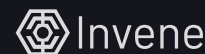


Cradling the AI: How SNOMED Enables Structure, Not Solutions



Rebecca Loar and Alex Melton | June 5, 2025

SNOMED CT has long been the common language healthcare systems claim to support, but rarely use well. Despite broad adoption on paper, especially across national systems like the NHS, most deployments treat it as a checkbox instead of a design principle. This leads to predictable results: limited adoption, low trust, and almost no impact on outcomes.

The problem isn't SNOMED itself. The issue is that SNOMED is mistaken for a solution. What it actually offers is infrastructure—a semantic framework for systems that understand what they're trying to do. Used well, SNOMED enables everything from cleaner documentation to better model performance. Used poorly, it becomes a dense vocabulary buried inside brittle workflows.

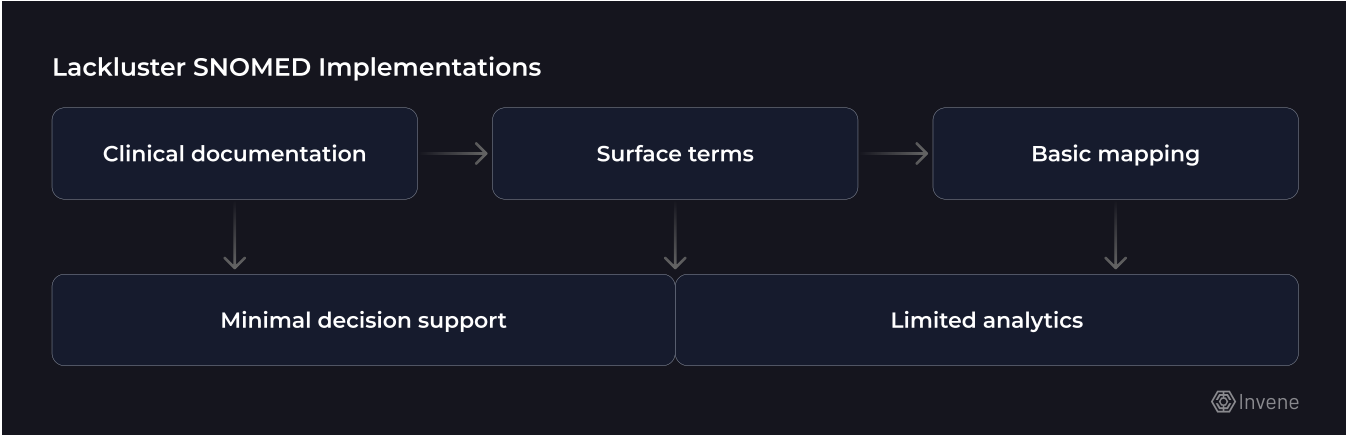
This article explores what SNOMED is (and isn't), why organizations often get it wrong, and how it can be treated not as a bolt-on but as a foundational element in systems that need structure in order to work.

A Tool Misunderstood

SNOMED CT is not a classification system like ICD. It isn't designed for billing or public health surveillance. It's a reference terminology: a clinical ontology meant to describe the real world in a format machines can process. With over 350,000 concepts and a deep hierarchy, it encodes meaning and relationships, not just terms.

But like any tool, it doesn't work on its own. SNOMED doesn't generate documentation or resolve ambiguity. It provides structure that allows those things to happen, assuming the surrounding system is designed with that in mind.

Unfortunately, most aren't. SNOMED often gets layered onto systems that were never built to support structured thinking—retrofitted UIs, rigid problem lists, or interfaces that default to free text. The result is either clutter or avoidance. Either SNOMED becomes overwhelming or it gets bypassed entirely. In both cases, the value is lost.



What Good Looks Like

When SNOMED is implemented with intention, it functions as a force multiplier. In the NHS, one hospital improved its pain assessment process simply by switching from free text to structured SNOMED-backed input. Clinician behavior didn't change. The workflow stayed the same. Quality scores tripled. Safety metrics improved. Time to extract data dropped by more than 70 percent.

In another case, a major NLP vendor saw a jump in precision after incorporating SNOMED relationships into its entity recognition pipeline. When the model was trained to treat “heart attack,” “myocardial infarction,” and “STEMI” as variants of the same concept, it began clustering usefully rather than guessing.

Even in billing workflows, SNOMED helps. When clinicians document with SNOMED and the system automatically maps those terms to ICD or CPT codes, specificity improves, coding errors go down, and back-and-forth with revenue cycle teams is reduced. Clinicians aren't forced to think like coders, but the backend still works.

