

Scientific Software Development with User Stories and Progress Tracking Cards

Osni Marques*, Reed Milewicz†, and Elaine M. Raybourn‡

*Lawrence Berkeley National Laboratory, oamarques@lbl.gov

†Sandia National Laboratories, rmilewi@sandia.gov

‡Sandia National Laboratories, emraybo@sandia.gov

Abstract—User stories are usually seen as a simple technique for communicating requirements between the customer and the developer in agile development, with a wide acceptance in industry. In this State of the Practice talk, we recast user stories in the context of Progress Tracking Cards (PTCs), which are designed to be used by teams toward software process improvement and for aligning software goals with scientific objectives. We argue that scientific software developers can benefit from user stories as a method for articulating requirements, and show how user stories integrated into PTCs can stimulate the adoption of best practices in software development.

I. INTRODUCTION

User stories are a well-known and popular technique for representing requirements in agile development [1]. While there are many ways of writing user stories, the most common template is the one popularized as: “As a *<type of user>*, I want *<goal>* so that *<benefit>*” [2]. Basically, user stories encode client and user requirements; they are intended to be simple, not comprehensive. More than anything else, user stories play the “crucial role of explaining *why* a requirement is expressed” [3]. Outside of user requirements, user stories have also been employed in software process improvement (e.g., on a meta-level as a measure of improvement) [4].

User stories have had great acceptance in industry but their acceptance in scientific software development has been rather limited. Smith et al. observe that requirements are often (mistakenly) considered infeasible for scientific software, with a prevalent negative attitude towards documenting requirements [5]. Scientific software developers tend to be their own users (cf. [6]), and requirements tend to evolve alongside the science and are therefore perceived as unknowable. However, given the ever-growing demand multiscale, multiphysics, and data-intensive simulations and analyses, there is an urgent need to ensure that we develop the right software in the right way. This situation presents an opportunity to re-imagine the culture and practice of scientific software development, including adapting and translating tools, techniques, and methodologies from conventional software industry to the scientific software domain.

As a way of resolving the tension around requirements, we recast user stories in the context of Progress Tracking

Cards (PTCs), which are designed to be used by teams toward software process improvement and for aligning software goals with scientific objectives. We present the application of user stories by the IDEAS-ECP¹ PSIP team, by providing examples of user stories implemented for PTCs in Productivity and Sustainability Improvement Planning (PSIP)². Finally, directions for future work are provided.

II. BACKGROUND

User stories were created for use in this project to engender empathy and deepen understanding of scientific software development needs. In an exercise involving 15-20 participants, 59 user stories were created, refined, and then grouped into categories as follows: (a) *training and documentation*, (b) *software integration and testing*, (c) *software quality*, (d) *practices and standards*, (e) *software requirements and development*, and f) *operational needs*. For example, it was determined that a user story addressing documentation [5] might appeal to teams who wanted to make improvements in this area. To illustrate, a story in category (d) was “*As a software team leader, I want to create a document that describes team values and policies so that team members discuss and understand how our team does its work.*”

III. USER STORIES AND PTCs

PSIP is a lightweight, rather intuitive process which utilizes PTCs to set software quality improvement goals through self-assessment. PTCs are PSIP tools which can be used as a stand-alone or alongside GitHub, GitLab, Kanban, Agile, etc. [9]. Notably: i) PTCs are brief documents articulating the target, or goal, of a software improvement planning activity; ii) with ranked outcomes (scores that can be used to track progress); and iii) indicative of self-assessment, not intended for external assessment or comparison with other projects.

¹IDEAS stands for *Interoperable Design of Extreme-scale Application Software* [7]. The project is funded by the Exascale Computing Project (ECP) [8] of the U.S. Department of Energy (DOE)

²The IDEAS-ECP PSIP team focuses on three thrusts: PTC tool automation (rateyourproject.org); team education/engagement with PSIP PTCs; and team of teams research leveraging *Reposcanner* (github.com/bssw-psip/), a software application used to characterize collaboration and best practices across ECP.

TABLE I
USER STORY IN A PSIP FOR CI.

Target	Testing is run at appropriate times without human involvement and reports are direct and concise.
User Story	As a person responsible for software quality and correctness for my project, I want code regularly tested so that regressions are guarded against and new code is tested against itself and other commits the developer might not have had.
PTC	(0) No CI testing adopted (1) Team adopts a CI method (2) Team adopts a standard time to run specific sets of tests (3) Team develops triggers and scripts to run the tests (4) Team develops methods to prohibit failing code/tests from being integrated (5) Team establishes policy to bypass required testing in the rare cases it is appropriate

User stories fit neatly in PTCs, by capturing desired outcomes for the workflow. Table I provides the example of a user story integrated into a PTC on Continuous Integration (CI). It starts with the *target* (i.e., naming the desired outcome), followed by the *requirements* (i.e., the user story) and the PTC (i.e. stages of goal completion). In this example, progress is scored from 0 (none) to 5 (complete). For example, the Exascale Atomistic capability for Accuracy, Length and Time (EXAALT) team has adopted CI for their long-timescale materials simulation PTCs [10].

