



Satya's practice focuses on technology transactions, intellectual property, and commercial agreements for startup and growth-stage technology companies across AI, robotics, cloud computing, and emerging tech sectors.

*Email: snarayan@gcalaw.com
LinkedIn Newsletter: [Techline Brief](#)
Connect on [LinkedIn](#)*

BUILDING WITH AI: HOW TO THINK ABOUT IP PROTECTION WHEN YOUR CODE IS AI-GENERATED

[Satya S. Narayan](#)

April 28, 2026

If your team is writing software without generative AI, your competitors may soon be outpacing you. If your team is using it without proper controls, you may be building a codebase you don't actually own.

That second risk is less obvious, and it tends to surface late. A company builds a product using AI coding tools, ships it, and begins attracting acquisition interest. During diligence, buyer's counsel asks about a portion of the codebase. The team cannot point to design documentation, cannot explain which decisions were made by developers versus the model, and cannot clearly articulate the basis for claiming the code as the company's original expression. The deal doesn't fall apart, but the valuation conversation gets uncomfortable and the reps and warranties negotiation gets harder. The AI tools saved months of development time. The lack of governance cost something too.

What the team also hadn't anticipated is that the buyer's counsel didn't stop at the AI-generated code. They asked about the internal development system, the agent architecture, the prompt libraries, the evaluation frameworks the team had built to quality-check agent output. And they asked about the AI components running inside the product itself. Each of those is a separate asset. Each has its own ownership question, its own protection strategy, and its own vulnerability to competitive displacement. The development team had thought carefully about none of them. This article addresses each in turn.

AI-POWERED DEVELOPMENT REQUIRES A DIFFERENT IP FRAMEWORK

The [U.S. Copyright Office has taken a clear position](#): content generated entirely by AI is not copyrightable. For AI-assisted works, where human developers substantially participate in the creative process and exercise human creative control, copyright protection may still be available for the human-authored portions (per [Copyright Office Registration Guidance](#)). Furthermore, courts have held that software elements dictated by efficiency, required by external factors, or in the public domain are filtered out of copyright protection, leaving standard boilerplate, conventional implementations, and functionally dictated patterns with little or no protection regardless of who wrote them. AI models tend to produce exactly these kinds of outputs. The more a model's suggestion looks like what every developer in your domain would write, the less protection it is likely to receive.

The risks extend beyond the copyrightability question. Major AI coding models were trained on large corpora of open source code that included GPL-licensed repositories. That code can surface in outputs, potentially importing open source license obligations your team never agreed to. The GPL requires that derivative works be released under the same terms, and those obligations do not disappear because a model was the intermediary. And because the same model serves many users, your AI-generated code may be difficult to distinguish from what a competitor's AI produced independently.

These risks are meaningfully different depending on how your team uses AI. In AI-powered development with human oversight, a developer uses AI for suggestions and reviews every output, remaining in the loop throughout. In agentic development, one or more AI agents are assigned roles and given goals, then autonomously plan, write, test, and iterate across multiple files with minimal human intervention. The human may review the result but did not participate in the thousands of micro-decisions the agents made along the way. The more autonomously agents act, the harder it is to demonstrate the human creative contribution that copyright law requires, and the more opportunity for open source license obligations to enter the codebase undetected.

For these reasons, legal departments may be tempted to advise against using AI tools in development altogether. That instinct is understandable but unlikely to hold given the pace of your competitor's innovation. Companies using these tools are compressing development cycles that used to take months into weeks. Opting out does not eliminate the risk; it transfers the competitive disadvantage to you. The better answer may lie in governance.

Across these scenarios, the issue is less whether AI was used and more whether the company can explain how design decisions were made and where human judgment actually shaped the system. Governance, in this context, is about preserving enough traceability to answer those questions later.

AI-POWERED DEVELOPMENT: WHAT GOOD GOVERNANCE LOOKS LIKE IN PRACTICE

Because AI-generated code is not itself copyrightable under U.S. law, the value of a codebase built using AI-powered development depends on what surrounds it: documented human creative involvement where it genuinely exists, and a clean provenance record that keeps problematic AI-generated open source code out of the repository. Together, these practices are designed to build the kind of explainable asset record that is better positioned to hold up in a deal, a dispute, or a registration.

