SBOM Challenges for Developers: From Analysis of Stack Overflow Questions

Wataru Otoda*, Tetsuya Kanda†, Yuki Manabe‡, Katsuro Inoue§, Yoshiki Higo*
* Osaka University, Japan, {wa-otoda, higo}@ist.osaka-u.ac.jp
† Notre Dame Seishin University, Japan, kanda@m.ndsu.ac.jp
‡ The University of Fukuchiyama, Japan, manabe-yuki@fukuchiyama.ac.jp
§ Nanzan University, Japan, inoue599@nanzan-u.ac.jp

***Abstract*—Current software development takes advantage of many external libraries, but it entails security and copyright risks. While the use of the Software Bill of Materials (SBOM) has been encouraged to cope with this problem, its adoption is still insufficient. In this research, we analyzed the challenges that developers faced in practicing SBOM use by examining questions about SBOM utilization on Stack Overflow, a Q&A site for developers. As a result, we found that (1) the proportion of resolved questions about SBOM use is 15.0% which is extremely low, (2) the number of new questions has increased steadily from 2020 to 2023, and (3) SBOM users have three major challenges on SBOM tools.**

***Index Terms*—SBOM, SPDX, CycloneDX, Software Supply Chain, Stack Overflow**

I. Introduction

Modern software development is often accomplished not by implementing all functionalities from scratch but by leveraging numerous external libraries. This approach not only reduces development costs and shortens development cycles but also allows for the use of mature software components, leading to robust and sophisticated software [1], [2]. However, such activity comes with risks concerning cyber security and copyright [3]–[5]. Indeed, in 2020, a cyber incident occurred where malware was embedded in the software update of SolarWinds' Orion software through a supply chain attack, leading to cyberattacks on numerous U.S. government agencies and private sectors that used Orion software [6]. Despite identifying and addressing these risks early on is crucial, the immense number of external libraries used in current software development, especially considering transitive dependencies like dependencies of dependencies [7], makes proper management challenging.

To address this issue, the use of the Software Bill of Materials (SBOM) is encouraged. SBOM is a formal and machine-readable record of software components (e.g. libraries) used in building software, which includes details such as licenses, versions, and vendors of each software component, as well as information on supply chain relationships between components [8], [9]. CycloneDX and SPDX are the primary formats for representing SBOM [10]. While SBOM is rapidly getting attention in recent years with government agencies promoting its use [11], [12], its adoption is still insufficient [13], [14]. While there are studies identifying causes and proposing solutions through surveys conducted on companies and organizations, no study has analyzed the challenges faced by ordinal developers in utilizing SBOM in practice. Therefore, this study analyzes questions on the use of SBOM on Stack Overflow, a primary question-and-answer (Q&A) site for developers, where we deemed developers facing challenges in utilizing SBOM likely to seek solutions on by asking questions, considering its popularity [15].

The remainder of this paper is organized as follows. Section II describes related works. Section III explains the research methodology. Section IV reports the results and discussion. Finally, section V concludes the research and discusses future works.

II. Related Work

The Linux Foundation conducted an SBOM readiness survey in 2022 targeting 412 organizations of various industry types and sizes [9]. While 76% of organizations work on SBOM adoption, the survey revealed concerns such as the industry's honest commitment to SBOM adoption, the lack of industry consensus on what an SBOM should contain, and the value to their customers by providing them with an SBOM being unclear. Xia et al. conducted an interview on the adoption status and challenges of SBOM among developers, highlighting the importance of incentives for SBOM generation, industry consensus on what to include in SBOM, mechanisms for selective sharing of SBOM content, SBOM content validation/verification, mature SBOM tools, and increasing awareness of SBOM [13]. Stalnaker et al. investigated the challenges stakeholders face during SBOM creation and usage, emphasizing the importance of multi-dimensional SBOM specifications, enhanced SBOM tooling and build system support, SBOM content validation, and incentives for SBOM adoption [16].

