Professional Perspective

Best Practices Guide on Using Generative Coding

David J. Kappos, Sasha Rosenthal-Larrea, Leslie Liu and Daniel M. Barabander, Cravath, Swaine & Moore

Bloomberg Law

Best Practices Guide on Using Generative Coding

Contributed by David J. Kappos, Sasha Rosenthal-Larrea, Leslie Liu and Daniel M. Barabander

Cravath, Swaine & Moore

ChatGPT can complete your sentences, emails, and travel plans. It can also help programmers complete their code. "Generative coding" refers to artificial intelligence (AI) applications that suggest executable code to programmers based on programmer commands.

Generative coding is a subset of generative AI—which uses techniques such as neural networks and deep learning to identify patterns and create new content, including text, images and code. Two of the most popular examples of generative coding are ChatGPT and GitHub's Copilot. Programmers are using these applications to update, or even generate, their code and improve efficiency.

However, in the same way that ChatGPT is raising plagiarism issues, generative coding produces copyright risks. The use of open source software (OSS) typically requires attributions and copyright notices. In some instances, it may also impose licensing requirements on the user. Further, because generative AI is trained on OSS, it could produce output that highly resembles—or may even be identical to—that of open source code used to train the AI model.

The legal uncertainty in the field of generative coding—and related open-source licensing issues—will likely take years to resolve. In the meantime, what should companies do to ensure productivity and competitiveness while protecting themselves as they enter a legal quagmire? This article recommends concrete steps that companies can take when using generative coding applications.

Legal Risks & Mitigation Strategies

Legal questions regarding use of generative coding are emerging—three are most prominent. First, output of generated code may violate OSS licenses, providing the owner of the open source code grounds for copyright infringement and contractual claims. Second, a user of generative code generated by a model that has trained on copyrighted code can be subject to liability, on the basis that training with that code constituted infringement. Third, generated code is likely not copyrightable.

Despite the shifting legal landscape, companies will be hard pressed not to use generative coding. Instead, what companies can do is to take steps to mitigate the risks of copyright infringement or other legal claims. Below, we outline the three main legal risks, with each one followed by specific mitigation strategies.

Copyright Infringement of Code

Certain generative coding models ingest data from open source platforms. But the output code provides no attribution or tracing. The user thus has no idea what inputs were used by the model that generated the code, or whether the output code infringes the copyright of another party. If the generated code does infringe on existing code, the user risks being liable for an extended timeframe, either on copyright infringement or contractual grounds. Even if a generative coding model did not ingest infringed code, there is still a copyright claim if the model happens to generate code identical or sufficiently similar to existing code owned by a copyright holder.

Recent developments are raising the stakes for infringement claims. For instance, the copyright discovery rule—which will be litigated before the Supreme Court in *Warner Chappell Music, et al. v. Nealy*, No. 22-1078 (Supr. Ct., Sept. 29, 2023)—could make companies liable for copyright infringement damages forever. In *Warner-Chappell,* the Supreme Court will resolve a circuit split over whether to allow open-ended copyright liability. Specifically, it will consider whether a copyright plaintiff can recover damages for acts that allegedly occurred more than three years before the filing of a lawsuit. If the court agrees with the Eleventh Circuit and the Ninth Circuit and allows damages beyond the statute of limitations, the copyright owner could have the right to bring claims of infringement and seek damages forever. This can mean that a company using AI-generated software can never be confident it will not be sued for infringement involving generated code it adopted even long ago in time.

Also, courts are increasingly likely to recognize OSS licensing claims on both copyright and contractual grounds. OSS-license compliance claims are rooted in contract and not copyright, and thus may be actionable as breach of contract even if applicable AI-generated code is found non-infringing under the fair use doctrine. In *Artifex Software, Inc. v. Hancom, Inc., Docket No. 3:16-cv-06982 (N.D. Cal. Dec 05, 2016)*, the US District Court for the Northern District of California found in 2017 that the terms of open source software licenses can be enforced as both a breach of contract and copyright infringement, making clear that a copyright infringement claim does not preempt a contractual claim.

To mitigate legal risk, companies should first check whether generated code matches any existing copyrighted code owned by a third party. AI models or the generative coding model itself can be used to detect if the generated code infringes existing code. For instance, GitHub's "Block" feature detects duplicates of code from an OSS library. When this feature is enabled, GitHub Copilot checks code suggestions along with the surrounding 150 characters against public code on GitHub. Copilot will not show the suggestion if there is a match or near match. Applications such as Black Duck can check whether code in a program is a duplicate of existing code by scanning for all open source components in a codebase.

Further, companies should require tracking of generative code usage through documentation and categorization of suggested code. Some generative coding software already provides such tracking tools. For a company's internal records, every interaction with a generative coding tool should be systematically documented. Documentation includes identifying the user who initiated each task, the date of interaction, the query requested and the code returned by the software.

In addition to documentation, the company should explore ways to meaningfully organize information about usage of these tools. Categorization can help by recording how much code is used and for what purpose. At a usage level, a company could categorize whether its employees use generative coding to complete, update or generate code. At a project level, a company could track whether specific types of projects see increased usage of generative coding tools. By understanding how employees use generative coding, categorization will provide clarity on whether generative coding is used to boost efficiency or as a substitute for human creativity.