- **Design before you prompt.** The most protectable expression in software often lives not in individual lines of code but in the non-literal elements, the overall structure, the sequence of operations, the organization of components, where original human judgment is most visible. Write design documentation that reflects your team's architectural thinking before reaching for an AI tool. Make the substantive decisions yourself and use the model to implement a plan that is already yours. Document the options you considered and rejected. For agentic development, this sequencing is even more important: provide detailed constraints and specifications rather than open-ended goals. An agent told "build an authentication system using our custom token schema with these specific security trade-offs" reflects more human creative direction than one told "add authentication," and that specificity is part of your authorship record.
- **Focus human creative energy on protectable elements.** Concentrate original design work on structures, abstractions, and control flow that reflect genuine design reasoning, such as a domain-specific plugin interface, a custom configuration schema, or a non-obvious orchestration approach tailored to your business logic.
- **Add a similarity and provenance gate.** Enable duplication filtering in your AI coding assistant and add an automated similarity check to your code review process before changes are merged. Maintain logs of the prompts your team uses, and for agentic development, the goals and constraints provided to agents, along with the edits made to AI suggestions. For agents that fetch external documentation or code during execution, log those sources. An agent that pulls from the web mid-task and incorporates what it finds creates a provenance gap that is difficult to close after the fact.

THINK BEYOND COPYRIGHT

Copyright frames many ownership questions around software, but it is not the only IP framework teams end up relying on when AI-assisted systems become difficult to classify cleanly. In practice, teams often rely on a mix of IP frameworks, depending on what parts of the system reflect human design, what remains confidential, and what could realistically be recreated by a competent third party without access to internal materials.

Patents. The USPTO has taken the position, in its [Revised Inventorship Guidance for AI-Assisted Inventions](#), that AI-assisted inventions are not categorically unpatentable, and that AI systems are treated as tools analogous to laboratory equipment or software. Patent law asks a different question than copyright: while copyright focuses on whether a human authored the expression, patent law asks whether a human conceived the underlying technical invention. Under longstanding doctrine, a natural person qualifies as an inventor if they formed a definite and permanent idea of the claimed solution; the fact that AI was used as part of that process does not alter the analysis. A developer who defines the technical problem, meaningfully directs the use of AI toward a particular solution, and evaluates and refines the output such that they can describe the invention with particularity may satisfy the conception requirement, though how courts will apply this standard in practice remains to be seen. Developers who believe their AI-assisted development work may involve a novel technical solution should consult patent counsel.

Trade secrets. Unlike copyright, trade secret law does not require human authorship, making it a potentially valuable alternative. It requires confidentiality and reasonable steps to maintain secrecy. However, to the extent that AI-generated code reflects material that was part of the model's training data, it may already be known or readily ascertainable and may not qualify. In addition, trade secret protection does not prevent independent development, which means its practical value often depends on whether a competitor could reasonably reconstruct the system without access to internal materials.

Contracts. Use contracts as your most reliable layer. Contractual protection operates regardless of how the statutory copyright questions resolve. Ensure that IP assignment provisions in employment and contractor agreements are drafted to capture all IP, industrial, and proprietary rights in work product incorporating AI-generated material. In addition, carefully evaluate whether the vendor agreements governing the AI tools your team uses include IP assignment provisions with respect to AI-generated output, including AI-generated code. The IP protections vary significantly across vendor platforms and tiers, and those differences matter. This article addresses vendor agreement considerations in more detail below.

A note on jurisdiction. The analysis above reflects U.S. law, but if your team includes offshore developers, your vendor agreements are governed by foreign law, or you plan to enforce rights in other markets, the picture changes significantly. The UK, for example, has expressly provided copyright protection for computer-generated works under Section 9(3) of the Copyright, Designs and Patents Act 1988, though the UK Government [proposed in March 2026](#) that this protection should be removed, subject to further monitoring of its use and impact. The EU has no direct legislation addressing ownership of AI-generated works, leaving outputs in uncertain territory across member states. The European Parliament adopted [non-binding recommendations in March 2026](#) calling for a permanent legislative solution, including applying EU copyright law to all AI systems made available within the bloc, though no legislation has yet been enacted. Italy, meanwhile, became the first EU

member state to [pass a law](#) explicitly addressing authorship of AI-assisted works, providing protection where sufficient human intellectual contribution is present.

MORE THAN JUST THE CODE

In many software companies, source code has traditionally been treated as the primary IP asset. As AI-powered systems become more complex and interconnected, that framing is increasingly incomplete. The assets that matter in practice now extend beyond code and into the surrounding systems that shape how the product behaves, improves, and differentiates over time.

Commercial software products increasingly embed AI components as part of their core operation, not as incidental features at the edges of the system. They are central to what the product does and why customers buy it. A product may include a third-party model accessed via API, a fine-tuned model the enterprise has developed and owns, an agentic runtime component operating within the customer's environment, or some combination of all three. Each of these is a distinct technology asset with its own ownership profile, its own competitive value, and its own vulnerability to competitive displacement.