While these studies provide insights into the current status and future challenges of SBOM adoption in the industry, they do not target ordinal software developers outside of companies or organizations. In contrast, Nocera et al. analyzed the SBOM adoption on open-source software (OSS) repositories on GitHub, the most popular version control platform for both personal and professional use [14]. The results indicated a low adoption of SBOM in OSS repositories, with few repositories meeting the guidelines set by the National Telecommunications and Information Administration (NTIA) for how SBOM

Fig. 1. A question found on Stack Overflow.

should be supplied, highlighting the importance of convenient SBOM generation tools that can be easily integrated into Continuous Integration and Continuous Delivery (CI/CD) pipelines and build-automation tools. However, this study provides quantitative insights and does not reveal the concrete challenges software developers face.

III. Analysis

In this study, we analyzed questions related to SBOM uses on Stack Overflow, focusing on the CycloneDX and SPDX formats. Stack Overflow is a developer Q&A site where users can post and search for questions and answers. Fig. 1 provides an example of a question on Stack Overflow[1]. A post on Stack Overflow typically consists of a title, question, multiple tags, and multiple answers to the question. Among answers, the one deemed most helpful by the question asker in resolving the issue is marked with a green checkmark indicating the *Accepted* status. However, no answers may get accepted in such a case as all of them fail to resolve the question.

We followed the process outlined in Fig. 2. First, we selected search keywords that are `SBOM`, `CycloneDX`, and `SPDX`. Then, we extracted questions that case-insensitively contained at least one of these keywords in the title or body. Although Stack Overflow allows users to set tags based on the technical domains of their questions, SBOM-related tags were not widely used at the time of the analysis. We then filtered out non-SBOM-related questions (referred to as noise) e.g. those containing a log text from an unrelated framework or program code that happens to contain an SBOM-related word in it.

For the keywords `SBOM` and `CycloneDX`, we first entered `SBOM is:question` and `CycloneDX is:question` into the search bar on the Stack Overflow site to obtain SBOM questions and then removed noise by hand. On the other hand, for the keyword `SPDX`, we used the Stack Exchange Data Dump 2023-09-12[2]. This was because the search results exceeded the limit of 500 questions imposed by Stack Overflow's

[1]https://stackoverflow.com/questions/69121175
[2]https://archive.org/details/stackexchange_20230912

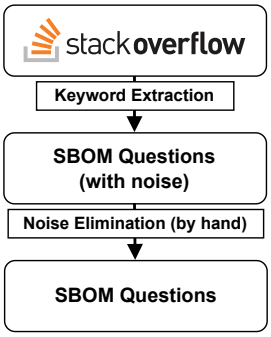

Fig. 2. Flow of this analysis.

TABLE I
Noise Pattern Strings for the Keyword *SPDX*.

Noise Pattern Strings	Count
`SPDX-License-Identifier`	1229
`License should be a valid SPDX license expression`	30
`spdy`	51
`spdx-correct`, `spdx-expression-parse`, `spdx-exceptions`, `spdx-license-ids`	62

search function. We filtered out posts that contained noise pattern strings at least once mechanically and then we removed the remaining noise by hand. Table. I shows the noise pattern strings and their frequencies in the `SPDX` keyword search.

The noise pattern strings were selected from frequent texts in noise and verified to not appear in an SPDX-formatted SBOM. The reasons why we deemed them noise pattern strings are as follows. `SPDX-License-Identifier` is a standardized notation used to declare licenses at the beginning of program code, which is specified as part of the SPDX standard [17] but the notation itself is not an SBOM. Similarly, `License should be a valid SPDX license expression` is a warning message output by the Yarn package manager when a package's license notation is incorrect. `spdy` was selected because it appeared in cases where `spdx` (with varying capitalization) was used as a variable name representing the velocity on the x-axis in programs, often accompanied by `spdy`, representing the velocity on the y-axis. `spdx-correct`, `spdx-expression-parse`, `spdx-exceptions`, and `spdx-license-ids` are package names on the NPM package manager, typically appearing in log outputs contained in questions regarding NPM errors where these packages are included in the project dependencies.

