The Pursuit of Practice-Impactful Research

Tao Xie
University of Illinois at Urbana-Champaign
taoxie@illinois.edu

DOI: 10.1145/2853073.2853077
http://doi.acm.org/10.1145/2853073.2853077

In the software engineering research community, there can be different target audiences that researchers' research aims to produce direct impact for [1]. For example, some researchers conduct their research to produce direct impact on software engineering practices (which broadly include those for both proprietary software and open-source software); some other researchers conduct their research to produce direct impact on other researchers in the research community (sometimes also indirect impact on software engineering practices eventually).

While continuing focusing on basic research [2], Microsoft Research has started "Microsoft Research New Experiences and Technologies, or MSR NExT, an organization of world-class researchers, engineers, and designers devoted to creating potentially disruptive technologies for Microsoft and the world. While NExT will continue to advance the field of computing research and produce work with significant scholarly impact, its priority is developing technologies that benefit Microsoft and the world more broadly.", as stated on the homepage of the MSR NExT leader Peter Lee [3].

Some academic researchers may say that such focus on developing technologies that benefit a company (and the world more broadly) would/should be limited in an industrial lab, and not be applicable to a university group. However, some other academic researchers disagree: without being satisfied by just publishing papers after papers (even in top venues), they take great efforts to transfer their research outcomes to practices via various forms, e.g., startups for product commercialization, tool/infrastructure releasing for practice adoption, and academy-industry collaborations for technology transfer.

In May 2015, MIT President L. Rafael Reif stated in his Washington Post editorial [4]: "Find ways to shorten the full span from idea to impact, reducing it from, say, 10 years to five. There's a growing body of evidence from MIT and elsewhere that in a range of high-potential tangible fields such as nano-manufacturing and materials science, it may be possible to reproduce the process of rapid, relatively low-cost refinement and iteration that is so powerful in advancing purely digital concepts. We could also speed the process by helping researchers more efficiently master the best practices of science-based entrepreneurship. LiquiGlide's founders did not wait to finish their scientific thinking before focusing on manufacturing: They refined the research and worked out how to scale up production in parallel."

The above statements from the MIT President remind me of a finding from the ACM SIGSOFT Impact Project (http://www.sigsoft.org/impact.html): "research impact might not be fully felt for at least 10 years" [1]. For some cases, the time may be even much longer. For example, it has taken nearly 30 years for refactoring [5] to become a central part of software engineering practice, rooted from Bill Opdyke's dissertation research (advised by Ralph Johnson) at the University of Illinois at Urbana-Champaign and Bill Griswold's dissertation research (advised by David Notkin) at the University of Washington at Seattle.

While acknowledging the difficulty and taking the patience for technology transfer, our research community shall sufficiently appreciate and celebrate the efforts and achievements made by researchers to shorten the span from research to impact on practice, as advocated by the MIT President.

Here are some startup examples (there are many more than those listed here) rooted from university research: Pattern Insight Inc. (http://patterninsight.com/) founded based on research from Yuanyuan Zhou's research group, Tasktop Technologies Inc. (http://www.tasktop.com/) founded based on research from Gail Murphy's research group, and Testfabrik AG (https://testfabrik.com/) founded based on research from Andreas Zeller's research group. Some university research groups take great efforts to develop, release, and maintain open source tools or infrastructures widely used in practices. There are many such examples. A noteworthy example is LLVM (http://llvm.org/), a high-impact compiler infrastructure, started and directed by my next-door-office neighbor Vikram Adve at the University of Illinois at Urbana-Champaign and his former PhD student Chris Lattner (now at Apple). In addition, many academic researchers have been collaborating with companies or industrial labs for producing high practice impact (some discussion and reflection were made by Lionel Briand [7]). My university research group has been collaborating with researchers from the Pex group (http://research.microsoft.com/pex/) and the Software Analytics group (http://research.microsoft.com/sa/) at Microsoft Research, resulting in various lessons and experiences learned in high-impact technology transfer [8, 9, 10, 11].

