2020 IEEE International Conference on Software Maintenance and Evolution (ICSME)

On the need for automatic knowledge management in modern collaboration tools to improve software maintenance

Vipin Balachandran
VMware
Palo Alto, USA
vbala@vmware.com

***Abstract*—The adoption of modern collaboration tools such as Slack has reduced email communication and discussions in other forums in software organizations. The absence of automatic knowledge management in these tools can significantly impact software maintenance and developer productivity in the long run. In this paper, we discuss the need and characteristics of such knowledge management systems.**

The adoption of modern collaboration tools such as Slack [1] is increasing in software organizations [2], [3]. Approximately $31,500$ Slack channels were created by various teams in our company between January 2017 and July 2020. Project conversations and troubleshooting discussions that happened over email and bug tracker forums are now moving to these modern collaboration tools in our organization.

It is essential to capture a team's tribal knowledge for effective maintenance. This knowledge includes troubleshooting tips, explanations of unexpected product behaviors, workarounds or fixes to common problems, and design rationale. The discussion threads and logs in bug tracker forums also offer valuable information to maintainers. They document the root causes of unexpected behaviors and the rationale behind fixes, which are valuable resources for troubleshooting. Emails and bug tracker forums facilitate the recording and retrieval of a team's tribal knowledge through threaded conversations and effective search mechanisms.

The increased adoption of modern collaboration tools is turning out to be counter-productive for software maintenance. The developer discussions in our Slack channels contain log messages, source code references, stack traces, troubleshooting tips, and the rationale behind fixes and design decisions. Often the discussions in other forums refer to this tribal knowledge. There are $\approx$2200 bugs in one of our bug trackers that refer to discussion threads in Slack using URLs. The referred thread may not be accessible to all developers because the corresponding Slack channel may be private to a team, and hence missing the necessary context. To make matters worse, we found a significant number of bugs ($\approx$1700) that refer to Slack conversations without providing any URL. This kind of knowledge fragmentation makes software maintenance difficult.

Chat messages of development teams discuss various topics [4]. In the case of Slack channels, these messages are often interleaved with bot messages. The high volume of messages about various topics overwhelms users and leads to delayed responses and repeated queries during maintenance tasks. The retrieval of a team's tribal knowledge using the in-built search mechanisms of these tools is challenging. This difficulty is partly due to the high volume of unstructured messages covering different topics. Threaded communication helps organize topics, but it is observed that such features in these tools are sparingly used. On the other hand, the threaded conversation is natural for emails and bug tracker forums. Another reason is the large number of channels that are private to teams or groups of developers, which also exacerbates the knowledge fragmentation.

We need tight integration of automatic knowledge management systems with modern collaboration tools to organize and improve the retrieval of a team's tribal knowledge and reduce knowledge fragmentation. Such a system should:

- identify an expert capable of answering a query;
- enable the developers to filter out irrelevant messages;
- extract query-response pairs to create a knowledge base; and
- capture troubleshooting discussions, including logs and stack traces, and associate it with related bugs and commits.

There are Slack bots such as Guru [5] that facilitate knowledge management, but it requires manual intervention to label conversations. In [3], the authors describe the challenges faced by agile teams due to a large number of private channels, and suggest a systematic process for the creation and archiving of channels for the discussion of bugs and features.

References

[1] "Slack," https://slack.com/.
[2] B. Lin, A. Zagalsky, M.-A. Storey, and A. Serebrenik, "Why developers are slacking off: Understanding how software teams use slack," in *Proceedings of the 19th ACM Conference on Computer Supported Cooperative Work and Social Computing Companion*, 2016, pp. 333–336.
[3] V. Stray, N. B. Moe, and M. Noroozi, "Slack me if you can! using enterprise social networking tools in virtual agile teams," in *2019 ACM/IEEE 14th International Conference on Global Software Engineering (ICGSE)*. IEEE, 2019, pp. 111–121.
[4] R. Alkadhi, T. Lata, E. Guzmany, and B. Bruegge, "Rationale in development chat messages: an exploratory study," in *2017 IEEE/ACM 14th International Conference on Mining Software Repositories (MSR)*. IEEE, 2017, pp. 436–446.
[5] "Guru Bot for Slack," https://www.getguru.com/.