As a result of this noise elimination process, 27 questions out of 58 remained for the keyword `SBOM`, 28 out of 31 for `CycloneDX`, and 6 out of 1432 for `SPDX`, totaling 42 questions due to duplicates. From these questions, we performed the following analyses:

Analysis 1: We examined the presence of answers and the resolution status of each question. Questions with an accepted answer were considered resolved, while those without were

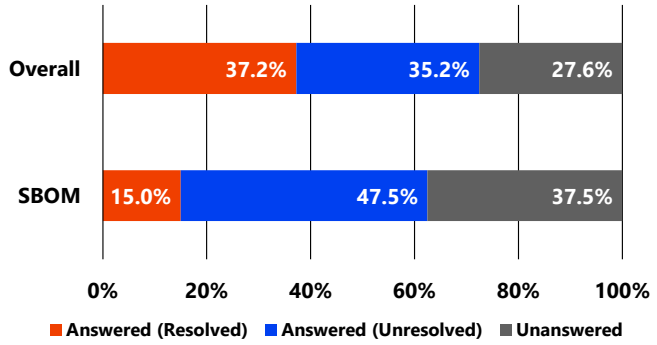

Fig. 3. Answered and Resolved Rate of SBOM Questions.

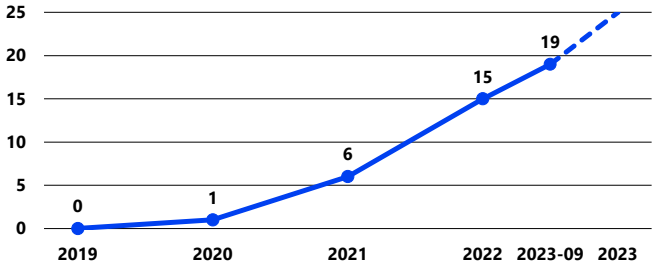

Fig. 4. Annual Trends in the Number of New SBOM Questions.

considered unresolved.

Analysis 2: We summed up the number of new questions posted each year.

Analysis 3: We extracted technical issues from the content of each question, identifying frequently asked questions and those seemingly unresolved based on the subjective judgment of the lead author.

IV. Analysis Results and Discussion

A. Analysis 1: Answered and Resolved Rate of SBOM Questions

Fig. 3 shows the proportion of *Answered (resolved)*, *Answered (unresolved),* and *Unanswered* SBOM questions, along with that of overall questions on Stack Overflow. To facilitate a fair comparison, we filtered out questions posted before 2021 when SBOM questions were not frequently asked. Among the total of 40 SBOM questions, 6 questions (15.0%) were answered (resolved), 19 questions (47.5%) were answered (unresolved), and 15 questions (37.5%) were unanswered.

Considering that the proportion of unanswered questions overall is 27.6%, SBOM questions are answered likely on average. However, the resolution rate of SBOM questions is at an extremely low level from the proportion of resolved questions in the overall being 37.2%. It suggests that it is challenging to find solutions on Stack Overflow when SBOM users face issues, indicating either a shortage of software developers knowledgeable enough to answer SBOM questions or SBOM tools being so immature that many SBOM issues are currently unsolvable at the time of questioning, as also discussed in Section IV-C3.

B. Analysis 2: Trends in the Number of SBOM Questions

The annual trends in the number of new SBOM questions from 2019 to 2023 as of September 2023 are shown in Fig. 4. Before 2019, the number of SBOM questions was 1 in 2012 and 0 for the rest.

Considering the popularity of Stack Overflow, SBOM questions extracted in this analysis remained small over the four years. However, we can observe a steady increase in the number of new SBOM questions from 2020 to 2023, with a notable acceleration from 2021 onwards, possibly correlated with the issuance of the U.S. executive order requesting SBOM use and the standardization of SPDX by ISO/IEC in 2021.

C. Analysis 3: Challenges Faced by SBOM Users

We found that the following challenges are significant:

1) Insufficient coverage of use cases by SBOM tools (7 posts)
2) Inability of SBOM tools to meet requirements (4 posts)
3) Immaturity of SBOM tools or unclear usages (19 posts)

Below are actual questions found on Stack Overflow highlighting each challenge.