In all of these cases, the key wasn't the vocabulary itself. It was the surrounding infrastructure that made it usable.

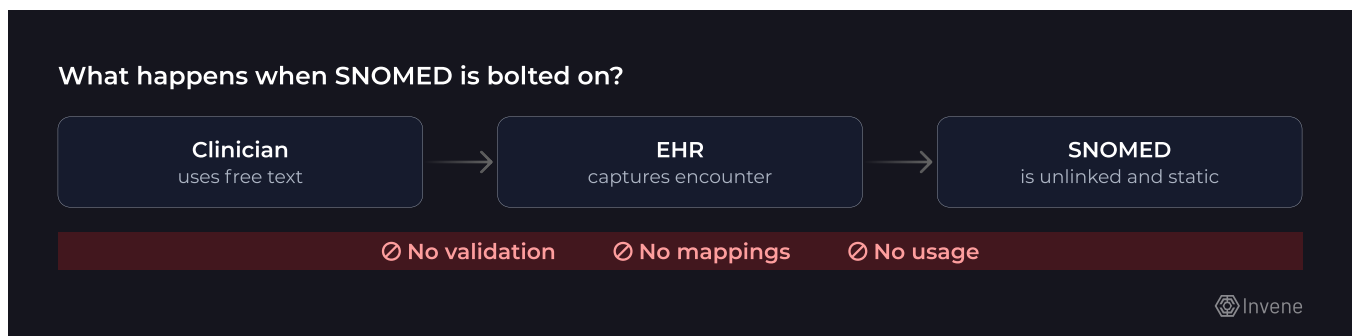
Why Most Teams Get It Wrong

Most failures fall into two categories. On the product side, SNOMED gets added too late to influence interface design or workflow logic. On the technical side, it gets treated like a reasoning engine that can infer or validate data on its own.

Neither approach works.

SNOMED isn't a logic engine, and it isn't a UI. It's a structured map of meaning. To make it work, you need tools that interpret it, workflows that depend on it, and models that can operate within its boundaries.

Without those things, it sits untouched, no matter how powerful it could be.



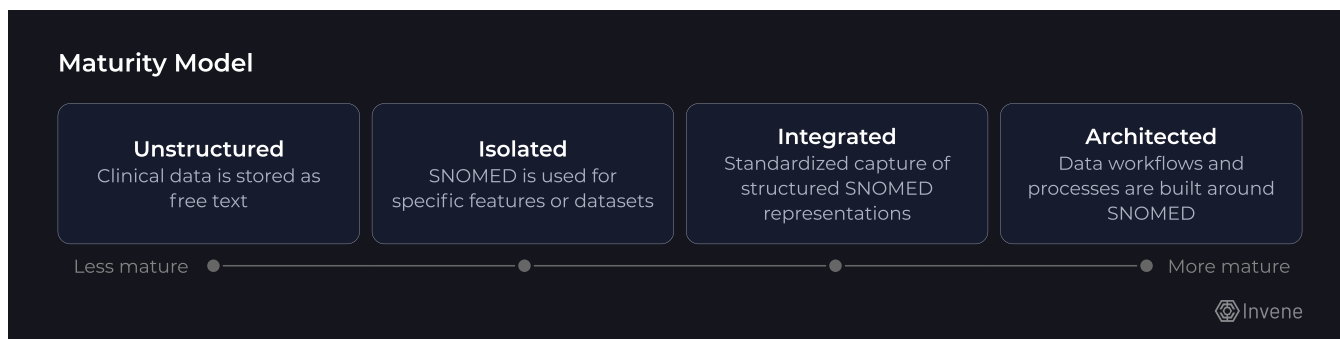
Cradling the AI

This becomes especially important when working with large language models. General-purpose models struggle with rare or domain-specific clinical terms. Prompting with SNOMED definitions can help. Verifier models that compare outputs against SNOMED

hierarchies can catch errors. And SNOMED can be used to generate structured data for training or QA purposes.

But the value isn't in SNOMED alone. It comes from the system design. You need scaffolding around the model—structured prompts, reliable mappings, and clear validation pathways. SNOMED helps only if you've already made the decision to build a system that needs structure in the first place.

That's what we mean when we talk about cradling the AI. The goal is not to constrain creativity. It's to create boundaries that allow the system to do something meaningful.



