

NOBODY NEEDS YOUR DEMO ANYMORE: FIELD GUIDE

A practical companion to the article. The essay makes the case. This turns it into moves you use on live deals. Take what fits the deal in front of you and leave the rest.

1. Shape the request before you accept it

A customer asks for a demo, a workshop, a proof of concept, an architecture review, or a document by tomorrow. Before you start the work, run five questions.

- What are you trying to decide?
- What concern sits behind this request?
- Who needs the answer?
- What happens once we provide it?
- Is there a simpler way to answer the question?

If the answers are unclear, the work is not ready. Shape it before you accept it.

2. Find the problem beneath the requirement

A requirement tells you what the customer says they need. It rarely tells you why. Walk the chain until the requirement connects to a real consequence.

Requirement → reason → effect → evidence needed

Worked example:

- Requirement: the platform must support the existing approval process.
- Reason: the customer will not change its operating model this year.
- Effect: a different process would delay adoption and add work.
- Evidence needed: the customer sees the approval flow working under its own constraints.

Now you know what matters, what to show, what to test, and what to leave alone.

3. Separate facts from assumptions

Speed pressure produces confident guesses. Keep a clean line between what you know and what you believe by labelling every material point.

Label	Meaning
Known	Confirmed by reliable evidence

Label	Meaning
Assumed	Believed, not yet confirmed
Open	Still unanswered
Dependent	Needs another condition, person, or system
Risk	Affects success, cost, effort, or timing

Use these labels in your notes, technical reviews, proof plans, and handovers. Do not let an assumption become a fact because it survived three meetings.

Phrases to catch yourself using without evidence:

- "That should work."
- "It's configurable."
- "We support that."
- "The integration is straightforward."
- "I can't see that being a problem."

When the evidence is thin, say so, then say how you will close the gap. A controlled answer beats a confident guess.

4. Run the demo as a working session

A polished demo that helps nobody decide is a failed demo. Give the session a job and hold it to four steps.

1. Set the question. "By the end of this session, we want to establish whether..." State it out loud and agree it.
2. Show only what matters. Remove anything that does not help answer the question.
3. Pause and check. Ask what the customer saw, what is still unclear, and whether the evidence was enough.
4. Record the result. State what was confirmed, what was not, and what happens next.

A demo succeeds because the customer decides better, not because it ran without errors.

5. Control the proof of concept before it controls you

A proof of concept without agreed success criteria is unpaid exploration. Agree the full set before anyone touches a keyboard.

- The question being tested
- What is in scope
- What is out of scope
- The success criteria
- The evidence to be captured
- The customer's responsibilities
- Who judges the result
- The dates
- The exit conditions
- What happens after success
- What happens after failure

Design the proof to find the truth, not to manufacture a positive result. A failed test that exposes a poor fit before the contract is cheaper than the same discovery during delivery.

6. Translate the consequence, not the technology

A technical fact is not useful until the customer understands its effect. Complete the thought.

Technical fact → practical effect → decision impact

Worked example:

- Technical fact: the solution requires access to this data source.
- Practical effect: the customer's data team must approve and prepare the connection.
- Decision impact: if that team cannot support the work this quarter, the deployment date is at risk.

Then pitch the depth to the person in front of you.

Audience	What they need
Architect	The dependency and the design detail
Security lead	The controls and the remaining exposure

Audience	What they need
Operations lead	The effect on process and ownership
Executive	The effect on time, cost, and risk

Answer the question first, explain the consequence, add detail only where it helps, then stop.

7. Make every meeting produce something

A meeting is not progress because it happened. An important customer interaction should leave behind at least one of these:

- A confirmed fact
- A new question
- An agreed criterion
- A recorded risk
- A decision
- A named owner
- A dated next step
- A change to the approach

Close every meeting able to answer: what did we learn, what changed, what is still open, who owns it, and what happens next.

Replace vague endings with a specific one.

- Weak: "We'll send something over."
- Better: "We'll confirm the integration dependency by Thursday. The customer's security lead reviews the access model. We then decide whether a technical test is needed."