1) Insufficient Coverage of Use Cases by SBOM Tools: There is a question titled *"How do I generate a Cyclonedx bom for a Java project built with Ant?"* posted on March 2022[3], which is from a software developer with experience generating an SBOM for Gradle projects but facing difficulties with an older Apache Ant project. While an answer suggests that CycloneDX Core (Java) might work, it remains unaccepted. This question highlights insufficient tool support for older project management systems. Another question titled *"Is there any tool through which we can generate SBOM report (SPDX / CycloneDX) for Windows programs?"* on September 2022[4] also highlights the absence of tools for generating SBOMs for applications installed on Microsoft Windows, despite the availability of such tools for Linux. While answers exist, none have been accepted. The lack of such tools could be attributed to the decentralized nature of software metadata management on Windows compared to Linux, making it challenging to develop SBOM generation tools for Windows.

2) Inability of SBOM tools to Meet Requirements: There is a question titled *"NTIA minimum SBOM requirement tool"* posted on April 2023[5], which illustrates a developer's attempt at generating SBOMs for Java, Python, iOS (Swift), and Android (Kotlin) projects while adhering to the NTIA's guideline. Despite trying various tools, none fulfilled all the required fields, indicating a gap between SBOM tools' ability and compliance requirements or challenges in obtaining all the necessary information as specified in the guideline from the perspective of SBOM tool developers. Another question titled *"How to include Open Source license in GitHub SBOM*

[3]https://stackoverflow.com/questions/71605182
[4]https://stackoverflow.com/questions/73648096
[5]https://stackoverflow.com/questions/76103711

export?" on August 2023[6] also addresses the absence of open source license information in GitHub's SBOM export, highlighting the limitations of GitHub's SBOM generation functionality for compliance management purposes.

3) Immaturity of SBOM Tools or Unclear Usages: There is a question titled *"How to create a BOM file in a Flutter project"* posted on June 2023[7], which details a developer's struggle to generate a CycloneDX SBOM for an Android portion of a Flutter project, encountering errors during the process. Similar questions addressing errors and unclear usages by SBOM generation tool users were prevalent. These questions indicate that SBOM generation tools are relatively immature, leading to defects, insufficient features, and a lack of comprehensive documentation or accumulated knowledge among software developers on basic usage, making the correct SBOM generation difficult.

Through this analysis, questions requiring fundamentally new approaches or systems were scarce, with many revolving around usages, issues, and minor missing functionalities on existing tools. This suggests that developers seek accumulated knowledge, refinement, and extension to existing tools.

D. Threats to Validity

We filtered out questions using the noise pattern strings listed in Table. I when analyzing questions containing the keyword `SPDX`. However, by doing so, there is a possibility that questions related to SBOM use, which should not have been excluded as noise, were excluded. However, in selecting noise pattern strings, we confirmed that posts containing those strings were likely noise by extracting multiple posts containing the string for investigation. Therefore, the threat to validity caused by this exclusion is considered small.

On another note, the number of questions extracted as meeting the criteria was only 42. While this is possibly due to the insufficient popularity of SBOM, it might affect the accuracy of the analysis results. Although there are also general Q&A sites like Quora apart from Stack Overflow, we observed no active usage of these platforms by developers or managers involved in SBOM adoption.

V. Conclusion and Future Work

In this research, we analyzed questions posted on the developer Q&A site Stack Overflow on the use of SBOM. Through analyzing the answered and resolved rate of SBOM questions (Analysis 1), we revealed that Stack Overflow is still not a satisfactory venue to ask SBOM questions to obtain solutions. The second analysis on trends in the number of SBOM questions (Analysis 2) revealed a notable increase from 2021, possibly due to the U.S. executive order mandating SBOM use and the standardization of SPDX by ISO/IEC. The last analysis on challenges faced by SBOM users (Analysis 3) identified three main challenges faced by SBOM users.

We consider it a future work to keep tracking the latest trends in SBOM use, in such a way as to continue investigating Stack Overflow which is showing an increasing trend in the number of SBOM questions, or to conduct analyses on websites other than Q&A sites such as Reddit or issues in the SBOM-related projects on GitHub.

