Software Bots in Software Engineering: Benefits and Challenges

Mairieli Wessel
m.wessel@tudelft.nl
Delft University of Technology
Delft, The Netherlands

Marco A. Gerosa
marco.gerosa@nau.edu
Northern Arizona University
Flagstaff, USA

Emad Shihab
emad.shihab@concordia.ca
Concordia University
Montreal, Canada

ABSTRACT

Software bots are becoming increasingly popular in software engineering (SE). In this tutorial, we define what a bot is and present several examples. We also discuss the many benefits bots provide to the SE community, including helping in development tasks (such as pull request review and integration) and onboarding newcomers to a project. Finally, we discuss the challenges related to interacting with and developing software bots.

KEYWORDS

Software Bots, GitHub Bots, Chatbots, Human-bot Interaction, Open Source Software, Automation, Collaborative Development

ACM Reference Format:
Mairieli Wessel, Marco A. Gerosa, and Emad Shihab. 2022. Software Bots in Software Engineering: Benefits and Challenges. In *19th International Conference on Mining Software Repositories (MSR '22), May 23–24, 2022, Pittsburgh, PA, USA.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3524842.3528533

1 WHAT IS A BOT?

Despite its increasing popularity, defining what a bot continues to be a challenge. The terminology used to describe software bots (i.e., bots, devbots, chatbots, chatterbots, software agents, etc.) is vast, diverse, and often inconsistent. Researchers and practitioners have defined software bots according to their specific characteristics and applications. For instance, bots are often defined as ***automation providers***, which is linked to their ability to automate tasks and act (to some extent) autonomously [19]. Bots can be also defined based on their ability to ***communicate using human language*** [2]. Although several popular bots do in fact have some language capability, engaging in conversations is not required for software bots. Erlenhov et al. [6] pose that bots are agents ***with human-like traits***: "*an artificial software developer who is autonomous, adaptive, and has technical as well as social competence.*" Recently, Lebeuf [9] explored the similarities between the multiple definitions of bots, and characterized bot as ***an interface*** "*that connects users to services.*" According to Lebeuf [9], this interface usually provides *"additional value (in the form of interaction style, automation, anthropomorphism, etc.) on top of the software service's basic capabilities.*"

MSR '22, May 23–24, 2022, Pittsburgh, PA, USA

ACM ISBN 978-1-4503-9303-4/22/05.
https://doi.org/10.1145/3524842.3528533

2 EXAMPLES OF BOTS

As an interface between human developers and other tools, bots support social and technical activities, including communication and decision making [14]. For example, researchers have designed bots for answering developers' questions on conversational platforms such as Slack and Gitter. For example, Romero et al. [12] implemented GitterAns, a bot to automatically detect when a developer asks a technical question in a Gitter chat and leverages the information from Q&A forums to provide answers. On social coding platforms, such as GitHub, open-source communities have been adopting bots to automate a variety of repetitive tasks, including repairing bugs [10], refactoring source code [19], suggesting code improvements [11], and predicting defects [8]. Bots even perform more activities than human developers in some projects [4].

3 BENEFITS

Why do developers use bots in domains such as those exemplified in the previous section? The literature poses that developers employ bots to improve productivity and the quality of the work and enable activities for which humans are not realistically suitable [7]. Automating simple, time-consuming, or tedious tasks and collecting dispersed information are some ways to improve productivity. Smarter bots can even gain new insights or find patterns that humans might overlook in the large amount of information available. Some software developers stress that bots may do some tasks better than humans [7]. For example, bots can handle tasks 24/7 and at scale, increase consistency, and mitigate human error. Some developers report that software bots are integral and indispensable part of the software development ecosystem [7].

Wessel et al. [17] also investigated the benefits for adopting bots (in this case, code review bots) and found similar results as Erlenhov et al. [7]. The main reasons for adopting bots are related to enhancing the feedback to developers, reducing maintainers' effort, enforcing high code coverage, ensuring high-quality standards, and automating routine tasks. Interestingly, the developers also report benefits related to interpersonal relationships. According to these developers, negative feedback in an automatic bot answer feels less rude or intimidating and, by providing quick and constant feedback, bots reduce the chance the author abandons a pull request.

