

Choosing the Right Policy Administration System for Your Insurance Business



Vendor-Neutral Buyer's Framework



10 Questions for Every Vendor



Backed by 2025 and 2026 Research

Contents

1.	The Decision That Outlives the People Who Make It	3
2.	What You Are Really Buying	4
3.	Why So Many PAS Projects Go Wrong	5
4.	The Products You Need Next Don't Exist Yet	7
5.	Four Roads to a New PAS	9
6.	Ten Questions to Ask Every Vendor	12
7.	Stress-Testing a Shortlist: A Product Nobody Has Built	14
8.	Where Openkoda Fits	16
9.	Conclusion: The Year-Three Test	19

WHO THIS EBOOK IS FOR

Insurers, MGAs and insurtechs getting ready to select or replace a policy administration system, and anyone on the buying committee who will have to live with the result.

What you will take away: a clear picture of what a PAS really has to do, why so many replacements go wrong, the four structural options on the market, ten questions that separate them, and a stress test you can run on any shortlist.

Authors



Arkadiusz Drysch

CEO & Co-Founder, Openkoda



Michał Głomba

CRO & Co-Founder, Openkoda



Arkadiusz Krysik

Marketing Manager at Openkoda

This ebook is distributed under the Creative Commons BY-ND 4.0 License. The license allows for redistribution, as long as the ebook is passed along unchanged and in whole, with credit to its authors.

The Decision That Outlives the People Who Make It

Every quote, every bound policy, every renewal and every claim your business handles will pass through your policy administration system. Few software decisions touch as much of an insurer, and fewer still stay in place as long.

That second point is the one buying committees tend to underestimate. A core system replacement is not a quarterly project. Roland Berger's 2026 study of core insurance transformations puts typical programs at **two to five years**, and once a system is live it tends to stay live. In West Monroe's 2025 survey of 300 US insurance executives, nearly half were running core platforms six to ten years old, and some were past fifteen. The people who choose a PAS are rarely the people who live with it at the end.

So it helps to picture the decision as it will actually play out, rather than as it looks in demo season. The questions that matter in year zero (does it cover our current products, does the dashboard look good, can we afford the licence) are not the questions that matter in year three or year eight.

The cost of getting it wrong doesn't arrive as a single bill. It arrives as a slow tax on everything else. Once a system can no longer keep up, the budget quietly shifts from building new things to keeping old ones alive, and the projects that would have made the business more competitive are the first to be cut.

The numbers from the last two years are blunt. More than half of US insurers now spend over half of their IT budget simply maintaining the systems they already have. Almost all of them have had to delay or cancel a strategic technology program because the money wasn't there. And the waste isn't only in the IT department: much of it is hidden in the hours underwriters and operations teams spend working around their own systems. Research by RSM US for INTX Insurance Software puts the hidden cost of legacy systems at **up to \$5 million a year** and finds that more than half of policy workflows still need manual work.

None of this means modern systems are cheap or easy. It means a PAS decision is never just an IT decision. It decides which products you will be able to launch in three years, how many people you will need to launch them, and how much of your roadmap will be set by someone else's release calendar. The rest of this ebook is a framework for making that decision on purpose.

FIGURE 1

Where legacy systems hurt

Share of US insurance executives, 2025



Source: West Monroe, 2025 (300 US insurance executives), via Insurance Business

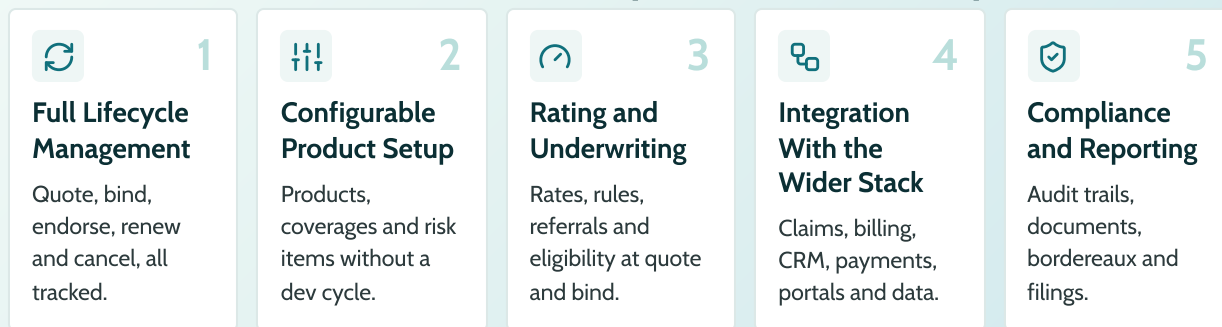
CHAPTER 2

What You Are Really Buying

Strip away the vendor language and a policy administration system has one job: to be the single source of truth for every policy you write, from the first quote to the last renewal or cancellation, and to hand that truth to underwriting, billing, claims and every portal built on top of it. It is the system of record for the promise you make to a customer. Everything else is detail.

