Seezo

Build vs. Buy

Threat Modeling & Security Design Review

Executive summary

For most security organizations, buying a purpose-built platform for secure design review and threat modeling delivers full coverage sooner, at a more predictable cost and with less operational risk than building one in-house.

Language models make a convincing prototype easy to build in a few weeks, but the harder and more expensive work comes afterwards, when the tool has to produce consistent, defensible results for every feature the business ships. This paper sets out what each path involves to help your team decide.

The problem: a security review for every feature

Engineering teams now ship features weekly, and AI coding agents are speeding that up, while a manual security design review can take days or weeks. As a result, many organizations review less than 15% of what they build, and design flaws in the features that go unreviewed become far more expensive to fix once code is written or running in production.

What an in-house build actually requires

A production-grade internal tool involves much more than connecting a language model to a design document, and the ongoing commitment is usually larger than the initial build. It requires security expertise, quality controls, integrations, and ongoing engineering effort to keep the tool reliable as your security requirements and models change.

Security content	Vetted security standards, threat patterns and compliance mappings (PCI DSS, ASVS, HIPAA, SOC 2) developed and maintained by AppSec practitioners.
Quality engineering	LLM outputs are non-deterministic, so consistent results require structured rules, test scenarios and an evaluation pipeline that catches regressions.

Model management	New models arrive every few months, and each must be evaluated for accuracy, cost and fit before it is adopted.
Integrations and workflow	Connectors across IT service management (ITSM), source code management (SCM), governance, risk and compliance (GRC), vulnerability management and productivity tools, plus access control, audit logging and privacy controls.
People	Engineers with both AppSec and mature AI development skills are scarce, and a tool that depends on one or two of them carries a key-person risk.
Maintenance and operating costs	The tool requires ongoing maintenance as models, prompts, security standards, and integrations change. This includes rerunning evaluations, fixing regressions, updating security content, and managing infrastructure, hosting, and token costs. These expenses grow with usage and the effort required to keep the tool reliable.

Side-by-side comparison

What matters	Build in-house	Buy Seezo
Output quality and consistency	Raw LLM analysis of a document, with results that drift from run to run	1,500+ security design rules written by security researchers, producing consistent, repeatable output
Cost model	Cost involves engineering time to build, ongoing upkeep, evals and regression testing, model and hosting spend, integrations, and AppSec time taken from reviews. Additional variable model and hosting spend that grows with usage	Predictable, fixed cost, with Seezo (token costs, upkeep, evals and integrations included)
Model selection	Your team evaluates and selects the right model for each task on every release	Seezo benchmarks which models fit which tasks as new models ship

What matters	Build in-house	Buy Seezo
Maintenance and accountability	Your team maintains the tool and owns the quality of its output	No upkeep on your side, Seezo maintains the controls, standards mappings and model testing, and is accountable for output quality
Time to value	Days to weeks for a prototype and months for a production system, followed by ongoing work to keep it running	Producing security guidance from day one, typically validated in a 2 to 4 week POC
Staying current	Security standards, threats and compliance mappings can go stale over time. Each standard update is manual work: research, gap analysis, remapping, regression testing. Mapping by LLM alone cannot confirm complete requirement coverage	Continuously updated standards and PCI DSS, ASVS, HIPAA and SOC 2 mappings. Updates are researched, remapped and regression-tested by Seezo
Customization	Adapting the tool to your organization means adding your organizational context, security posture and standards. Each change to a prompt or check can shift results system-wide, so every change needs full retesting	Seezo maintains a context layer that captures your security posture, and your security standards. You can add or change security standards independently without affecting other controls. The Seezo team handles the changes and testing at no additional cost
Expertise and risk	Hard-to-hire AppSec and AI development talent, with key-person risk if they leave	Researcher expertise built in, improving as Seezo's research team grows its standards
Review coverage	Limited by internal engineering capacity and ongoing maintenance requirements	Scale reviews across features without proportionally increasing AppSec headcount. Customers like Sidekick Security report 3× faster security design reviews and 100% feature coverage.

Total cost and time to value

We suggest comparing a three-year total cost of ownership rather than the cost of a prototype.

THREE-YEAR TOTAL COST OF OWNERSHIP (TCO)

TCO = **Initial development** + **Engineering salaries** + **Security research** + **Model and hosting costs** + **Evaluations and testing** + **Integrations** + **Ongoing maintenance** + **Token costs** + **Opportunity cost of AppSec engineering time**

Opportunity cost is the AppSec engineering time diverted to building and maintaining the solution, instead of performing core security activities.

Cost component	Build in-house	Buy Seezo
Engineering and development	Internal engineering resources	✓ **Included**
Security research and standards mapping	Internal AppSec resources	✓ **Included**
Model and hosting costs	Variable, usage-dependent	✓ **Included**
Evaluations and testing	Internal engineering effort	✓ **Included**
Compliance updates	Research, mapping and validation	✓ **Included**
Integrations and maintenance	Build and maintain internally	Managed by Seezo
AppSec opportunity cost	Time diverted from security priorities	Reduced internal maintenance burden

Seezo is priced as a flat-rate platform, so the cost per review falls as coverage grows. Time to value matters as well: Seezo assessments complete in minutes from existing PRDs, tickets and diagrams. Sidekick Security, for example, reported 3× faster security design reviews and 100% feature coverage.

When building makes sense

Building in-house is a valid option, but the decision shouldn't be based on how quickly you can build a prototype. Consider the effort required to maintain it, keep it accurate, and adapt it as models and security standards change. Building in-house may be appropriate if you have:

- A dedicated AppSec team with the expertise to validate security findings, design and run evaluations, and maintain LLM-based systems in production.
- Highly specialized internal security requirements.
- Existing evaluation and security research capabilities.
- A strategic requirement to own and maintain the entire solution.

> **A hybrid approach.** The decision does not have to be all or nothing. Seezo's security controls, standards mappings, evaluation approach and integrations provide the base, so your team does not build or maintain them. Your team adds what is specific to your organization, such as your own security standards and organizational context. Seezo tests each addition so results for other controls do not change. This suits teams with specialized requirements that do not want to carry the evals, model testing and standards upkeep of a full build.

Making the decision

The right answer depends on your starting point, and it is best reached by testing the options against the same criteria. These questions can help frame the internal discussion:

1. **Capacity:** Do you have AppSec engineers to build and maintain the tool for several years, and what would they stop doing?

   Building the tool is only the beginning. AppSec engineers still need to maintain it, rerun evaluations after model or prompt changes, update security controls, and fix regressions. This work takes time away from design reviews and other security priorities. Account for that before committing to build.

2. **Quality bar:** How will you measure accuracy and consistency from run to run, and who will own that over time?

3. **Total cost:** What does a three-year view look like once salaries, token and hosting costs, integrations and content updates are included?
4. **Speed:** How long can you accept partial design review coverage while a solution is built or rolled out?
5. **Ownership:** Is owning this capability a strategic goal in its own right, or is the real goal simply to have every feature reviewed?

NEXT STEP

Whichever way you lean, a short, time-boxed evaluation of Seezo on your own PRDs, tickets and architecture documents, with success criteria such as accuracy, coverage and time saved agreed up front, gives your team an objective basis for building, buying or combining the two.

Seezo

See Seezo on your own designs.

Run a short, time-boxed evaluation on your PRDs, tickets and architecture documents, and decide with real results.

Get in touch
sales@seezo.io

Learn more
seezo.io

Seezo, Inc.
355 Bryant Street, Unit 403
San Francisco, CA 94107