A second example of a user story integrated into a PTC is related to (practice) performance regression testing, targeting the development of test suites that exercise performance capabilities and plans for practical use of performance data, given the requirements. Thus the user story is *“As a project lead, I want to be able to identify when changes to the code introduce performance issues so that we can address them sooner rather than later in the development process.”* In this case, scores range from (lowest) no regular performance testing to (highest) performance tracking is run at regular intervals on relevant platforms, and team commits to reviewing the performance tracking information.

The examples above were adapted from the PTC catalog [11], where additional practical applications can be found.

IV. DIRECTIONS AND FUTURE WORK

In our treatment of user stories for scientific software development, the user story becomes a tool for software process improvement and for aligning software goals with perceptible objectives. We have argued that scientific software developers can benefit from user stories as a method for articulating requirements, and have introduced how user stories can be integrated into PTCs with the aim of stimulating the adoption of best practices in software development. The integration of user stories into PTCs serves as an opportunity to identify emerging areas ripe for improvement in scientific software development. Our

community can only control what it can measure and describe, and user stories are a proven tool for describing requirements; this can help our community achieve a better state of practice. Our findings indicate that user stories can be integrated into software development progress tracking tools for process improvements and cultural change that constitute more than HPC.

ACKNOWLEDGEMENTS

This research was supported by the Exascale Computing Project (ECP 17-SC-20-SC), a collaborative effort of the U.S. Department of Energy Office of Science and the National Nuclear Security Administration.

This work was carried out in part at Lawrence Berkeley National Laboratory, managed by the University of California for the U.S. Department of Energy under contract number DE-AC02-05CH11231.

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy’s National Nuclear Security Administration under contract DE-NA-0003525.

REFERENCES

- [1] X. Wang, L. Zhao, Y. Wang, and J. Sun, “The role of requirements engineering practices in agile development: an empirical study,” in *Requirements Engineering*. Springer, 2014, pp. 195–209.
- [2] M. Cohn, *User stories applied: For agile software development*. Addison-Wesley Professional, 2004.
- [3] G. Lucassen, F. Dalpiaz, J. M. E. van der Werf, and S. Brinkkemper, “The use and effectiveness of user stories in practice,” in *International working conference on requirements engineering: Foundation for software quality*. Springer, 2016, pp. 205–222.
- [4] S. Küpper, D. Pfahl, K. Jürisoo, P. Diebold, J. Münch, and M. Kuhrmann, “How has SPI changed in times of agile development? Results from a multi-method study,” *Journal of Software: Evolution and Process*, vol. 31, no. 11, 2019.
- [5] S. Smith, M. Srinivasan, and S. Shankar, “Debunking the Myth that Upfront Requirements are Infeasible for Scientific Computing Software,” in *2019 IEEE/ACM 14th International Workshop on Software Engineering for Science (SE4Science)*. IEEE, 2019, pp. 33–40.
- [6] A. J. Ko, R. Abraham, L. Beckwith, A. Blackwell, M. Burnett, M. Erwig, C. Scaffidi, J. Lawrence, H. Lieberman, B. Myers *et al.*, “The State of the Art in End-User Software Engineering,” *ACM Computing Surveys (CSUR)*, vol. 43, no. 3, pp. 1–44, 2011.
- [7] IDEAS, <https://ideas-productivity.org>.
- [8] The Exascale Computing Project, <https://www.exascaleproject.org>.
- [9] M. A. Heroux, E. Gonsiorowski, R. Gupta, R. Milewicz, J. D. Moulton, G. R. Watson, J. Willenbring, R. J. Zamora, and E. M. Raybourn, “Lightweight software process improvement using productivity and sustainability improvement planning (psip),” in *Tools and Techniques for High Performance Computing. HUST 2019, SE-HER 2019, WHPC 2019. Communications in Computer and Information Science*, J. G and C. S, Eds. Springer, Cham., 2019, vol. 1190, pp. 98–110.
- [10] R. Zamora, “Adopting continuous integration for long-timescale materials simulation,” Sep 2018. [Online]. Available: https://bssw.io/blog_posts/continuous-integration-for-long-timescale-materials-simulation
- [11] PSIP Progress Tracking Card Catalog, <https://github.com/bssw-psip/ptc-catalog>.