The detail is where evaluations are won or lost, though, because almost every vendor will say yes to the five functions below. They are table stakes. What actually separates systems is a question most RFPs never ask directly: **how much work does each function take to change, and who is allowed to do it?**

The Five Core Functions of a Policy Administration System



Now picture a specific product, ideally one you don't sell yet. A new coverage touches all five core functions at once. It needs new fields in the product definition, new factors in the rating, new referral rules, a new document template and new lines in your bordereaux. If each of those changes is a ticket in someone else's queue, your launch date belongs to whoever owns the queue.

Configuration is not the same as customization

Vendors use the two words as if they meant the same thing. They don't, and the difference shows up in every budget from year two onward. **Configuration** means changing how the system behaves through settings it was designed to support: a new rating factor, a new referral rule, a new form field. It survives upgrades, and a trained business user can usually do it. **Customization** means changing the code. In a closed system only the vendor or its certified partners can do that, it is billed as a project, and it often has to be redone every time the vendor ships a new version.

So when you walk through a vendor's demo, ask which of the two each change on your list would require. The answer tells you far more about your future costs than the licence fee does. The best systems push as much as possible into configuration, and for whatever is left they let you customize freely, in a standard language, without breaking the upgrade path.

That is the lens for the rest of this ebook. A PAS that handles the products it shipped with is the minimum. A PAS that lets your team extend every one of those functions for the product you haven't designed yet is the thing you are actually paying for.

CHAPTER 3 —

Why So Many PAS Projects Go Wrong

If the stakes are that high, you would expect replacements to be run with great care. Most are. They still go wrong remarkably often: Roland Berger's 2026 study found that around half of core system replacement programs miss their time or budget targets, on programs that typically cost anywhere from €20 million to more than €100 million. The reasons are more consistent than you might think.

Roland Berger's explanation for the overruns is not exotic technology. It is two very ordinary things: **underestimated data migration and weak ownership**. Both get worse the more a program tries to change at once. BCG's broader research is harsher still, finding that 74% of corporate transformations across industries fail to create value in either the short or the long term. Insurance core programs are not an exception to that pattern. They are a textbook case of it.

Four failure patterns come up again and again.

The big bang. Replacing products, rating, underwriting, billing and claims in one cutover concentrates every risk into a single date. When that date slips, everything slips with it, and the old system has to be kept alive in parallel for longer than anyone budgeted.

Buying today's requirements. RFPs are written from the current product catalogue, so vendors are scored on how well they handle what you already sell. The system that wins is the one that fits the past best, which is not the same as the one that adapts best to what comes next.

Customizing instead of configuring. Every change that goes into code rather than configuration becomes something to retest, and often rewrite, at the next upgrade. Programs that start with a handful of “small” modifications regularly end up unable to upgrade at all, which quietly turns a modern system back into a legacy one.

Pricing the licence and ignoring the rest. The subscription fee is the one number everyone compares, and it is often the smallest part of the bill. Implementation, migration, integrations, the steady stream of change requests, upgrades and the eventual cost of leaving usually sit below the waterline.

It helps to put a number on the total before you start. BCG estimates that core IT modernization costs insurers **2% to 4% of non-life gross written premium** in capital expenditure. For a carrier writing \$500 million a year, that is \$10 million to \$20 million, which is exactly why the choice deserves more rigor than a feature scorecard can give it.

WHAT THIS MEANS FOR BUYERS

Favor approaches that let you move one product or one layer at a time, score vendors on the product you haven't built yet, and ask for the full cost of ownership, including the exit, in writing.

TOTAL COST OF OWNERSHIP

What the quote shows, and what you actually pay for



Illustrative. Relative sizes vary by program; data migration and ownership are the leading causes of overruns (Roland Berger, 2026).

The Products You Need Next Don't Exist Yet

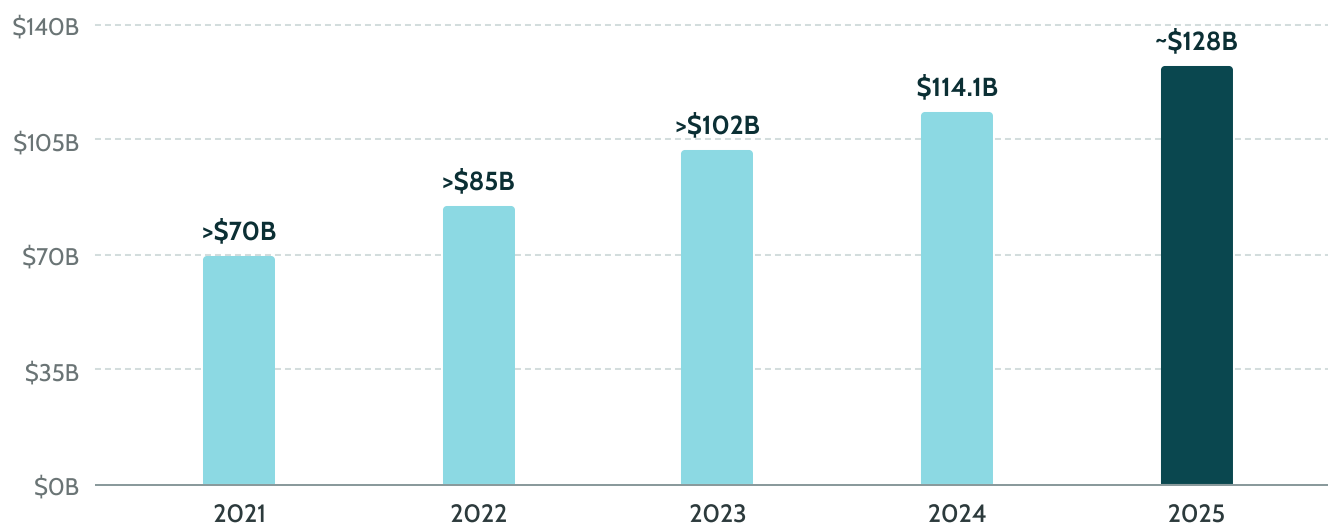
A PAS decision would be simpler if the market stood still. It doesn't. The fastest-growing corners of insurance are exactly the ones legacy systems were never designed for: specialty risk, delegated authority, embedded distribution and AI-assisted underwriting. The products your business will depend on in five years are, for the most part, products nobody has configured yet.