Bots can also support stakeholders unfamiliar with the project or the SE practices and technologies. Dominic et al. [5], for example, propose the use of software bots to onboard newcomers to open source projects. In their vision, after a newcomer joins an open-source project, the bot could collect newcomers' experience and interests and recommend tasks, pull and summarize relevant information from sites as Stack Overflow, recommend mentors, prompt the newcomer, answer questions, congratulate accomplishments, etc.

Students may also benefit from software bots. For example, Brown and Parnin [3] propose a bot to nudge students toward better SE behaviors.

The adoption of new technology such as bots can change human activities. Wessel et al. [16] investigated how activity traces change after the adoption of bots and found that, after a bot adoption, projects show more merged pull requests, fewer comments, fewer rejected pull requests, and faster rejections. Developers explain that some of these effects are caused by benefits such as increased visibility of code quality metrics, immediate feedback, test automation, increased confidence in the process, change in the discussion focus, and the fact that bot feedback pushes contributors to take action.

4 CHALLENGES

Although bots provide many benefits, the literature has reported a number of challenges associated with bots in SE. These challenges can be categorized into interaction and development challenges. **Interaction** challenges refer to challenges that users of bots in SE face in interacting with the bots and **Development** challenges refer to challenges that developers of bots face.

To examine *interaction* challenges, Wessel et al. [18] interviewed 21 practitioners to understand the challenges caused by bots in pull request interactions. The authors reported a number of challenges posed by bots in SE, including expectation breakdowns (where bots might be intimidating to newcomers), communication issues (where bots lack contextualizing their actions), managing bots' configurations (i.e., limited and burdensome configuration options), among others. However, the authors note that the noise (i.e., interference produced by a bot's behavior that disrupts the communication between project maintainers and contributors) was the most prominent challenge. The authors highlight that developing ways to deal with such noise (e.g., [15]) is critical for the future success of bots in SE.

Santhanam et al. [13] performed a systematic literature study on bots in SE. The authors noted a number of challenges found in the literature that need to be overcome for bots' success in the future. Particularly, the authors note that the interfaces need to evolve (most are text based and will need to incorporate voice interfaces, for example), that bots need to improve their social interaction to improve trust and have more natural conversations, and that dialogue management frameworks will need to become more flexible to allow for better flowing and intelligent interactions. They also mention the need for bots to be able to explain decisions and interactions.

To examine *development* challenges, Abdellatif et al. [1] studied Stack Overflow posts related to the development of chatbots. They identified 12 unique topics that developers face when developing bots, such as user interaction, model training, development frameworks, NLU integration, messenger integration, and intent and entity recognition. The authors identified that topics related to messenger integration, user interaction, and chatbot model training are some of the most popular yet difficult questions to answer.

REFERENCES

[1] Ahmad Abdellatif, Diego Costa, Khaled Badran, Rabe Abdalkareem, and Emad Shihab. 2020. Challenges in chatbot development: A study of stack overflow posts. In *Proceedings of the 17th International Conference on Mining Software Repositories*. 174–185.

[2] Ahmad Abdellatif and Emad Shihab. 2019. MSRBot: Using Bots to Answer Questions from Software Repositories. *arXiv preprint arXiv:1905.06991* (2019).

[3] Chris Brown and Chris Parnin. 2021. Nudging Students Toward Better Software Engineering Behaviors. In *2021 IEEE/ACM Third International Workshop on Bots in Software Engineering (BotSE)*. IEEE, 11–15.

[4] Tapajit Dey, Sara Mousavi, Eduardo Ponce, Tanner Fry, Bogdan Vasilescu, Anna Filippova, and Audris Mockus. 2020. Detecting and Characterizing Bots That Commit Code. In *Proceedings of the 17th International Conference on Mining Software Repositories (MSR '20)*. 209–219.

[5] James Dominic, Jada Houser, Igor Steinmacher, Charles Ritter, and Paige Rodeghero. 2020. Conversational bot for newcomers onboarding to open source projects. In *Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops*. 46–50.

