

HOW TO BUILD AN SPM RFP THAT REVEALS THE RIGHT FIT

A practical guide to turning business priorities
into comparable, decision-ready vendor responses



What an Effective RFP Is Designed to Do

The goal of a Request for Proposal (RFP) is to identify the solution that can support your highest-value outcomes with acceptable effort, cost, complexity, and risk. It should give vendors enough context to respond to your reality, require evidence behind each claim, and create a fair basis for comparing fit, effort, risk, and total cost.

Sales Performance Management has changed a lot over the years. Today, most established platforms can satisfy a broad checklist on paper. The meaningful differences are in how a capability is delivered, what must be configured vs. customized, how the solution fits into the wider technology environment, and how confidently your team can operate it after launch.

A deadline may explain why the evaluation is happening now but it should not define your future requirements or cause you to recreate yesterday's operating model on a new platform.

Before Beginning: The RFP Is Not the Evaluation

The RFP is one instrument in a larger tech-evaluation process. Before drafting, align on four things:

1. **The problem:** why you are evaluating, what must change, and what is in or out of scope
2. **Future state:** what needs to change for stakeholders (administrators, sellers, leadership, etc.)
3. **Current state:** map the plans, processes, data, integrations, etc. the solution must support
4. **Priorities:** which requirements are mandatory now vs. later and how they should be weighted

*NOTE: You **don't** need to produce a full current-state assessment for the RFP. However, you do need enough shared understanding so it doesn't become a collection of disjointed stakeholder requests.*

1. Define Requirements That Can Drive a Decision

Start with the users and outcomes you want, not with a feature list. Ask where time, trust, control, visibility, or adaptability needs to be improved.

Stakeholder	Questions to Ask
Compensation administrators	What takes too long? Where is technical support or manual reconciliation required? Etc.
Sellers and managers	Where do users lack timely visibility or trust? What creates disputes? What should they be able to understand without admin support?
Finance and leadership	Which cost, forecast, approval, or scenario insights are difficult to produce? How quickly can proposed plan changes be modelled?
IT and security	Which integrations, controls, architecture constraints, recovery standards, and maintenance burdens must be addressed?
The organization	What future scale, regions, acquisitions, sales models, or operating changes must the environment support?

Separate Four Types of Requirements

1. **Mandatory:** use sparingly or risk eliminating viable solutions before trade-offs are understood
2. **Weighted:** areas where different degrees of fit, effort, and risk should affect the decision
3. **Future consideration:** what's likely to matter later, but not enough to outweigh present-day fit
4. **Informational:** useful for diligence/planning but not directly scored

Use your current system as evidence, not as a blueprint. Existing reports, workflows, etc. may have been created due to past limitations and should probably not survive the transition.

2. Build the RFP Around the Decision

The RFP should help vendors understand your environment, respond consistently, and show where their proposed approach creates advantage or introduces compromise.

1. Organization and project context

Explain why the evaluation is happening, the desired outcomes, scope, who is participating, plan complexity, geographic footprint, current systems, known pain points, and major dependencies.

2. Response instructions

Define the timeline, clarification process, required format, page limits, templates, evidence expectations, assumptions, exceptions, and demonstration requirements.

3. Business and functional requirements

Organize around real workflows: plan modelling, participant management, transactions, crediting, calculations, adjustments, approvals, statements, disputes, reporting, forecasting, and support.

4. Technical, data, and security requirements

Include: architecture, APIs, identity, data movement, integrations, performance, environments, recovery, privacy, accessibility, auditability, and ongoing ownership.

5. Implementation and operating model

Ask for methodology, expected client-side resources, testing, change management, hypercare, post-go-live support, and the division of responsibilities between vendor, partner and client.

6. Product direction and commercials

Separate generally available capability from roadmap promises. Standardize pricing for software, implementation, integrations, support, growth, renewals, and exit.

7. Relevant proof

Don't get distracted by vendor logos. Request customer examples and references that resemble your scale, complexity, geography, technology environment, industry, and operating model.

Do not let future roadmap functionality receive the same weight as demonstrated and generally available capability.

3. Ask Questions That Produce Evidence

Generic questions produce generic claims. Strong questions describe the business scenario and require the vendor to disclose effort, dependencies, limitations, and proof.