8. Prioritise the work

Demand has no natural limit. Before you start a piece of work, ask:

- Is this tied to a real opportunity?
- Is the customer ready?
- Does the activity support a decision?
- Is the expected value worth the effort?

- What has the customer committed?
- What other work will this delay?
- When should we stop?

If the answers are thin, the work is not ready for your time.

9. Use tools against real friction

Start with the friction, not the tool. Find where you are:

- Repeating work
- Losing information
- Relying on memory
- Rewriting the same documents
- Copying information between systems
- Giving inconsistent answers
- Failing to reuse what you learned last time

Then pick a tool that removes that friction. Before you trust any AI output, run four checks:

1. Is the source reliable?
2. Is the answer correct in this customer's situation?
3. Can I explain and defend it?
4. Am I willing to take responsibility for it?

AI removes low-value effort. It does not take over judgement. It will also produce a tidy summary of a poor meeting and repeat a wrong claim with a straight face, so read before you rely.

10. Make your work reusable

Do not start from a blank page unless the work is new. Keep a small set of working assets and reuse the stable parts.

- Meeting preparation template
- Discovery note
- Assumption and risk log
- Demo plan
- Proof charter

- Technical decision summary
- Handover record
- Deal review format

A template should prevent important omissions, not create admin. Keep them simple. Good work should be easier the second time.

11. Learn while the deal is still live

Review the work while there is still time to change it. On the deal:

- What do we believe, and what evidence supports it?
- What is still unclear?
- What could stop the deal?
- Which person is missing?
- Where are we spending too much effort?
- What are we avoiding?

On your own performance, skip "how did I do" and ask sharper questions:

- Which part was unclear?
- Where did I go too deep?
- Which question produced the most useful answer?
- What assumption did I fail to test?
- Which risk did I miss?
- What should become reusable?

Ten years of repeating the same weak habit is still repetition. Improvement needs honest review and a change in behaviour.

12. Protect the handover

The job does not end when the customer accepts the technology. A weak handover loses the information delivery depends on. A useful one carries:

- What the customer is trying to achieve
- What was agreed
- What was proven

- What was assumed
- What remains open
- What was not promised
- Who owns each next step
- Where delivery risk remains

The handover is part of the sale. A technical win that creates a poor delivery experience is not a complete win.

13. For managers: coach judgement, not compliance

Individual habits survive or die on the conditions you set. When a Sales Engineer brings you a deal, resist taking it over. Ask:

- What do you think is happening?
- What evidence supports that?
- What are the options?
- What are the risks?
- What do you recommend?
- What would change your view?

Rescuing every deal buys short-term results and long-term dependence. The aim is people who think clearly without you in the room.

Copyright © 2026 Apex Edge Sales Engineering Limited. All rights reserved.

Certain names, logos, and other marks are trade marks of Apex Edge Sales Engineering Limited in the UK and other jurisdictions. A current, non-exhaustive list is set out in our [Trade Marks](#) notice.

Registered in England and Wales | Company No. 15821626 | ICO Registration No. ZB796431
71–75 Shelton Street, Covent Garden, London WC2H 9JQ, United Kingdom

Disclaimer

This document is provided for general information purposes only. It does not constitute legal, financial, or professional advice and should not be relied upon as such. While we aim for accuracy, Apex Edge Sales Engineering Limited accepts no liability for loss, damage, or consequences arising from reliance on this material. Content is provided “as is” and may be updated or withdrawn without notice. This disclaimer is governed by the laws of England and Wales.

Intellectual Property Notice

Unless otherwise stated, all content, including text, graphics, diagrams, and branding, is the intellectual property of Apex Edge Sales Engineering Limited. Unauthorised reproduction, distribution, or commercial use is prohibited without prior written consent. Use of this material is subject to UK and international copyright, trademark, and intellectual property laws. Disputes shall be subject to the exclusive jurisdiction of the courts of England and Wales.