[6] Linda Erlenhov, Francisco Gomes de Oliveira Neto, Riccardo Scandariato, and Philipp Leitner. 2019. Current and Future Bots in Software Development. In *Proceedings of the 1st International Workshop on Bots in Software Engineering* (Montreal, Quebec, Canada) *(BotSE '19)*. 7–11.

[7] Linda Erlenhov, Francisco Gomes De Oliveira Neto, and Philipp Leitner. 2020. An empirical study of bots in software development: Characteristics and challenges from a practitioner's perspective. In *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. 445–455.

[8] Chaiyakarn Khanan, Worawit Luewichana, Krissakorn Pruktharathikoon, Jirayus Jiarpakdee, Chakkrit Tantithamthavorn, Morakot Choetkiertikul, Chaiyong Ragkhitwetsagul, and Thanwadee Sunetnanta. 2020. JITBot: An Explainable Just-in-Time Defect Prediction Bot *(ASE '20)*. Association for Computing Machinery, 1336–1339.

[9] Carlene R Lebeuf. 2018. *A taxonomy of software bots: towards a deeper understanding of software bot characteristics*. Ph.D. Dissertation.

[10] Martin Monperrus. 2019. Explainable Software Bot Contributions: Case Study of Automated Bug Fixes. In *Proceedings of the 1st International Workshop on Bots in Software Engineering* (Montreal, Quebec, Canada) *(BotSE '19)*. 12–15.

[11] Purit Phan-udom, Naruedon Wattanakul, Tattiya Sakulniwat, Chaiyong Ragkhitwetsagul, Thanwadee Sunetnanta, Morakot Choetkiertikul, and Raula Gaikovina Kula. 2020. Teddy: Automatic Recommendation of Pythonic Idiom Usage For Pull-Based Software Projects. In *2020 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. IEEE, 806–809.

[12] Ricardo Romero, Esteban Parra, and Sonia Haiduc. 2020. Experiences building an answer bot for gitter. In *Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops*. 66–70.

[13] Sivasurya Santhanam, Tobias Hecking, Andreas Schreiber, and Stefan Wagner. 2022. Bots in software engineering: a systematic mapping study. *PeerJ Computer Science* 8, e866 (2022).

[14] Margaret-Anne Storey and Alexey Zagalsky. 2016. Disrupting Developer Productivity One Bot at a Time. In *Proceedings of the 2016 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering* (Seattle, WA, USA) *(FSE 2016)*. ACM, New York, NY, USA, 928–931. https://doi.org/10.1145/2950290.2983989

[15] Mairieli Wessel, Ahmad Abdellatif, Igor Wiese, Tayana Conte, Emad Shihab, Marco A Gerosa, and Igor Steinmacher. 2022. Bots for Pull Requests: The Good, the Bad, and the Promising. In *Proceedings of the 44th ACM/IEEE International Conference on Software Engineering (ICSE'22) (ICSE '22)*. To Appear.

[16] Mairieli Wessel, Alexander Serebrenik, Igor Wiese, Igor Steinmacher, and Marco A Gerosa. 2020. Effects of adopting code review bots on pull requests to OSS projects. In *2020 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. IEEE, 1–11.

[17] Mairieli Wessel, Alexander Serebrenik, Igor Wiese, Igor Steinmacher, and Marco A Gerosa. 2020. What to expect from code review bots on GitHub? a survey with OSS maintainers. In *Proceedings of the 34th Brazilian Symposium on Software Engineering*. 457–462.

[18] Mairieli Wessel, Igor Wiese, Igor Steinmacher, and Marco Aurelio Gerosa. 2021. Don't Disturb Me: Challenges of Interacting with Software Bots on Open Source Software Projects. 5, CSCW2, Article 301 (oct 2021), 21 pages.

[19] Marvin Wyrich and Justus Bogner. 2019. Towards an Autonomous Bot for Automatic Source Code Refactoring. In *Proceedings of the 1st International Workshop on Bots in Software Engineering (BotSE '19)*. 24–28.