[6]https://stackoverflow.com/questions/76962834

[7]https://stackoverflow.com/questions/76515339

Acknowledgment

This work was supported by JSPS KAKENHI Grant Numbers JP24K14895, JP21K02862, JP23K28065, JP21K18302, JP21H04877, JP22H03567, JP22K11985, and Nanzan University Pache Research Subsidy I-A-2 for the 2024 academic year.

References

[1] O. P. N. Slyngstad, A. Gupta, R. Conradi, P. Mohagheghi, H. Rønneberg, and E. Landre, "An empirical study of developers views on software reuse in Statoil ASA," in *Proc. ISESE2006*, 2006, pp. 242–251.

[2] W. C. Lim, *Managing software reuse : a comprehensive guide to strategically reengineering the organization for reusable components*. Prentice Hall, c1998.

[3] Y. Wang, B. Chen, K. Huang, B. Shi, C. Xu, X. Peng, Y. Wu, and Y. Liu, "An empirical study of usages, updates and risks of third-party libraries in Java projects," in *Proc. ICSME2020*, 2020, pp. 35–45.

[4] M. Ohm, H. Plate, A. Sykosch, and M. Meier, "Backstabber's knife collection: A review of open source software supply chain attacks," in *Proc. DIMVA2020*, 2020, pp. 23–43.

[5] J. Cho, Ed., *Understanding Open Source: Compliance and Enforcement*, ser. The SciTech Lawyer, vol. 10, no. 4, American Bar Association, Summer 2014.

[6] E. D. Wolff, K. M. Growley, M. O. Lerner, M. B. Welling, M. G. Gruden, and J. Canter, Eds., *Navigating the SolarWinds Supply Chain Attack*, ser. The Procurement Lawyer, vol. 56, no. 2, American Bar Association, Spring 2021.

[7] R. Kikas, G. Gousios, M. Dumas, and D. Pfahl, "Structure and evolution of package dependency networks," in *Proc. MSR2017*, 2017, pp. 102–112.

[8] "The minimum elements for a software bill of materials (SBOM)," The United States Department of Commerce, 2021. [Online]. Available: https://www.ntia.gov/report/2021/minimum-elements-software-bill-materials-sbom

[9] S. Hendrick and J. Zemlin. (2022, Jan) The state of software bill of materials (SBOM) and cybersecurity readiness. Linux Foundation. [Online]. Available: https://www.linuxfoundation.org/research/the-state-of-software-bill-of-materials-sbom-and-cybersecurity-readiness

[10] B. Bensing. (2022) History of the software bill of material (SBOM). [Online]. Available: https://billbensing.com/software-supply-chain/history-software-bill-of-material-sbom/

[11] J. R. Biden, Jr., "Executive order on improving the nation's cybersecurity," The White House, May 2021. [Online]. Available: https://www.whitehouse.gov/briefing-room/presidential-actions/2021/05/12/executive-order-on-improving-the-nations-cybersecurity/

[12] "Cyber resilience act," European Commission, Sep 2022. [Online]. Available: https://digital-strategy.ec.europa.eu/en/library/cyber-resilience-act

[13] B. Xia, T. Bi, Z. Xing, Q. Lu, and L. Zhu, "An empirical study on software bill of materials: Where we stand and the road ahead," in *Proc. ICSE2023*, 2023, pp. 2630–2642.

[14] S. Nocera, S. Romano, M. Penta, R. Francese, and G. Scanniello, "Software bill of materials adoption: A mining study from GitHub," in *Proc. ICSME2023*, Oct 2023, pp. 39–49.

[15] All sites. Stack Exchange Inc. Accessed Feb 2024. [Online]. Available: https://stackexchange.com/sites?view=list

[16] T. Stalnaker, N. Wintersgill, O. Chaparro, M. Di Penta, D. M. German, and D. Poshyvanyk, "Boms away! inside the minds of stakeholders: A comprehensive study of bills of materials for software systems," in *Proc. ICSE2024*, 2024.

[17] *Annex E: Using SPDX short identifiers in Source Files*, Linux Foundation and its Contributors Std., Rev. 2.3.0, 2022. [Online]. Available: https://spdx.github.io/spdx-spec/v2.3/using-SPDX-short-identifiers-in-source-files/