The ownership question for each of these assets is distinct. A third-party model accessed via API is owned by the vendor; the enterprise's protectable interest lies in the integration architecture, system prompts, and orchestration logic built around it. Copyright may be available for the structure and expression of system prompts where they reflect genuine creative authorship, and for the human-authored elements of the integration and orchestration layer. Trade secret protection applies to the broader integration architecture to the extent it is kept confidential. A fine-tuned model developed on proprietary data is a stronger ownership story: the training data, fine-tuning methodology, and resulting model weights are potentially protectable as trade secrets, and the training dataset may carry copyright protection for human-authored selections, arrangements, or annotations within it. The model weights themselves are not copyrightable, as they are mathematical values rather than expressive works.

An agentic runtime component operating in the customer's environment raises the hardest questions. Its logic and any embedded model weights are physically accessible to the customer, making technical protection measures and careful decisions about what to expose versus retain server-side particularly important. The human-authored portions of the runtime logic may be protectable under copyright, subject to the same limitations that apply to AI-generated code generally, and the same governance practices apply: design documentation, prompt logging, and similarity checks help establish the human creative contribution.

The asset inventory has expanded on the development side as well. The internal systems enterprises build to support AI-powered development, including agent architectures, prompt libraries, orchestration logic, and evaluation frameworks, are themselves valuable

and protectable. A proprietary agentic development workflow refined through months of team experience can be genuinely difficult for a competitor to replicate. So can the telemetry and feedback data generated as the product operates in the field, the training datasets assembled to fine-tune models, and the model weights that result from that process. None of these appear in a traditional IP audit. All of them can be material to a buyer, a partner, or a competitor.

On the development side, the agent architectures, prompt libraries, and evaluation frameworks your team has built are not owned by the AI vendor and do not leave your environment. These are among the stronger copyright candidates among the newer asset categories, since they are built by human developers and reflect genuine design reasoning. Human-authored structures, abstractions, and non-literal elements may be protectable under copyright, while AI-generated portions require the same governance practices to establish the human creative contribution. Trade secret protection is also a natural fit here. The secrecy requirement is achievable through standard access controls and confidentiality agreements, and unlike copyright, trade secret law does not require human authorship.

The practical implication is that enterprises need to inventory their technology assets more broadly than they may be accustomed to, recognizing that the newer AI-related assets, from fine-tuned models and model weights to agentic runtime components and telemetry data, may represent as much competitive advantage as the source code itself, and in some cases more. The threshold question, what do we actually have, is one that many enterprises have not yet asked systematically.

EVALUATE YOUR VENDOR AGREEMENTS — THE DIFFERENCES ARE MATERIAL

Most teams make tool decisions based on capability and price without realizing that the IP protections vary significantly across vendors and tiers.

[Anthropic's consumer Terms of Service](#) assign to users "all of our right, title, and interest, if any, in Outputs." The "if any" qualifier is deliberate: it acknowledges that some AI outputs may carry no rights to assign because the underlying copyright may not exist. For commercial and API users the language is more explicit: the customer owns all outputs, Anthropic disclaims any rights it receives to customer content, and Anthropic will defend customers against third-party IP claims. Anthropic also commits that it will not train models on customer content from commercial services.

[GitHub Copilot's terms](#) state that customers retain ownership of their code and responsibility for suggestions they include in it. For Business and Enterprise customers, use of GitHub Copilot is now governed by the [GitHub Generative AI Services Terms](#), which replaced the prior Copilot Product Specific Terms for new subscriptions and renewals as of March 5, 2026, while preserving the prior commitments. IP indemnification is available to Business and Enterprise customers for unmodified suggestions, but only when Copilot's duplication detection filter is set to block suggestions matching public code ([per GitHub's](#)

[Plans and Pricing FAQ](#)). That condition matters. Both platforms contractually disclaim ownership and direct it to the user. Neither can guarantee the underlying copyright exists, as that is a question of law, not contract. What the indemnification commitments on paid tiers provide is protection against third-party infringement claims, which is the more practically valuable commercial protection in most scenarios. Look for explicit output ownership language, training data opt-outs, indemnification scope and conditions, and the gap between consumer and enterprise tiers. Many teams default to lower-cost licenses without realizing the IP protection difference. If your team is building proprietary software using agents, the enterprise tier is often worth it, not just for the indemnification, but for the data handling commitments that protect your agent configurations and prompt libraries.

BUILDING A POLICY YOUR TEAM WILL ACTUALLY FOLLOW

A clear policy governing AI tool use is worth having. The harder question is how to roll it out without creating friction that causes it to be quietly ignored.