Start with delegated authority. Managing general agents have gone from a niche to one of the main engines of growth in US property and casualty. Conning's annual estimates show MGA premium growing by double digits every year since 2021, far ahead of the wider market. AM Best counts nearly 800 MGAs, growing at more than three times the pace of P&C as a whole.

FIGURE 2

US MGA premium, 2021 to 2025

Estimated premium written through MGAs, USD billions • double-digit growth every year



Source: Conning, annual MGA market studies (2022 to 2026 editions). Conning's scope widened over time; the 2025 estimate includes Lloyd's and other non-statutory premium.

MGAs are demanding customers for a PAS. They write unusual risks, often for several carriers at once, each with its own rules, capacity limits and reporting cadence. The same shift is visible in the specialty market as a whole. Surplus lines premium reached a record \$143.3 billion in 2025, the eighth year in a row of double-digit growth, and its share of the market has more than tripled since 2000.

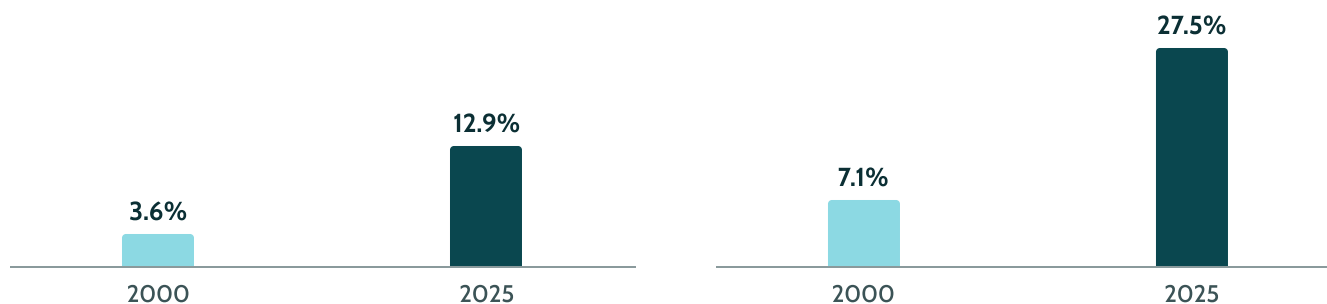
FIGURE 3

Surplus lines' share of US P&C premium, 2000 vs 2025

The business that doesn't fit standard products is no longer a rounding error

Share of all P&C direct premium

Share of commercial lines premium



Source: AM Best, 2026 surplus lines report, via Insurance Business

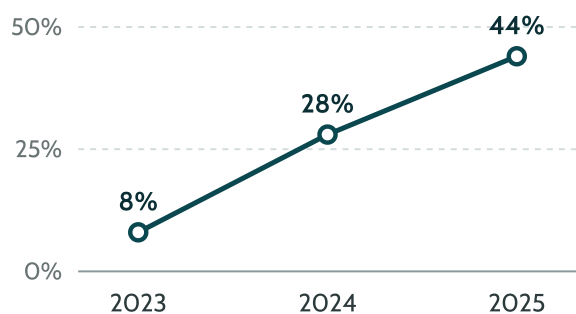
Distribution is changing shape too. Instead of waiting for customers to come to them, insurers are increasingly selling cover inside someone else's purchase: a travel booking, a car lease, a software subscription, a delivery. Deloitte, citing Conning and Simon Torrance, puts embedded P&C sales at around **\$70 billion in the US and \$700 billion globally by 2030**. Every one of those partnerships needs its own product variant, its own embedded form and its own data exchange.

Then there is AI, which has moved from pilot to production faster than most technology in this industry. According to Celent, the share of North American P&C carriers with generative AI in production went from 8% to 44% in two years, and underwriting is where much of it lands: summarizing documents, analyzing cases and ingesting submissions. In Europe, EIOPA's 2026 survey of 347 insurers found nearly two-thirds already actively using it.

