What Developers (Care to) Recall: An Interview Survey on Smaller Systems

Jacob Krüger[1] Regina Hebig[2]

Abstract: This extended abstract summarizes our paper with the homonymous title published at the International Conference on Software Maintenance and Evolution (ICSME) 2020 [KH20].

Keywords: Knowledge; Information needs; Developers' memory

Developers have to understand the behavior and properties of the software in their system in order to extend and maintain it, which is referred to as *program comprehension*. While studying program comprehension, researchers have conducted empirical studies aiming to analyze the readability of source code (e.g., based on different identifier names), investigated developers' information needs (e.g., what questions come to a developer's mind during their tasks), and designed techniques to support knowledge recovery (e.g., by reverse engineering information). Interestingly, researchers have rarely investigated developers' memory decay or what knowledge they consider important to remember, and thus keep in their mind. Understanding what knowledge developers memorize helps to scope tools, practices, and research. For instance, experts may require more light-weight code searching capabilities, due to their memorized knowledge. In contrast, novices or new team members may require extensive documentation or the help of experts to understand the system architecture, which may be tacit knowledge experts aim to memorize. Moreover, such needs may vary depending on the type of knowledge, for example, the system's architecture versus meta information about the team or the system's evolution.

Motivated by our previous work, we investigated the connection between developers' memory decay, types of knowledge, and what knowledge developers consider important to remember. To this end, we started with a systematic literature review of 14 papers that are concerned with a total of 456 questions developers ask during maintenance and development tasks. We analyzed the 420 questions that the authors classified into 81 classes to gain a first understanding of developers' knowledge needs. Then, we re-classified all questions to unify the classes and get a consolidated overview. Building on our insights, we derived a semi-structured interview guide, which we used to conduct 17 interviews with developers from different areas (i.e., academia, industry, open-source), domains (e.g., web services, machine learning, static code analysis), and countries (e.g., Germany, Sweden, France). We

[1] Otto-von-Guericke-University Magdeburg, Germany
jkrueger@ovgu.de

[2] Chalmers | University of Gothenburg, Sweden
regina.hebig@cse.gu.se

doi:10.18420/SE2021_24

remark that most of the systems we asked our interviewees about were comparably small (i.e., 1–6 developers, four with more than 100 k lines of code). During each interview, we started with a self-assessment of the interviewee's familiarity (i.e., remaining knowledge) with the system overall and with respect to three types of knowledge (i.e., architecture, meta, code), which we repeated after each section related to these types. We then asked our interviewee to recall answers to six questions from the systematic literature review on each of the three different types of knowledge from their memory (i.e., a total of 18 different questions). Afterwards, but before checking for correctness, we asked each interviewee to rate the importance of the three knowledge types and individual questions.

Triangulating from the results of our systematic literature review and our interview survey, our core findings are:

- Developers consider more abstract knowledge about their system (e.g., the system architecture and intentions of the code) more important to remember.
- Developers can recall knowledge for questions they consider important more often correctly than for those they consider less important.
- Developers may be reliable in doing self-assessments of their familiarity, but, interestingly, these self-assessments usually decreased after answering questions about their system from memory.

In our paper, we report various additional insights that have important implications for practice and motivate new research directions. For instance, our results support our aforementioned assumptions that developers aim to remember a system's architecture. Consequently, practitioners have to think about how to document and maintain the corresponding information explicitly, to avoid that the tacit knowledge is lost over time. Moreover, our findings provide guidance on how to structure teams or onboard new developers, while researchers may explore new techniques for reverse engineering information. As direct future work, we are working on extensions of our study to overcome its limitations (e.g., small system sizes, number of participants). For this purpose, we are planning and conducting additional empirical studies (e.g., surveys, controlled experiments) to reinforce our findings.

Bibliography

[KH20] Krüger, Jacob; Hebig, Regina: What Developers (Care to) Recall: An Interview Survey on Smaller Systems. In: International Conference on Software Maintenance and Evolution. ICSME. IEEE, pp. 46–57, 2020.