

УДК 004.415

ДОСЛІДЖЕННЯ МЕТОДОЛОГІЙ РОЗРОБКИ ПЗ ЩОДО ПРОБЛЕМ КОМУНІКАЦІЇ ПІД ЧАС ПРОЄКТУВАННЯ АРХІТЕКТУРИ ПЗ

Шевченко Б. М.

Науковий керівник – доц. Афанасьєва І. В.

Харківський національний університет радіоелектроніки, каф. ПІ
м. Харків, Україна

тел.: +38(050) 578-21-49, e-mail: bohdan.shevchenko1@nure.ua

This work studies the communication issues during modern software development. The ultimate goal is to find a tool that addresses the shortcomings of communication during software design. To do that a Systematic Literature review was performed. Multiple architecture drift types and instruments were found. The found tools were analysed and compared, but only few of the instruments were considered as ready for industry use. To illustrate the requirements a vision of an ideal tool was presented. Also, a 'Battleship' game was developed to show a very simplistic approach to addressing the captured issues.

Розробка програмного забезпечення (ПЗ) існує понад 50 років і залишається динамічною галуззю, якій прогнозується розвиток і надалі. Розробка ПЗ стає все більше командною діяльністю, що підкреслює значення ефективної комунікації та управління знаннями під час розробки. Крім того, розробники ПЗ можуть працювати в різних куточках світу. З цих причин, з початку 21 століття, комунікація стала центральною темою для процесу розробки, і народилася нова низка методологій під назвою Agile. Agile-методології спрямовані на швидку та ефективну комунікацію, а не на розгорнуте планування та документування. Як результат методології Agile зараз використовуються в більшості проєктів [1]. Однак, планування архітектури є важливою частиною розробки, а управління технічними та архітектурними боргами – важливою частиною документування. Планування визначає більшість майбутніх рішень, тоді як документація фіксує відомі проблеми. Найпопулярніші гнучкі методології, такі як Scrum, Kanban, Extreme Programming та DevOps [1], не надають явних настанов щодо керування цими аспектами планування та документування. Дослідження показують, що існують додаткові інструменти для ефективного управління цими аспектами розробки ПЗ.

У ході даного дослідження було виявлено конкретні проблеми, що виникають під час проєктування архітектури програмного забезпечення. Також, було виявлено існуючі інструменти, які можуть використовуватись для зменшення проблем комунікації під час розробки програмного забезпечення. Основний метод дослідження – це систематичний огляд літератури (СЛО) [2]. На додачу, пропонується авторська візія ідеального інструменту. Додатково висновки СЛО було перевірено на прикладі

невеликого сурогатного проекту гри "Морський бій", де було застосовано звичайні нефункціональні вимоги з реальних індустріальних проектів з високим рівнем співпраці [3]. Для інструментів особливу увагу акцентовано на простоті та легкості використання інструментів, а також важливості популяризації та просування як знайдених, так і майбутніх інструментів для полегшення перевірки та впровадження в галузі [4].

Підчас аналізу існуючих інструментів з оглянутої літератури усі інструменти порівнювалися з авторським баченням ідеального інструменту. Для кожного інструменту перевірялося, чи використовувався він на практиці, чи має він іншу форму зворотного зв'язку, наприклад, у формі відгуків з інших досліджень. На жаль, жоден інструмент не відповідає вимогам авторської візії та не був ретельно перевірений на практиці. Найбільш близькими до вказаних вимог є 2 галузеві статті, які зосереджені на перепрофілюванні існуючих практик Agile для зменшення проблем планування та документації. Було виявлено декілька інструментів, які можна відносно легко перевірити на практиці та використовувати щоденно в промисловості. Також допускається, що подальші дослідження знайдених інструментів, які початково не відповідали нашим вимогам, можуть виявити бажані властивості. Наостанок, було визначено дорожню карту для майбутніх досліджень, як для пошуку існуючих інструментів, так і для розробки нових. У будь-якому випадку, спільна риса усіх можливих інструментів – це простота та легкість у використанні, а необхідною вимогою для впровадження в галузі програмної розробки інструменту є популяризація та просування для ефективної перевірки [5].

Список використаних джерел:

1. Marinho, M., Noll, J., Richardson, I., & Beecham, S. (2019). Plan-driven approaches are alive and kicking in agile global software development. 2019 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM), 1–11.
2. Kitchenham, B., Brereton, O. P., Budgen, D., Turner, M., Bailey, J., & Linkman, S. (2009). Systematic literature reviews in software engineering—a systematic literature review. *Information and software technology*, 51(1), 7–15.
3. Bourque, P., & Fairley, R. E. (2014). *Swebok v3. 0: Guide to the software engineering body of knowledge*. IEEE Computer Society, 1–335.
4. MacKay, J. (2017). The psychology of simple. Retrieved October 20, 2020, from <https://web.archive.org/web/20201020072438/https://medium.com/swlh/the-psychology-of-simple-fd5d25a3ca>
5. Gaina, R. (2022). How to market your software or saas product in 11 easy steps. Retrieved May 25, 2022, from <https://web.archive.org/web/20220525201015/https://blog.2checkout.com/how-to-market-software-saas-products/>