FIGURE 4

Generative AI in production at North American P&C carriers

Share of carriers with GenAI in production



Top underwriting use cases



Source: Celent research, via hyperexponential, State of AI in Underwriting 2026

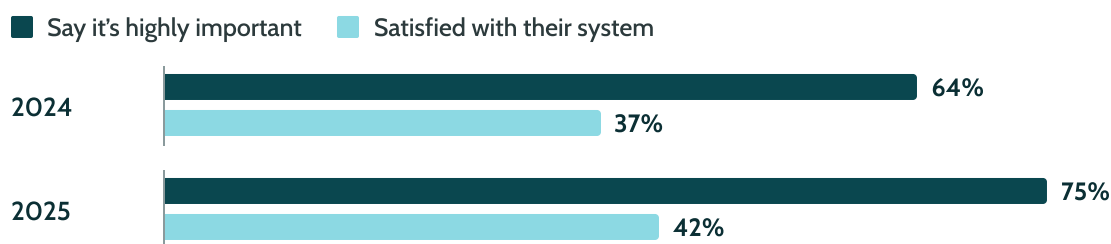
Put these four trends side by side and a pattern appears. Each one asks the PAS for something new: new risk attributes, new rating factors, new partners, new rules, new data flowing in from new places. The insurers who benefit will be the ones whose systems let them say yes quickly. Right now, most of the industry knows it can't. BriteCore's annual survey of P&C leaders shows the gap between how much launch speed matters and how satisfied carriers are with their systems' ability to deliver it. The gap is getting wider, not narrower.

The day-to-day reality behind that gap is easy to recognize. In West Monroe's survey, 46% of insurers said even a minor product update takes nine to sixteen weeks. That's a quarter of a year to change something small, in a market where partners, regulators and competitors move far faster. It is also the clearest argument for the main idea of this ebook: **when you choose a PAS, the most important thing you are buying is the ability to change it.**

FIGURE 5

Launching new products: how much it matters vs. how well systems deliver it

Share of P&C leaders



The gap grew from 27 to 33 points in a year

Source: BriteCore, Annual P&C Core Systems Reports, 2024 and 2025

CHAPTER 5

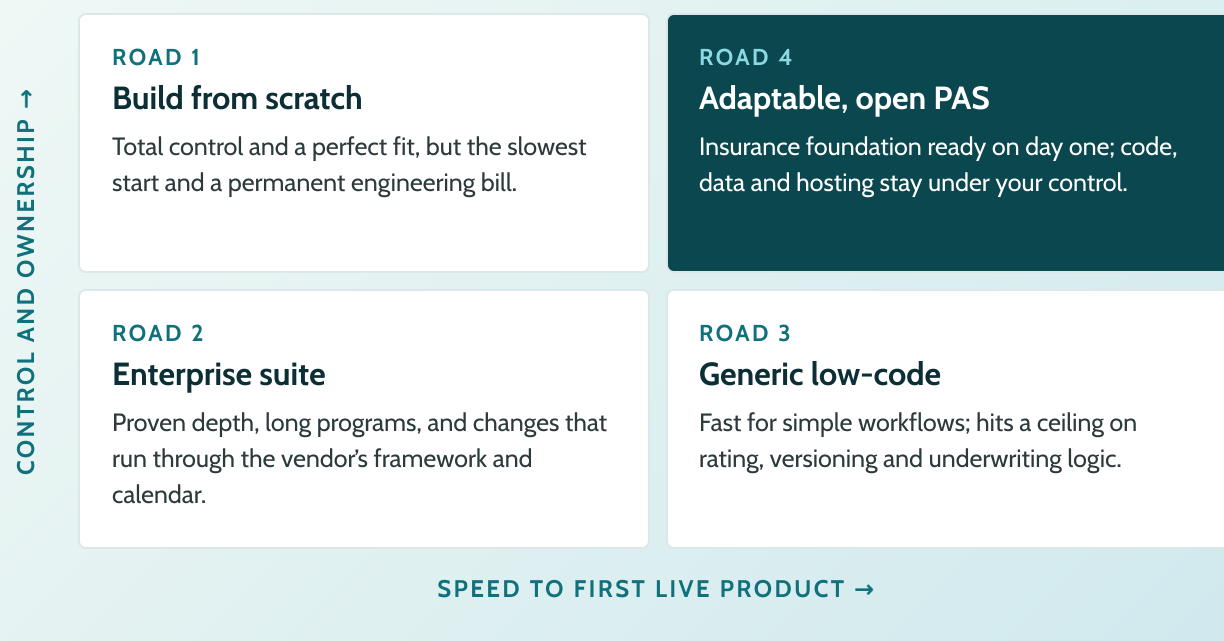
Four Roads to a New PAS

Every insurer, MGA or insurtech choosing a PAS is really choosing between four structurally different approaches, not a dozen vendors. It pays to decide on the road before comparing the cars, because the category determines what will still be true about your system in five years, long after the feature comparison is forgotten.