Pro-Tip: Use a 5-Part Question Pattern

- **Context:** describe the relevant environment, role, scale, process, or constraint.
- **Scenario:** state the task, exception, or outcome the solution must support.
- **Method:** ask how the vendor would handle it in the proposed solution.
- **Classification:** require the vendor to identify how the capability is delivered.
- **Evidence:** request a demonstration, workflow, documentation, customer example, etc. as proof.

Weak example	Decision-ready example
Do you support retroactive changes?	Describe how an administrator would process a retroactive plan change affecting closed periods, x currencies, and approved statements. Identify recalculation, approval, communication, and reconciliation steps.
Does the platform integrate with our CRM?	Explain how opportunity, account, participant, and territory data would move between the proposed platform and our CRM. Identify what is native, configured, customized, partner-built, or separately licensed.
How easy is the system to use?	Show how an administrator would create, test, approve, and publish a mid-year rate change. State the technical skills and outside support required.
Do you use AI?	Identify each proposed AI-supported capability, the user problem it addresses, the data it accesses, the human controls that apply, and whether it is generally available in the proposed edition today vs. future roadmap.

4. Make “Yes” Harder to Hide Behind

Require a response classification before the explanation for every material requirement:

- **Native or standard:** available in the proposed product and edition without material development
- **Configuration:** achieved through supported setup and maintained through standard upgrades
- **Customization:** requires custom development, scripts, or non-standard extensions
- **Partner or third party:** depends on another provider, product, connector, or contract
- **Roadmap or planned:** not generally available today
- **Not supported:** cannot be met as proposed or requires a materially different approach

Ask for evidence, dependencies, client responsibilities, additional costs, estimated effort, known limitations, and proposed alternative. This exposes the work and risk behind the answer.

Score Fit, Evidence, and Risk

Approve the scoring model before responses arrive.

Score	Meaning
0	Not supported
1	Major gap or unproven approach
2	Partial fit with significant compromise, customization, cost, or complexity
3	Meets the requirement with acceptable effort and evidence
4	Strong fit with low complexity and convincing proof
5	Distinct strategic or operational advantage that matters to the stated outcome

For the most material requirements, record three things separately:

1. **Fit:** how well the approach supports the requirement and business outcome
2. **Confidence:** whether the claim was demonstrated, documented, referenced, contractually committed, or merely asserted
3. **Risk and effort:** customization, implementation complexity, client workload, third-party dependency, cost, upgrade impact, and operational skill required

A solution may score well on functional fit and still introduce unacceptable risk.

5. Normalize Commercials and Pressure-Test the Draft

Vendors package products, services, and assumptions differently. A lower-looking proposal may simply omit work or shift it to the client. Use one template and require every vendor to price against the same volumes, modules, integrations, support level, contract term, growth assumptions, etc.

Separate the Full Cost

- **Software:** subscriptions, modules, user or participant tiers, environments, storage, overages, etc.
- **Implementation:** discovery, design, configuration, migration, integrations, reporting, testing, etc.
- **Operations:** support, managed services, release assistance, modernization, training, etc.
- **Growth and exit:** renewal increases, added populations or regions, professional-service rates, data extraction, transition support, and post-contract access

Final Release Sanity Check

- Does every scored question connect to an agreed priority?
- Will strong and weak approaches produce meaningfully different answers?
- Is every mandatory item genuinely non-negotiable?
- Have current-system terminology and incumbent architecture been challenged?
- Have vendors been given enough context to estimate accurately?
- Are product, implementation, operating, and commercial assumptions visible?
- Can the evaluation team realistically review everything requested?
- Do the RFP, response template, scorecard, demo script, and commercial model use the same requirement identifiers?

What Comes Next

The written response creates an evidence baseline. Carry the same priorities and requirement identifiers into:

- Scripted demos using common workflows, representative data, and edge cases
- Technical, security, and architecture diligence
- Reference conversations with comparable customers
- Commercial normalization and contract review
- Implementation-readiness assessment across data, process, people, governance, etc.

Before selecting a preferred solution, document not only which vendor ranked first, but why the organization is willing to accept the accompanying trade-offs, spelling out which assumptions, commitments, responsibilities, and protections must appear in the contract.

The best-written proposal is still a hypothesis. Demonstrations, references, diligence, and implementation planning test whether the claim is credible in your environment.

How InnoVyne Helps

InnoVyne is a vendor-agnostic consultancy dedicated to providing products and services that support the full Sales Performance Management lifecycle. We help organizations define future-state requirements, evaluate platforms, select implementation approaches, and prepare the business context needed for long-term success.

www.innovyne.com



Your Partner in Sales Performance Management