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
All of the worlds most successful tools started with an idea. An idea in which to solve the problems that inhibit social growth professionally, culturally, and resourcefully. An example of this tool development can be found in the pockets of people across the nation – the Apple iPhone. The iPhone was a tool which the world did not realize would be something of great necessity. The marketplace was unsure of the value behind a smart cellular device and could not imagine the impact it would have on the world. Since then, this tool has solved problems of quick communication, reacted with users in distinguishing further development, and applied additions to make a better product. It is a prime example of a ceaseless redevelopment tool as it continues to provide updated versions to the public – as the iPhone 6 is underway. Therefore, 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 computational 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 encompasses 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 re-development 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). Therefore, the research question can be raised – “What is the importance and process behind tool development and how do users play a significant role?”

Methodology
- This poster employed the grounded theory approach (Corbin & Strauss, 1980) 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

Initial Needs
Most computational tools are developed with existing needs and/or known problems in mind. A marketplace must exist prior to the development of a tool. Therefore, assessing these initial needs and understanding the problems is key.

- “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.”
  (Computational Physicist & Institute Administrator, UK, November 18, 2013)
- “[Building it around the use case… I really think it is important in most cases to have that. Without a good use case or user community to drive [the] development of the software or platform, it is really easy to get into the… theoretical [i.e., hypothetical], not based on real needs,]… 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 users’ needs may evolve, and the scientific problems driving tool development may be a moving target during the research process. Therefore, user feedback is key to allow the tool to adapt to emerging needs.

- “So there are some cases where it’s important that the user 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.”
  (Computational Physicist & Institute Administrator, UK, November 18, 2013)
- “[Software architects don’t get it right every time… once you see the tool laid out on a piece of paper all sort of specked out. Stakeholders [i.e., users] might actually have further ideas. After seeing that so you might actually end up with a better tool”
  (Physicist, Institute Administrator, UK, November 18, 2013)

Ceaseless Re-Development
Recursive development of tool and how it is evaluated by the user and developer together. This ceaseless cycle distinguishes the next steps in the tools existence and the adaption that occurs for the tool to continue to prosper. It is important for a tool to continue to develop so it remains an important aspect and valuable tool in today’s ever-changing society.

- “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”
  (Computational 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 continue to improve 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. Concerning an education implication: Assessing initial needs and staying on top of how the needs evolve, then engaging in multiple and ceaseless re-development is the reality of software development in the real world. However, computer science students are not trained to do these three things. This process of evolving and developing through initial drafts or “rough drafts” is uncommon for many students today. Therefore, it is almost difficult for science professional because during academia assignments of codes and programs to address a hypothetical need, are turned in for a grade, and a good grade is all that is sought after. Therefore, computer science educators need to re-think the way computer science students are trained – their integration will falter if they are unable to expect an experience that requires patients and time (that may not always end with an exceptional grade.)

References

This material is based upon work supported by the National Science Foundation under Grant No. ACI 1323265. This poster was presented on May 14, 2014 at the 2014 Chapman University Student Research Day, Orange, CA.