Concerted efforts to track generative coding usage during an uncertain time in the law is not only helpful to a responsive narrative if questions are later raised regarding copyright or contractual violations, it will also make rectification possible. Without proper tracing methods, product development using generated code becomes an OSS house of cards. As code get integrated into a software solution, more code is subsequently built around it. Dependencies increase and become harder to track—memories fade as to why the code was written the way it was, how a specific portion fits in, or what could be broken if it is removed.

In short, without adequate documentation, it becomes hard to remove or replace the code if that later becomes necessary in light of a copyright or contract claim. Heartache on the back end can be avoided by taking care on the front end.

Users Can Be Sued for Using Generative Coding Models

Depending on the resolution of open legal questions, it is even possible that users may be exposing themselves to liability for infringement simply for using generative coding models, even if the resulting code does not infringe existing code. In November 2022, a coding lawyer and a plaintiff law firm brought a class action suit against Microsoft, GitHub, and OpenAI. The plaintiffs allege that OpenAI's Codex and GitHub's Copilot caused copyright infringement through their use of OSS code, in that the generative coding models themselves are copying code from OSS projects without complying with their licenses.

Specifically, in *Doe 1 et al. v. GitHub Inc.*, 4:22-cv-06823 (N.D. Cal.), the plaintiffs allege that Copilot violates the plaintiffs' licenses because 1) Copilot fails to give attribution to the author of the open-source code when Copilot outputs that same code for a given user, 2) Copilot fails to include a copyright notice, and 3) Copilot fails to provide a copy of the license terms to the Copilot user. On May 11, 2023, the court dismissed all but two counts and allowed the case to proceed. The court denied defendants' efforts to dismiss two of the plaintiffs' most significant claims: the breach of open-source licenses and the removal of copyright management information under Section 1202(b) of the Digital Millennium Copyright Act.

In the GitHub lawsuit—and other ongoing litigation in the AI space this year—the focus has been on the providers of generative AI software. But it could eventually shift from the providers to the users. Providers of generative AI are claiming that "fair use" allows their training systems to use licensed content. Copyright grants rights to control reproductions, distributions of copies, making of derivative works, and public performances and displays. Fair use of copyrighted content is not infringement. Whether a usage is fair use largely depends on the nature of the use and whether it is "transformative."

Without proper precautions, a company's AI-generated work-product could qualify as a derivative work, and subject the company to infringement liability.

To mitigate potential liability, companies should conduct their due diligence regarding generative coding applications and consider tailoring standards for different applications. Users should interrogate what training data the provider of generative coding software used and the terms and conditions of usage for that training data, in particular the license to the data. If the license of an OSS library cannot be determined—which would be the default for generative coding as of now—it may not be acceptable for the company employee to use that OSS library for a given codebase. A company could also require stricter standards for the use of a generative coding software known to have trained on proprietary data, as opposed to freely usable data. For sensitive projects, it could prohibit use of the generative coding software altogether. It could also require multiple levels of tracing, such as watermarking tools, to check whether generated code is copyrighted.

Further, companies should examine each provider's terms and conditions for use of the generative coding program. Some terms now contain indemnification for third party copyright infringement claims. Copilot's terms, for instance, defends users against third-party infringement claims, subject to the user enabling its block feature for identical code. While such terms might not provide a shield against litigation, they are nevertheless helpful in support of compliance, and of course would provide a backstop against liability. If such terms are not present, companies could consider negotiating for indemnification with the provider of the generative AI solution.

Generated Code May Not Be Copyrightable

Users of generative coding may not be eligible for copyright protection for generated code, based on the argument that the code is not produced by humans. While it does not have the force of law, the US Copyright Office signaled its view in early 2023 that content generated via AI is not copyrightable. If little or no human creative effort is put into code generated by generative coding, the resulting code may not be protected under copyright.

For example, Kristina Kashtonova wrote the text of the comic book, "Zarya of the Dawn," but used a generative AI program to create its images. On Feb. 21, 2023, the US Copyright Office took the position that while the text and "selection, coordination, and arrangement of the [book's] written and visual elements" are copyrightable, the images themselves are not, as they were "not the product of human authorship." U.S. Copyright Off., Letter re Zarya of the Dawn (Registration # VAu001480196), Feb. 21, 2023.

If a company would like to produce a product that is copyrightable, it should avoid using generative coding models to the extent possible. This is where tracking usage of generative code can help, as it avoids surprises and enables companies to be deliberate and purposeful in selecting generative code versus programmer-developed code. Differentiating between generative AI applications would also be helpful, as different copyright terms for generated code in providers' user agreements should guide a company's use of the application.

Copilot's Customer Terms note, for instance, that the user, not GitHub, retains ownership of code generated by Copilot. Similarly, OpenAI assigns to the user "all its right, title and interest in and to" output provided by its generative AI models. Stable Diffusion, a generative image model, on the other hand, only gives users the right to "use" its products, stating that the provider owns outputs generated by the model and that this ownership is protected by "U.S. and international copyright laws."

Conclusion

As generative coding continues to develop, companies should be proactive in mitigating legal risk. Rather than attempting to entirely avoid use of generative AI, users should implement best practices early in the process. Importantly, companies should have employee training in place so that employees and management are well-versed in risk mitigation.

Ongoing training—including in handbooks and presentations—for employee awareness on best practices surrounding generative coding helps the company minimize risk while taking advantage of new technology using AI to improve coding efficiency and quality. Training and communication can help employees recognize the importance of provenance documentation and incent compliance. To ensure that policies are effective, companies should periodically adjust them based on legal developments, and provide avenues for employee feedback.

Cravath, Swaine & Moore LLP