Given that new generation of young researchers may tend to put their eye sights on publishing (many) papers in top venues without paying sufficient attention to research impact, it is time for our research community to incentivize impact [12], including but not limited to impact to practice.

Note that the above discussion is not limited to software engineering practices for proprietary software but also applicable to software engineering practices for open-source software. These days, open source projects supply valuable data to academic researchers for them to carry out or evaluate their research (on either empirical studies or tooling). However, many academic researchers often publish their research and then stop, without enabling/helping open source projects' practices to adopt their research (which was "empirically shown" to be able to benefit these projects substantially based on the data from these projects). As further reading, you may read and reflect on some complaints in a blog post (http://exple.tive.org/blarg/2015/07/24/hostage-situation/) made by a manager of the Mozilla project.

If you have thoughts (either agreement or disagreement with my thoughts above) on the topics of history and impact, please submit your contributed articles to this History and Impact Column by sending them to taoxie@illinois.edu. Have a wonderful impactful 2016!

1. L. J. Osterweil, C. Ghezzi, J. Kramer, A. L. Wolf. Determining the Impact of Software Engineering Research on Practice. IEEE Computer, 41(3), pp. 39-49, March 2008.
2. J. M. Wing. Here's Why Microsoft Cares about Basic Research -- and You Should, Too. Inside Microsoft Research. 19 Oct 2015. http://blogs.technet.com/b/inside_microsoft_research/archive/2015/10/19/why-microsoft-cares-about-basic-research-and-you-should-too.aspx
3. Peter Lee's Microsoft Research Homepage. http://research.microsoft.com/en-us/people/petelee/. Accessed in December 2015.
4. L. R. Reif. A Better Way to Deliver Innovation to the World. The Washington Post. May 22, 2015. https://www.washingtonpost.com/opinions/a-better-way-to-deliver-innovation-to-the-world/2015/05/22/35023680-fe28-11e4-8b6c-0dcce21e223d_story.html
5. W.G. Griswold and W.F. Opdyke. The Birth of Refactoring: A Retrospective on the Nature of High-Impact Software Engineering Research. IEEE Software, 32(6), pp. 30-38, Nov.-Dec. 2015.
6. T. Xie. Learning and Celebration of Software Engineering History and Impact. SIGSOFT Softw. Eng. Notes, 40(1), pp. 7-8, February 2015.
7. L. C. Briand. Embracing the Engineering Side of Software Engineering. IEEE Software, 29(4), pp. 96, July/August 2012.
8. N. Tillmann, J. de Halleux, and T. Xie. Transferring an Automated Test Generation Tool to Practice: From Pex to Fakes and Code Digger. In Proc. ASE 2014, Experience Papers, pp. 385-396, 2014.
9. Y. Dang, D. Zhang, S. Ge, C. Chu, Y. Qiu, and T. Xie. XIAO: Tuning Code Clones at Hands of Engineers in Practice. In Proc. ACSAC 2012, pp. 369-378, 2012.
10. D. Zhang, S. Han, Y. Dang, J.-G. Lou, H. Zhang, and T. Xie. Software Analytics in Practice. IEEE Software, Special Issue on the Many Faces of Software Analytics, 30(5), pp. 30-37, September/October 2013.
11. J.-G. Lou, Q. Lin, R. Ding, Q. Fu, D. Zhang, and T. Xie. Software Analytics for Incident Management of Online Services: An Experience Report. In Proc. ASE 2013, Experience Papers, pp. 475-485, 2013.
12. B. Friedman and F. B. Schneider. Incentivizing Quality and Impact: Evaluating Scholarship in Hiring, Tenure, and Promotion. CRA Best Practices Memo. February 2015. http://cra.org/resources/best-practice-memos/incentivizing-quality-and-impact-evaluating-scholarship-in-hiring-tenure-and-promotion/