Copyright in code developed using generative AI

April 11th in 2024
Masaki Miura
Global Fujitsu Distiguished Engineer
Senior Director
Software Strategy Division
Software Open Innovation Business Unit
FUJITSU LIMITED

This document contains interpretations of Japanese copyright laws, but is not an official opinion of Fujitsu. Personal interpretation and may not be accurate.
Copyright in code developed using generative AI

How do your organizations handle copyright risks?

- Copyright infringement
  - Enough measures against the risk of copyright infringement by using generative AI?
    - Conditions for incorporating code developed with help by generative AI into OSS/product
    - How to minimize the risk of copyright infringement by code developed with help by generative AI
    - Effectiveness of marking to later identify "code with help by AI"

- Copyright claim
  - Criteria for claiming copyright to code developed with help by generative AI
Risks and mitigation measures

- Risks other than copyright issues can be sufficiently mitigated by existing methods

<table>
<thead>
<tr>
<th>No.</th>
<th>Risk</th>
<th>Risk mitigation measures</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Accuracy</td>
<td>Developer's own understanding of the code, review by reviewer, and testing of the code</td>
</tr>
<tr>
<td>2</td>
<td>Vulnerability Contamination</td>
<td>Developer's own understanding of the code, review by reviewer, testing, and applying SCA</td>
</tr>
<tr>
<td>3</td>
<td>Fairness (Ethics)</td>
<td>Developer's own understanding of the code, review by reviewer</td>
</tr>
<tr>
<td>4</td>
<td>License Infringement</td>
<td>Developer's own understanding of the code, review by reviewer, applying SCA, use of a function to suppress code generation if it matches existing code (such as GitHub Copilot)</td>
</tr>
</tbody>
</table>
| 5   | Copyright Infringement      | Developer's own understanding of the code  
Review by reviewer  
Developer's determination of the uniqueness of the generated code  
Input detailed specifications to generative AI  
Modify the generated code to suit your needs |
| 6   | Securing Copyright          | Input detailed specifications to generative AI  
Modify the generated code to suit your needs |
## Relationship between usage patterns of generative AI and risk of copyright infringement

- There are risks, large and small, but not zero unless there is no code generation or very small amounts of code.
- Developers should understand, check, and modify the code.

<table>
<thead>
<tr>
<th>Amount of Output Code</th>
<th>Amount of Input Information (Code and/or documents)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code generation without original specifications or code (large amount)</td>
<td>Code completion (without any related information (Few-shot, RAG, GitHub Copilot tabs, etc.))</td>
</tr>
<tr>
<td>Code generation with a small amount of instructions</td>
<td>Code completion (with related information (Few-shot, RAG, GitHub Copilot tabs, etc.))</td>
</tr>
<tr>
<td>Code generation not based on proprietary specifications/code (small amount)</td>
<td>No code generation / no creativity</td>
</tr>
</tbody>
</table>

- **Requires special attention**
  - Code generation based on original specifications and code
    - Code generation based on detailed design
    - Test code generation
    - Refactoring
    - Bug fix
    - Code completion (with related information (Few-shot, RAG, GitHub Copilot tabs, etc.))
- **No code generation / no creativity**
  - Generation of code less than approx. 10 lines
    - It is generally accepted that creativity is not permitted for very small size of code (*The number of lines needs to be investigated and examined.*)
  - Program description, inserting comments, finding bugs
    - No generation or modification of code

- The possibility of copyrighted code contamination is very low, as original specifications or code are reflected.
- It is very unlikely to contain copyrighted code because the output of generative AI reflects proprietary code and its specifications.
- It is unlikely to contain copyrighted code because generated code reflects proprietary code or its specifications.

From the viewpoint of originality, it is unlikely to contain copyrighted code, but caution is required.
Proposed method for making it as clear as possible that there is no similarity of code to existing code

1. Only developers approved by the project manager are allowed to use generative AI
   - Developers should be able to understand whether code is highly unique and the logic of the code.
   - But the conditions need not be too strict.

2. Establish guidelines for the use of generative AI to code generation
   1. Ensure that the generated code does not have any obvious unique features (Looks like obfuscated code, includes proper nouns in code, etc.).
      1. If code is considered highly unique, it should be regenerated, coded by developer based on the logic as a reference, or modified to meet development specifications.
      2. Keep verified evidence in review records, commitment logs, etc.
   2. A comment “With help by AI” should be added to functions or files based on code generated by AI, to ensure that the use of generative AI can be identified in the future,
   3. When reviewing code, pay special attention to ensure that code is free of copyright infringement.
   4. Apply SCA and clear the indications. (Some of risk reduction can be automated, though not perfect)
Relatively safe or risky use

● Premise for use
  ● Do not use generative AI for coding for design or technology that you do not understand
    ● When logic is specified in detail by humans, creativity occurs. Therefore, the generated code is highly unlikely to constitute copyright infringement.
    ● Project managers should not allow developers with limited development experience to use generative AI, to develop OSS and product code. (excluding research and learning purposes)

● Usage considered relatively safe
  ● A developer inputs a detailed design into a generative AI to generate code. (There is creativity in the design)
  ● Code generation based on in-house/original code (Unit test generation, refactoring, and bug fixes for existing in-house code)
  ● Using generative AI just to know how to implement something in general

● Usage considered high risk
  ● Developers ingest generated code without understanding its content and logic. (Not allowed also from the viewpoint of software quality)
Patterns of relatively safe or risky codes

- Output code patterns considered relatively safe
  - Few lines of code, uncreative (No matter who writes it, it's the same code.).
  - The data being processed is proprietary, such as internal documents, and the code depends on it.
  - Library- and API-call-driven code (* This is the case for high-level languages)

- Output code patterns that may need to be fully checked
  - Only the outline of the processing is given to the generative AI to output a large amount of code, and there are no API or library function calls.
  - Unique looking code (Uncommon variable names such as proper nouns, including company names, one line is noticeably long, the whole is condensed, etc.)
Thank you