FOUR ROADS

Speed to launch vs. control over the result

Where each approach usually lands once the first product is live



Build from scratch. You get a system shaped exactly to your process and full ownership of every line of code. You also sign up to build everything that isn't insurance first: authentication, permissions, audit logging, document handling and admin screens. That is typically many months of work by a dedicated team before the first policy can be quoted, and the team that built it has to stay to keep it alive.

Buy an established enterprise suite. Deep, proven insurance functionality, reference customers and a large partner ecosystem. The trade-off is where the flexibility lives: inside the vendor's framework, often in a proprietary configuration language, on the vendor's release calendar and usually through vendor-certified consultants. You can change a great deal, but rarely on your own.

Adopt a generic low-code platform. Quick to stand up simple forms and workflows, and attractive to teams that want to move without developers. The abstraction that makes it fast is also what caps it. Multi-factor rating, versioned products, referral logic and bordereaux are not what general-purpose workflow tools were designed for, and teams tend to hit the ceiling a few months after go-live.

Start from an adaptable, open PAS. A production-ready insurance foundation (products, rating, workflows, documents, permissions and audit) built on a standard technology stack that you can own, host where you like and extend without limit. The foundation is shared. The products, rules and customer experience on top of it are yours.

No road is wrong for everyone. A large carrier with a permanent platform team and a ten-year horizon can justify building. A carrier replacing a stable, high-volume personal lines book may reasonably prefer a mature suite with a long reference list. What has changed is the default. Building from zero is harder to justify when solid foundations already exist, and low-code keeps hitting the same wall: insurance logic is more intricate than the workflows it was built for.



In custom insurance software projects, you don't need to reinvent the wheel. It's smarter to start from a solid foundation of ready-made application templates, RBAC, dashboards, and integrations, and build the specific logic you really care about.

Arkadiusz Drysch, CEO & Co-Founder, Openkoda

The comparison table sums up the trade-offs a buying committee usually spends weeks arguing over.

TABLE 1
Four roads to a new PAS, compared

Dimension	Build from scratch	Enterprise suite	Generic low-code	Adaptable PAS
Time to first live product	6 to 12+ months	Often 1 to 2+ years	Weeks to months	Weeks
Ceiling on customization	None	Within vendor framework	Reached quickly	None
Who owns the code	You	Vendor	Vendor	You
Typical pricing model	Team cost	Per user, module or premium	Per user or workflow	Often flat
Deploy on your own infrastructure	Yes	Sometimes	Rarely	Yes
Fit for niche and specialty risk	Yes, eventually	Depends on module	Usually not	Yes
Insurance foundation already built	No, you build it	Yes	Partly	Yes

The architecture question hiding inside the choice

There is a second decision tucked inside the first. For twenty years the assumption was that a PAS should do everything: hold the policy record, rate, run underwriting rules and provide the screen the underwriter works from. The analyst view has shifted. Gartner has predicted that core systems

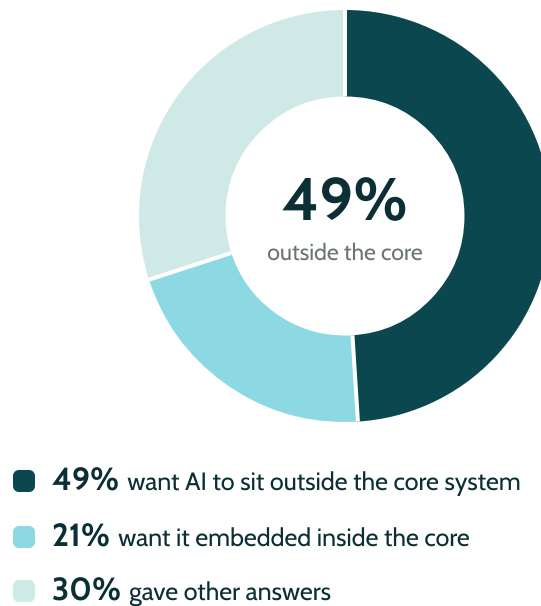
will increasingly be reduced to record keepers by 2030, and Celent now describes the underwriting workbench as the underwriter's new command center. The pattern gaining ground is a leaner policy core that owns the policy record, lifecycle transactions and financials, with underwriting, AI and customer channels in clearly separated layers around it.

Buyers are voting the same way. In BearingPoint's 2026/27 European core systems study, 49% of insurers want AI to sit outside the core system, against 21% who want it embedded inside. The appeal is less about elegance than about risk: separate layers let you modernize the part of the stack that actually differentiates the book, which is risk selection, without betting the company on one all-or-nothing replacement. The practical question for any shortlist is whether each of these layers can be configured on its own, and replaced on its own if it ever has to be.

FIGURE 6

Where European insurers want AI to sit

Share of insurers, 2026/27



Source: BearingPoint, P&C Core Systems Study 2026/27 (Europe). "Other answers" is the remainder.

