



The Role of Feedback in the Nonlinear Development and Use of Computational Tools in Big Data Science

Dean Navarro, Brett Robertson, and Kerk Kee

COM-491: Spring 2014, Chapman University; Orange, California

Introduction

This poster is motivated by the observation that cyberinfrastructure (CI) implementation, especially the computational tools embedded in the larger platform, is not well understood because these tools represent a new generation of technologies that are 'dynamic'. These tools are considered dynamic because they are permanently in a beta phase, being launched with the intent to be constantly updated after feedback is collected on the development and usability of the tools. The key focus on this poster is therefore on the role of feedback in linking development and use through the different stages of tool evolution. By analyzing interview data about computational tools, a theoretical framework on 'feedback' is proposed to further understand the ongoing process of a tool being simultaneously developed and tested, redeveloped and retested. The ultimate aim of this project is to help identify strategies to promote prototype computational tools developed in virtual scientific teams to evolve and mature into stable tools for the greater scientific community.

Literature Review

"Although the definition and process of translation is often murky within Actor-Network Theory research, it is generally viewed to be a political process whereby actors use any means at their disposal to build a network of heterogeneous actors that will ultimately allow a technology to develop in a certain desired direction" (Neff & Stark, 2002). Researchers have suggested that technological development always occurs within a sociopolitical context in which actors negotiate and use persuasive techniques to influence the evolution of a technology's material features. Researchers have linked this technological change to the activities associated with a technology's development (Leonardi, 2009). "Because different sets of actors often have diverse interests, the development of a technology rests on the ability to translate—reinterpret, re-present, or appropriate—others' interests to one's own" (Law, 1992). "The process of continual technological change necessitates a responsiveness to change through openness in organizational form, adaptability by employees, and, in the most positive form of permanently beta, broad participation in design" (Neff & Stark, 2002). Researchers such as Layton (1971) and Callon (1980) encouraged the idea of how technology rarely develops in a linear fashion. Instead, they were subject to changes at many stages brought about by social and political interests in their development (Leonardi, 2009). Without linear innovation the process of development occurs in a permanently beta model, requiring users' feedback and collaboration in order to meet their demands of a successful tool.

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

Throughout the coding process, three common themes were found within the framework of 'feedback':

Collaborating In Development	Importance Feedback	Collaborating With Feedback
<p>The nonlinear model incorporates collaboration with the user and developer through the development of a tool. In order to stay consistent with a permanently beta model, this collaboration is needed before and during development.</p> <p>"When they develop for core simulation code that can be a problem because one person can introduce a bug and then 10 minutes later all you know is it's broken, you don't know what happened. That creates problems so for large cyber-infrastructure projects, you really need to have defined methods of collaboration. Actually, for the Amber project, I implemented a continuous integration system to help us with this development process. And what that does is to test a code every time after one person adds a change to the main repository and that way you can see step-by-step basically who broke it and what kind of changes were involved in that. So, trying to fix it all before release" (Undergraduate Molecular Biologist at San Diego Supercomputing Center, CA, 3/19/2014).</p> <p>"On the other hand, there are other areas whereby you could say that the research, the knowledge in the computational tool, that comes from say computer science, is relatively old compared to the use of the tool in a new domain. In that case, there tends to be less collaboration, although that's maybe a bad thing, and there should still be a strong feedback dynamic, but the difference is that there is less of an incentive for that feedback to be taken in by the computer scientist who sits on the left of your diagram because here, if you imagine that there has been research that leads to the development of the tool, this is very much more of a one-way process than the two-way process you have between the developers of the tool and the users of the tool" (Theoretical Physicist, Edinburgh, 11/18/2013).</p>	<p>Making the user's ability to give feedback easy will yield significant results needed for tool development. Without user's feedback, the development process will never supersede the linear development model of a tool.</p> <p>"I am the user, and oftentimes I was saying to the computer scientist 'I know this is possible, I want to do it, it shouldn't be my problem. You are the developers. You make it happen. I want to see results. ' You know? But I'm interested enough in being able to do this routinely that I think it is worth my time to participate in the develop process, or else it will just never happen" (Computational Biologist, LA, 4/22/14).</p> <p>"I think that it would be really important to continuously receive feedback from that tool and not abandon the people that were helping adopting it." (XSEDE Trainer and Outreach Coordinator, IL, 3/19/2014)</p> <p>"In the words of Marc Andreessen, the founder of Netscape, the philosophy behind so many beta releases was to "kick it out the door. It may not even work reliably, go out and get feedback, [Customers] will tell you, often in no uncertain terms, what's wrong with it, and what needs to be improved. Netscape risked reliability for responsiveness"</p>	<p>The creation of a tool must be based off of what users want user demand and in order to do so, you must be in tune with their feedback. The ability to receive their feedback gives the developer the chance to use the right information in collaboration with their development.</p> <p>"But if I'm trying to get it adopted or trying to get the feedback from people who adopt, then I have to make it easy for them. And this is another thing that people don't put enough effort into: they don't learn the standards; they don't go to the effort, and it is effort to make sure it will work well in other places. So, there's a lot of blame-the-customer sometimes, and so we put a lot of energy into that" (Computational Scientist, IL, 11/21/2013).</p> <p>"People start using it, and then they start giving you questions. It's like, I would like to do this; why doesn't it do that; or this didn't work. And you go back and you look at it, and you say, oh, we never thought anybody would use it like that, and we don't have an algorithm for that yet, but we have an idea. Okay, and well that... and now you can do research and you know... I mean one of the things that's very hard is picking research directions: what topic do you do research on? And we got just incredibly valuable feedback about what was important in the community, where was research needed, which we would not have gotten without people using the stuff we had done and giving us that kind of feedback" (Computational Scientist, IL, 11/21/2013).</p>

Conclusion

Given the CI investments by NSF and other federal agencies, our findings have helped identify the best practices for successful development of computational tools, as well as optimized existing and future research. In depth research and interviews from professionals in the science/computer science industry have given us a strong hold on the importance of feedback from users of computational tools. We have discovered that in order to create a successful tool which will satisfy consumers, developers publish their tool with foreknown strategy to upgrade it rather than publishing a final product. These upgrades occur in a nonlinear fashion of the development/publication of a tool consistent with the feedback they receive from the tool's users. The daily consumer demands the ability to tell a developer what it is they would like to be updated in their favorite app, and the developer cannot succeed until the consumer's needs are satisfied. Our findings enable the public to have a say in what new technology they would like to be developed, as well as certain upgrades they demand for existing tools. In this study we have not just discovered the importance of user's feedback for developers, but also when the feedback is the most impactful, who needs to hear the feedback, and what type of feedback can have the largest influence on developers.

References

- Corbin, J., & Strauss, A. (1990). Grounded theory research: Procedures, canons, and evaluative criteria. *Qualitative Sociology*, 13(1), 3-21.
- Leonardi. (2009). Crossing the implementation line: The mutual constitution of technology and organizing across development and use activities, 19(1), 278-310.
- Law, J. (1992). Notes on the theory of the actor-network: Ordering, strategy, and heterogeneity. *Systems Practice*, 5(4), 379-393.
- Neff, G., & Stark, D. (2002, January 1). Permanently Beta: Responsive Organization in the Internet Era . . Retrieved May 1, 2014, from http://www.umass.edu/digitalcenter/events/2002Workshop/Papers/stark_permanently.beta.pdf