Being Permanently Beta:
The Iterative Nature of Computational Tools in Big Data Science

Tracey Hartsock, Brett Robertson, and Kerk Kee
COM-491: Spring 2014, Chapman University; Orange, California

Introduction

This project focuses on the overlapping and recursive relationship between technology development and use, based on the case of computational tools in big-data science. It elaborates on the key activities of these tools and how they are used and implemented by researchers in computational and data-enabled science and engineering. This use and development relationship is a ceaseless test-cycle until the prototype computational tool is mature enough to leave the incubator. Through analyzing 25 semi-structured interviews, the importance of understanding this overlapping and recursive relationship is illustrated through several main themes: the initial need, feedback, and redevelopment of tools through interactive communication between user and developer. This study strives to offer practical strategies and best practices that can be helpful for developers and users of computational tools.

Literature Review

The inquiry of computational tool use and development is rapidly distinguishing the processes of cyberinfrastructure and virtual organization. Communication between developer and user of computation tools proves to be an external analysis as "organizational structure and form are contingent upon factors not inherent to the organization" (Leonardi, 2009). Use and development of the tool encompass a broad range of factors including "discovering the needs of their researchers, setting priorities for support, developing support strategies, funding and implementing cyberinfrastructure, and building partnerships to enhance research support" (Agee, 2010). These factors encompass the steps of initial need, feedback, and redevelopment which are recursive as "research needs must be an ongoing process, not a one-time exercise" (Agee, 2010). Through the steps of computational tool use and development, one can distinguish the steps as successful as they provide "reproducible results, usable and useful, and can be easily maintained and updated" (Baxter, 2006)..

Methodology

- This poster employed the grounded theory approach (Corbin & Strauss, 1990) and analyzed 25 interviews conducted with domain scientists (in bioinformatics, computational chemistry, theoretical physics, etc.) and computational technologists. Interview participants came from across the US (including CA, IL, IN, SC, MI, TX, etc.) and three from the UK (specifically Scotland). Interviews range from 16 minutes to 2:25 hours, with 10 conducted in person at the Supercomputing 2013 conference in Denver, and 15 over the phone, between Nov 2013 and April 2014. Guided by the stated research question, the co-authors performed multiple iterations of data analysis and literature integration, yielding preliminary findings presented in this poster.

Findings: Part 1

Throughout the coding process, three common themes were found within interactive communication between user and developer:

Initial Needs

A marketplace must exist prior to the development because the tool must have problem to solve. However, the need for a tool is forever-changing, including tool type, users, and how the tool will be used. Assessing these initial needs is key.

- "…even before the information gathering, there has to be… a time when you sort of catch the net looking for… what's out there [and] what do people need." (Institute Administrator, IL, November 18, 2013)
- "Sometimes you start off with the users, and then you understand what it is that they need, or sometimes you look for a gap in the market, maybe in functionality or in a way of doing things, and then look for the users who will help refine that." (Physicist & Institute Administrator , IL, November 18, 2013)
- "…[B]uilding it around the use case and I really think it is important in most cases to have that, without a good use case or user community to drive development of the software or platform, it is really easy to get into the… theoretical,… especially when people driving it are computer scientists and developers." (Theoretical Particle Physics Research Scientist, CA, March 19, 2014)

User Feedback

Interactive communication between the developer and user is in a constant feedback loop, as the tool is in a beta-phase and can continue to change. Feedback is key to allow the tool to be tested and re-tested repeatedly.

- "So there are some cases where it's important that the use of the tools and the development of the tools are kept in a very tight feedback loop, specifically for where it's unclear whether or not the hypothesis on the scientific side and the implementation of those hypotheses are correct." (Physicist & Institute Administrator , UK, November 18, 2013)
- "But they don't have that second stage of going back to the stakeholders and say this is how we think it should work. There's a missing step there. So that they can feed back. 'Cause software architects don't get it right every time." (Institute Administrator, UK, November 18, 2013)

Iterative Re-Development

Recursive development of tool and how it is evaluated by the user and developer together.

- …there's also a part from the fact they might not get it right there's also this secondary benefit where basically you get you might once you see the tool laid out on a piece of paper all sort of specked out. Stakeholders might actually have further ideas. After seeing that so you might actually end up with a better tool." (Institute Administrator, UK, November 18, 2013)
- "Every use we've had just develops them further… constantly a process of testing it and optimizing it, and making it better. Every little bit is like…a constant iteration so with every approach or result we get then we have to optimize it and hone it, so its never just a finished tool" (Media Studies Scholar, CA, May 1, 2014)
- "And many time they will – we wouldn't know that at the beginning we'd need this, but after while you need it and oh – maybe this is the way to do it – and this doesn't seem to be working –so then we go back and work on it and estimate when the need comes" (Graduate Student in Computer Science, IL November 11, 2013)

Conclusion

Based on the analysis of interviews, we concluded that the development of computational tools for big data science involves assessing users' initial needs, receiving feedback, and engage in iterative redevelopment. First, before a tool can be developed, used, and tested, there has to be a preexisting problem the development of the tool will solve. The "marketplace" as interviewees called it, is the place in which the initial need is identified. From that market, developers worked with users to distinguish a tool that would meet the initial need. Second, any type of development feedback is key to help developers perfect their tool. Third, computational tools are often in a beta-phase, developed, tested, re-developed, and re-tested. Although this three-step sequence appear linear at first glance. Future research will explore how communication between user and developer is not always as effective and efficient as expected.

References

Agee, A. et al. (2010). Building Research Cyberinfrastructure at Small/Medium Research Institutions. Educause Review Online. 1-16.
Baxter, S. et al. (2006). Scientific Software Development Is Not an Oxymoron. PLoS Computational Biology. 975-978.
Corbin, J., & Strauss, A. (1990). Grounded theory research: Procedures, canons, and evaluative criteria. Qualitative Sociology, 13(1), 3-21
Leonardi, P. (2009). Crossing the Implementation Line: The Mutual Constitution of Technology and Organizing Across Development and Use Activities. Communication Theory, 278–310