CHAPTER 6

Ten Questions to Ask Every Vendor

Demos are built to make every platform look capable, and most of them are. The ten questions below are designed to find out what is true once the sales team leaves the room. They fall into three groups (speed, ownership and fit), because those are the three things that decide whether year three goes well. Ask them of every vendor on your shortlist, and ask them hardest of the one you already like best.

Speed

- 1 Can a product owner change a coverage, a rating factor or a form without a development ticket?**
Ask them to do it live, on a product they didn't prepare. If every change needs a developer and a release, your speed to market is capped by someone's backlog, not by your ideas.

- 2 What is the demonstrated time to a first live product, and can a customer confirm it?**
Ask for a reference, not a roadmap slide. Weeks and quarters describe two different categories of system.
- 3 Can underwriting rules and the underwriter's workspace be configured per product?**
Referral thresholds, decline conditions and the fields an underwriter sees should be something your underwriting lead can adjust, not a change request.



Ownership

- 4 Who owns the code, the configuration and the data after go-live?**
If the honest answer is “the vendor,” every customization you pay for becomes leverage in your next renewal negotiation.
- 5 Does the price grow with seats, premium or API calls?**
Pricing tied to users or a percentage of gross written premium means the system charges you more for succeeding. Model the cost at three times your current volume before you sign.
- 6 What does it actually cost to leave?**
Ask for the exit process in writing: data export formats, code handover, notice periods. If leaving means rebuilding from zero, you are renting, on a lease you haven't read.



Fit

- 7 Can the data model hold risks it has never seen before?**
A specialty product needs specialty fields. If the system can't add a horse's breed, a vessel's flag state or an AI agent's autonomy level without a vendor project, it wasn't built for you.
- 8 Can it run on your infrastructure as well as the vendor's cloud?**
Cloud strategies change. BearingPoint found the share of European insurers expecting to run fully on public cloud fell from 60% to 41% in a year, as more chose hybrid setups. Data residency rules or a regulator can force the question too, so know the answer before you sign.
- 9 Is a complete audit trail built into every transaction by default?**
Compliance bolted on afterwards is compliance you can't fully trust. Ask to see the log of a change that actually happened, not a slide describing one.
- 10 Can one installation run several brands, programs or carriers with genuinely separate data?**
If you run more than one book today, or expect to, this isn't optional. A filtered view of a shared database is not the same thing as real isolation.

None of these questions is exotic. Most buyers already half-know to ask them. The value is in asking them plainly, writing down the answers and comparing the answers side by side, instead of comparing demos.

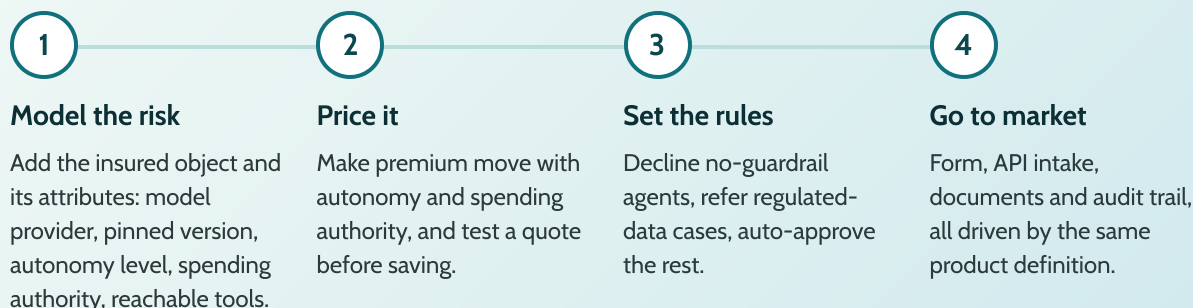
Stress-Testing a Shortlist: A Product Nobody Has Built

Checklists are easy to pass with a product the vendor has seen a hundred times. So test your shortlist against one they haven't. Here is a useful example: an MGA wants to insure **AI agents** (software that acts on its own, sometimes spends money and sometimes touches regulated data) as a liability risk. No off-the-shelf line of business exists for it, and none will for years. That is exactly what makes it a good test.

THE STRESS TEST

Four steps to put any PAS through an unfamiliar product

Run the same exercise on every system on your shortlist and time it



The data model (question 7). The insured object isn't a building or a vehicle, it's an agent. The underwriter needs to know who built the underlying model and which version is pinned, how autonomous the agent is (from "suggests" to "acts without a human"), how much money it can commit unsupervised and how many tools or systems it can reach. A system limited to predefined fields fails here immediately. One with an open, configurable data model adds these as real, structured attributes in an afternoon.

Rating and underwriting (questions 1 and 3). Premium should move with autonomy and spending authority, because that is what moves the exposure. The rules should decline a fully autonomous agent with no guardrails, refer one that handles regulated data without oversight and approve the rest automatically. That is ordinary underwriting judgment applied to an unfamiliar risk. The question is whether your product team can configure it or has to request it.

