Increasing Developer Productivity by Assigning Well-Defined Roles in Teams

Sandra Gesing
University of Notre Dame, sandra.gesing@nd.edu

Developer productivity can be defined and measured in diverse ways. We support that counting lines, number of solved bugs and working hours are not good measures for analyzing developer productivity [1] - we prefer to look at impact and sustainability of scientitic software. Based on these measurements, developer productivity for scientific software is influenced by various factors, some are more objective like good software engineering practices and some are subjective such as teamwork skills or preferences for software development kits.

The work environment as well as team formation have a big influence on software sustainability [2] and, thus, developer productivity. Developers often still work in silos in academia on a specific scientific software and might have a dual role as researcher and developer of scientific software. They are “all-rounders” for the development process and normally have to fill all roles in the development process from gathering requirements to implementing software to quality assurance measures. For the remainder of this paper we focus on developers working in teams for scientific software.

Several universities or national labs have been building developer teams and open up the possibility to offer different roles in such a team. Some developers might be more productive working mostly on specific aspects of the development such as the user interface, some might prefer to work on alll aspects of the project from requirement gathering from researchers to implementation to quality assurance of the resulting product. For different teams at different universities, different approaches work well. The Center for Research Computing (CRC) [3] at the University of Notre Dame and the HUBzero® Team [4] at the University of California, San Diego are examples with more than 10 team members in the role of developers, well-defined responsibilities and successful models for software sustainabiity.

Roles in the HUBzero® team are mostly focused on specific expertise to split work on different projects efficiently. The team consists of experts for different aspects of the HUBzero® framework such as submission to different distributed computing infrastructures or integrating notebooks such as Jupyter notebooks [5]. It works well for HUBzero® since the scientific projects are focused around one technology with a plethora of application areas.

Software development at the CRC involves several different roles. CRC computational scientists are researchers - mostly research faculty in different departments at the university from computer science to psychology - who write proposals with colleagues on campus and/or on national and international level. Besides conducting their own research, their role at the CRC is to bring in funding for the software developers at the CRC. Dependent on their background, they also contribute to the software design. This model allows that developers can focus on the software development process without being concerned about funding.

If a project involves use of HPC infrastructures or Cloud computing, installations and maintenance are supported by the HPC team of the CRC. The programmers are also here welcome to get into the topic as far as they are interested but they do not need to be concerned about maintaining the infrastructure.

The CRC uses the Scrum framework [6] for agile development and has clear role definitions in the development process. While Scrum product owners and Scrum masters are not necessarily also software developers, they are an essential part of the development team. The Scrum product owner is the interface to the Pis of a project and take care of that the development is aligned with the goals of the project, which might involve a deep dive into the research tackled by the scientific software. While the developers can dive deep in the involved research of projects if they would like to, the structure with computational scientists, HPC support and the Scrum framework assures that developers can stay also at a higher level supporting science without becoming necessarily experts in the research topics or on computing or data infrastructure level. The programmers additionally can decide about focus areas they are interested in such as backend or frontend development, research areas, and different programming languages. Most of the projects are developed in Python and Django but there are also projects working with C++ or JavaScript, for example. The CRC supports models for peer-programming, hours for training and takes care of that each project has at least two programmers assigned if it is not a “mini-project” of a small number of weeks. All these measures and responsibilities define the role of developers well and have led to an efficient development process supporting impact and software sustainability. While this model might not fit for every campus, it is one example that works well. Part of the success is not only a clear project management process but also expectations from all team members are managed well.

[1] https://www.7pace.com/blog/how-to-measure-developer-productivity#:~:text=Developer%20productivity%20is%20something%20that's,the%20work%20that's%20being%20done.

[2] Gesing, S., Lawrence, K., Dahan, M., Pierce, M.E., Wilkins-Diehr, N. and Zentner, M. Science gateways: Sustainability via on-campus teams. Future Generation Computer Systems, volume 94, pages 97-102, May 2019.

[3] https://crc.nd.edu/

[4] https://hubzero.org/

[5] Gesing, S., Stirm, C., Zentner, M., Haley, B., Clark, S., and HUBzero Team HUBzero®: Novel Concepts Applied to Established Computing Infrastructures to Address Communities’ Needs. PEARC '19: Proceedings of the Practice and Experience in Advanced Research Computing on Rise of the Machines (learning), Chicago, IL, 2019, July 28 - August 1, 2019

[6] https://www.scrum.org/