Levels of Use

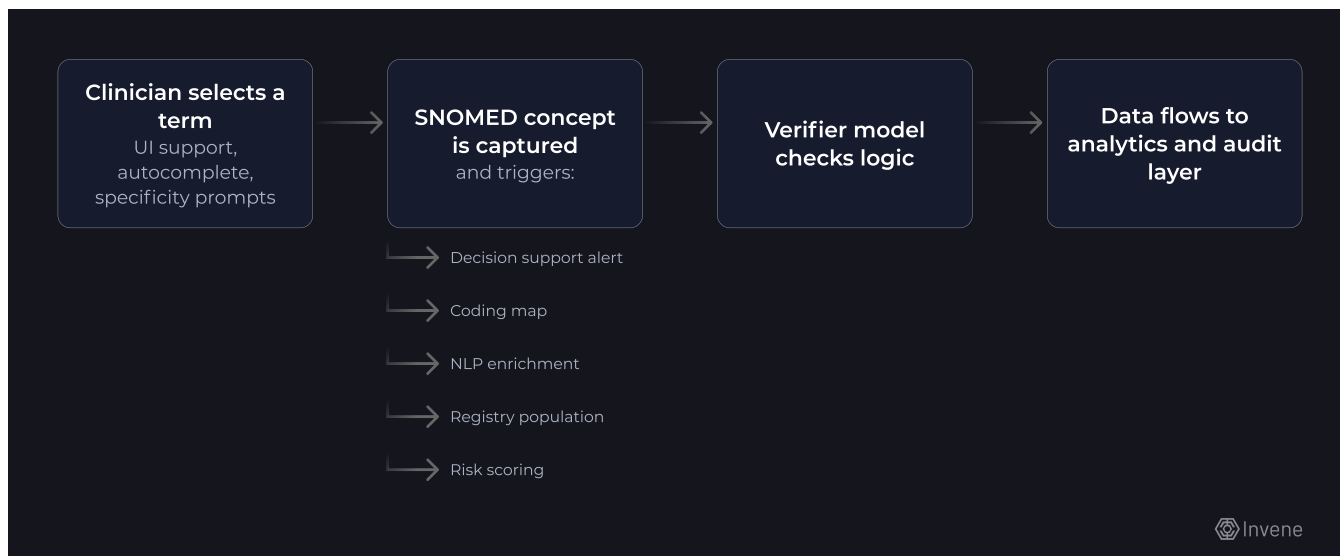
SNOMED adoption tends to evolve through a series of increasingly effective use cases. Each stage reflects a deeper understanding of how structure supports real-world utility.

Prompt injection is the most basic application. SNOMED terms or definitions are inserted into the prompt to help a language model disambiguate clinical concepts. It's quick to implement but limited by token space and context sensitivity.

Verifier models are a stronger approach. A lightweight secondary model checks LLM output for alignment with SNOMED logic. If the output contradicts a defined relationship or omits required attributes, the verifier can flag or revise it.

Inference and ensembling take this further. SNOMED's internal structure allows systems to reason about what terms imply. For example, if a clinician documents a traumatic brain injury, the system can check for required supporting details and suggest related conditions. When this logic is combined with a verifier model, the system becomes more resilient to edge cases.

Synthetic data generation is the most advanced application. SNOMED can be used to generate structured examples that train or fine-tune domain-specific models. It also helps simulate edge cases for testing and validation. Very few organizations are doing this, but the potential is enormous.



What Real Implementation Looks Like

Teams that succeed with SNOMED build around it intentionally. They don't just load the terminology into their EHR and walk away. They design interfaces that help users find the right term without breaking focus. They build decision support that uses SNOMED codes to trigger alerts, recommend care pathways, or populate registries. They map SNOMED to billing codes behind the scenes, so documentation drives revenue without changing clinician behavior.

They also invest in training. Not to teach clinicians the entire SNOMED hierarchy, but to show how structured input improves everything downstream—from coding accuracy to risk scoring to patient handoffs.

One of the clearest lessons from recent deployments is that specificity matters. When clinicians document in vague terms, downstream systems can't act. When they use detailed, SNOMED-coded inputs, everything from model accuracy to population health analytics improves.

Again, this isn't about SNOMED as a solution. It's about designing a system that knows what to do with structured data.

Conclusion: The Scaffolding Is the Product

SNOMED won't fix your workflow. It won't make your model accurate. It won't prevent errors. But it gives you the structure you need to build systems that do all of those things—if you're willing to treat it as infrastructure, not decoration.

The teams that succeed aren't just integrating SNOMED. They're designing for it. They're thinking in terms of interpretable output, shared language, and system-wide consistency.

This is especially important now, as generative models scale. The further these systems reach, the more they need something to stand on. SNOMED is one of the best foundations we have.

Use it accordingly.

Interested in starting a conversation?

Discuss SNOMED or your next big idea with one of our experts.