Everything downstream (questions 2 and 9). Because the new attributes are structured data rather than free text, the quote form asks exactly the right questions, the same submission can arrive from a person or through an API with identical validation, and every decision lands in the audit trail. A good system also lets you dry-run a test quote before launch, so a broken formula is caught before it prices a real policy at zero.

What to watch while they do it. The result matters less than how the vendor gets there. Notice who does the work: a product person clicking through configuration, or an engineer writing code off screen. Notice whether anything needs a deployment or a restart before it can be tested. Check whether a test quote can be priced before the change is saved, and whether the audit trail shows exactly what was changed, by whom and when. Ask how long the same change would take after go-live, without the pre-sales team in the room. Those observations tell you what your own team's Tuesday afternoons will look like for the next ten years.

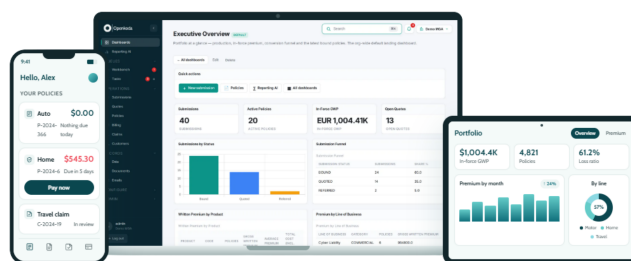
The point isn't that every insurer should cover AI agents. It's that sooner or later every insurer meets a product nobody has a template for, and that is the day the PAS decision made years earlier either pays off or sends you back to the vendor with a change request. Run one unfamiliar product through every system on your shortlist. It will tell you more than the rest of the evaluation put together.

Where Openkoda Fits

So how would Openkoda fare against those ten questions? Hold it to exactly the same standard as everyone else on your list. What follows is what sits behind the answers, shown in screenshots rather than slogans, with all ten summed up at the end.

Openkoda is a highly customizable policy administration system on the fourth road: a complete insurance foundation that is shaped around your products and then handed over as yours. It covers the full lifecycle in one connected system, from product setup, rating, forms and underwriting to policy servicing, billing, claims, documents, reporting and bordereaux. Two ideas shape how it is built. **Adaptability** means every layer bends to the way your business already works. **Freedom** means that once it does, the code, the data and the choice of where it runs stay with you.

It is built for the organizations that feel the pressures from chapter 4 most sharply: carriers launching new products without a multi-year program, MGAs and specialty insurers whose risks don't fit a standard data model (parametric, marine, cyber, niche commercial), mutuals that want to modernize without handing control to a SaaS vendor, and insurtechs, brokers and embedded insurance operators running several brands or programs from one installation.



One connected system. Underwriter and operations workspaces, executive dashboards and customer self-service, all running on the same policy data.

“Openkoda was born from a simple idea: insurers should be able to launch and evolve digital products fast, on their own terms, without vendor lock-in.

Michał Głomba, CRO & Co-Founder, Openkoda

Adaptability, layer by layer

A new product starts the way most real products do, from something that already exists. Your team can pick a template from a global product library, clone one of your own products or start from a blank page. From there, the product definition drives everything else.

Risk items and their attributes are defined on the product itself. Add a flood-risk dropdown or a roof-type field to a property product, mark it as used in rating, and it appears on the quote form and in the rating engine without anyone wiring it up by hand. This is how question 7 gets answered: a horse's breed, a vessel's flag state or an AI agent's autonomy level is simply another attribute.

Product Builder

1 Start 2 Product Basics 3 Risk Types & Coverage 4 Endorsements

RISK TYPES & COVERAGE

Add a risk item type by picking from the catalogue, or build a completely custom one.

Add a risk item type: + From catalogue... + Completely new type

LOCATION Insured Property × Remove

ATTRIBUTE CODE	LABEL	TYPE	ALLOWED VALUES / UNIT	REQUIRED?	USED IN RATING?	
address	Property address	Text	A B C or unit	☒	☐	×
postcode	Postcode	Text	A B C or unit	☒	☐	×
construction_type	Construction	Dropdown	Masonry / Brick Timber Frame C	☒	☒	×
year_built	Year built	Number	A B C or unit	☐	☒	×
flood_risk	Flood risk	Dropdown	Low Medium High	☒	☒	×

Risk types and coverage. Attributes, allowed values and whether each one feeds the rating, all configured on the product.

The same goes for underwriting. Referral, decline and auto-approval rules are configured per product, and so is the underwriter's workspace. Referred cases arrive with the reason already visible, the underwriter can approve different limits from those requested, and the premium recalculates from the approved figures on the spot. If a product needs an extra tab, an underwriter-only checklist or a manual loading line, the product owner adds it once and every case on that product picks it up.

Underwriter Workbench

SUB-OK-PROPERTY-PRIME-09FD5084

Premium recalculated from the approved limits — see the Pricing tab.