Making governance work in practice:

- **Tier your controls by risk.** Not all code carries the same commercial and IP stakes. Apply stricter controls, including similarity checks, prompt logging, and design documentation, to code going into your core proprietary product. Internal tools and throwaway scripts warrant a lighter touch. Matching the governance burden to the actual risk keeps teams from feeling over-managed. The same tiering logic applies to the broader asset categories discussed above: agentic runtime components embedded in products and proprietary development workflows warrant the same deliberate treatment as core source code.
- **Embed controls into existing workflows.** Controls that require developers to adopt entirely new habits are often circumvented. Work with your engineering leads to integrate similarity checks and prompt logging into the steps developers already follow when submitting code for review.
- **Frame logging and documentation as authorship protection, not surveillance.** Developers are more likely to maintain prompt logs and design notes if they understand that this record protects their creative contribution and the company's IP. The framing matters as much as the policy itself.
- **Maintain an allowlist of approved licenses.** This applies to both open source code and to the AI tools and models your team uses. For open source, verify that AI-generated code does not import obligations beyond approved terms such as MIT, BSD, and Apache 2.0 before merging. For AI tools and models, confirm that vendor agreements permit your intended use and include appropriate IP protections.

- **Establish core controls with counsel.** Work with counsel to create an initial policy that establishes the core controls across code, development workflows, and AI components. Build in a review cadence, as the legal landscape and your team's use of AI-powered development will both continue to evolve.

THE TAKEAWAY

Enterprises building with AI are managing a broader and more complex IP asset portfolio than most have yet recognized, one that extends well beyond source code to include AI components embedded in products, agentic development systems, model weights, training datasets, and telemetry. The practices above are not a compliance burden. They are the record of human creative involvement that copyright law requires, the confidentiality infrastructure that trade secret protection depends on, and the contractual foundation that operates regardless of how the statutory questions resolve. Patents may also warrant attention where enterprises invest substantive human effort in the ideation and conception phase before deploying AI for development, as patent law focuses on human conception of the underlying technical idea rather than the means of implementation, though how courts will apply this standard to AI-powered development in practice remains to be seen. The asset landscape will continue to evolve. The governance infrastructure to protect it should be in place before you need it.

Sources

- U.S. Copyright Office, Copyrightability Report (2025) — copyright.gov
- U.S. Copyright Office, Registration Guidance for Works Containing AI-Generated Material (March 16, 2023) — copyright.gov
- USPTO, Revised Inventorship Guidance for AI-Assisted Inventions, 90 FR, Docket No. PTO-P-2025-0014 (Nov. 28, 2025), rescinding prior Inventorship Guidance for AI-Assisted Inventions, 89 FR 10043 (Feb. 13, 2024) — federalregister.gov
- Doe v. GitHub, Inc., No. 22-cv-06823 (N.D. Cal.), on appeal, No. 24-7700 (9th Cir.), oral argument Feb. 11, 2026
- Thaler v. Vidal, 43 F.4th 1207 (Fed. Cir. 2022), cert. denied, 143 S. Ct. 1783 (2023).
- UK Government, Report on Copyright and Artificial Intelligence (March 18, 2026), Department for Science, Innovation and Technology and Department for Culture, Media and Sport — gov.uk/government/publications/report-and-impact-assessment-on-copyright-and-artificial-intelligence
- European Parliament, Committee on Legal Affairs, Report on Copyright and Generative Artificial Intelligence, A10-0019/2026 (adopted March 10, 2026) — europarl.europa.eu
- Italy, Law No. 132 of September 23, 2025 (Provisions and Delegations to the Government Regarding Artificial Intelligence), Article 25, Official Gazette No. 223 of September 25, 2025, entered into force October 10, 2025 — normattiva.it
- Anthropic, Consumer Terms of Service (effective October 8, 2025, updated January 31, 2026), Section 5.2, last visited April 26, 2026 — anthropic.com/legal/consumer-terms

- Anthropic, Commercial Terms of Service, last visited April 26, 2026 — anthropic.com/legal/commercial-terms
- GitHub, Terms for Additional Products and Features (Copilot section), last visited April 26, 2026 — docs.github.com
- GitHub, Generative AI Services Terms (effective March 5, 2026), last visited April 26, 2026 — gh.io/terms
- GitHub, Copilot Plans and Pricing, FAQ — Responsible AI: "Can GitHub Copilot users simply use suggestions without concern?", last visited April 26, 2026 — github.com/features/copilot/plans

Notice: This content is provided for general informational purposes only and does not establish an attorney-client relationship or constitute legal advice. It may not be complete, accurate, or current, and you should seek guidance from a qualified attorney before taking any action. This content may have been organized with the assistance of artificial intelligence. The content may change without notice and could be considered Attorney Advertising in certain jurisdictions.