:** "No standard onboarding process," "Don't track performance," "Hard to compare freelancers," "Offboarding is chaotic"

**Compliance Requirements:** Data privacy (GDPR, CCPA), IP ownership

**Critical Vendors:** Software vendors (can't operate without them), agency partner

## Why It Works

Most vendor relationships are ad hoc.

This framework improves outcomes by forcing:

- onboarding standards (quality control)
- performance metrics (accountability)
- evaluation cadence (oversight)
- communication protocols (clarity)
- offboarding checklists (risk management)

Great vendor management doesn't just sign contracts — it manages relationships systematically.

## **Build Better AI Systems**

Subscribe for advanced prompt engineering, AI business strategy tools, operational systems frameworks, and practical strategies for leaders and operators.

Carefully engineered prompts for people doing real work.

### **Share this:**

- [Share on Facebook \(Opens in new window\) Facebook](#)
- [Share on X \(Opens in new window\) X](#)

See also [The Meeting Rhythm System Designer](#)