SUB-OK-PROPERTY-PRIME-09FD5084 In Review

Eleanor Whitfield - Openkoda Home

Referral reason: High-value building sum insured - refer to underwriter for manual review

Submission Insured Risk Items **Coverage** Pricing Documentation Tasks Approval

Requested coverage

COVERAGE	REQUESTED LIMIT	APPROVED LIMIT	DEDUCTIBLE
Buildings	500,000 EUR	450000	Buildings deductible
Contents	75,000 EUR	65000	Contents deductible
Property Owner Liability	2 000 000 EUR	1800000	Property Owner Liability deductible

Work Management

ASSIGNED UNDERWRITER

AU Admin User

WORKFLOW STATUS

New In Review

Awaiting Info Referred / Escalated

On Hold Quoted

Declined

Underwriter Workbench. A referred case with its referral reason, approved limits and a one-click premium recalculation.

AI speeds all of this up without taking control away. The AI Product Builder reads a plain-language product specification and proposes the full configuration (risk items, rating tables, form questions, underwriting rules and a connected workflow) as a line-by-line change set. **Nothing is applied until a person reviews and approves it.** Reporting AI does the same for questions about your book: ask in plain English and get a report you can check, refine and pin to a dashboard. The model sees the structure of your database, never the rows in it.

Freedom, in the details that matter later

Openkoda is built on a standard, widely used stack (Java, Spring Boot and PostgreSQL), so any capable enterprise development team can read and extend it. You can run it in Openkoda's managed cloud, in your own private cloud or fully on premises, and change your mind later. Every customer-facing touchpoint, from portals and forms to emails and documents, carries your brand and your domain rather than ours.

Pricing is a flat annual rate: no per-seat fees, no percentage of premium and no usage metering, so the system doesn't get more expensive as you grow. Implementation is done with you, not handed to you as a kit. Openkoda's own implementation team configures your products, rating, rules and branded portals alongside your people, and a tailored system is typically live in about three weeks.

OPENKODA AGAINST THE CHECKLIST

Ten questions, ten answers

1 Change products without a ticket

Coverages, rating tables, forms and rules are configured per product, versioned and testable before they go live.

2 Time to a first live product

Typically about three weeks, working with Openkoda's implementation team.

3 Underwriting configured per product

Referral, decline and approval rules plus the workbench layout live on each product.

4 Who owns code and data

You do, on a standard Java, Spring Boot and PostgreSQL stack.

5 Price that grows with you

Flat annual rate. No per-seat fees, no percentage of premium, no usage metering.

6 Cost of leaving

Your code and data in standard formats, so leaving never means rebuilding from zero.

7 Risks it has never seen

Custom risk items and attributes flow into forms, rating and underwriting automatically.

8 Your infrastructure

Managed cloud, private cloud or on premises, and you can switch later.

9 Audit trail by default

Every create, update and delete is logged automatically: who, when and which fields changed.

10 Several brands or programs

Many organizations on one installation, each with its own products, branding and isolated data.

Don't take any of this on trust. The fairest test is the one from chapter 7: bring a product you haven't launched yet, ideally the awkward one, and watch it being configured. If the ten answers hold up against your own risk, rules and rating, you will know. If they don't, you will know that too, and you will have a much sharper checklist for everyone else on your list.

CONCLUSION —

The Year-Three Test

Every PAS shortlist eventually collapses into a feature comparison: whose dashboard is nicer, whose form builder is smoother, whose integration list is longer. Those comparisons feel important in the room and matter surprisingly little afterwards. Almost every serious vendor can demo a convincing dashboard.

What almost none of them can demo is year three, when your roadmap has moved somewhere no demo script anticipated. A new distribution partner wants an embedded product. A capacity provider wants a different bordereau. Your underwriters want to price a risk that didn't exist when you signed. That is the real test of the decision: not whether the system does what you need today, but whether it lets your own team build what you need next, without asking permission, without a change order and without discovering that what you thought you owned was only ever licensed to you.

So take the ten questions into your next vendor meeting. Ask them of the system you already like best, not just the ones you are comparing it with. Run one unfamiliar product through every option. The answers, not the demo, are what you will live with for the next decade.

BEFORE YOUR NEXT VENDOR MEETING

Three steps that sharpen any PAS evaluation

1

List your next three products

The ones on the roadmap, not the ones you sell today. Score vendors against those.

2

Send the ten questions in advance

Get written answers first, then use the demo to check them.

3

Run the stress test live

Same unfamiliar product for every vendor. Time how long it takes.



TAKE THE NEXT STEP

Launch your next insurance product in weeks, not years.

Book a personalized demo with Openkoda's implementation team. We'll walk through your own products, rating and underwriting rules, and show you what it takes to run them on a policy administration system you fully own.



Live in about three weeks



Flat annual rate



Your code, your data

openkoda.com/demo

About Openkoda. Openkoda is a highly customizable policy administration system for insurers, MGAs and insurtechs, covering the full lifecycle from quote to claim. It is configured around your products and handed over as yours: your code, your data and your choice of cloud or on-premises hosting, delivered with Openkoda's own implementation team on a flat annual rate.

© Openkoda · openkoda.com · Distributed under the Creative Commons BY-ND 4.0 License.