Statement on Research, Teaching and Service

Premkumar Devanbu
Research Professor,
(Distinguished Professor Emeritus, On Recall Appointment for Research)
UC Davis

1 Research

"Naturalness" of Software This line of work, pioneered here at UC Davis, in 2012 [13, 27, 20] began with the observation that code is repetitive & predictable, like Natural Language. We realized that a range of different NLP and machine-learning methods could be adapted to work for software engineering tasks. We have developed several applications: to code suggestion [13, 28], defect finding [6], error correction (in review), code de-obfuscation [1], and also on adapting NLP methods for code [9, 2] and the other on the value of Deep Learning methods [2] for language modeling for source code. Other related works include a study of the causes of Naturalness [14], and a survey of the field [8]. We have also examined the proliferating use of enoder-decoder deep learning models in software engineering, showing that these models are of questionable value in synthesizing patches for defective code [32] and comment synthesis from code [32]. One project, using language models to repair syntax errors is in submission, and a revision is in review. A new project, on using "naturalness" phenomena in binaries for reverse engineering tasks has begun, with a contract from Sandia Labs; paper is in review. This work has very substantial impact. Microsoft, Facebook, Google, and Amazon have deployed applications, and groups actively doing R&D. Visual Studio has features using our work, as does Amazon's CodeGuru product. Multiple workshops (several NSF sponsored) have been held, at ESEC/FSE, ICSE, ASE, Dagstuhl, and AAAI on this topic, and a very active academic community is engaged in the topic; it's very gratifying that it all started here, at Davis!

Cognitive Science of Program Comprehension In collaboration with Emily Morgan (Linguistics, UC Davis), we have been exploring the role of predictability in Code Comprehension. This connection is well explored for natural languages, but is novel for code. Our first papers were very well received, at the flagship Cognitive Science Conference [11], at ICSE NIER [10] and a journal paper for Cognitive Science Journal is in review.

Empirical Studies of Software Engineering In collaboration with Vladimir Filkov I have been pursuing various observational studies in the area of software quality & productivity, using open-source project historical data.

Social/Human Factors: One line of work explored the effect of social and human factors on software productivity and quality in GitHub. Thanks to the large volume of data, it is possible to perform natural experiments using multivariate regression with large sample sizes. Our studies include the effect of Gender in software teams [12], collaboration [7], multi-tasking [5], and also the ways that people join open-source projects [18]. Our early work in mining email social networks [23], and our work emergent latent social structures therein [24] were quite influential. More recently we looked at social processes in GitHub surrounding code review and defect repair [35, 34].

Tools & Methods: We have worked on the evaluation of the effectiveness of tools and methods choices in software engineering. We have evaluated the value of asserts [30], the value of continuous integration [17] and the factors influencing acceptance of new code under the pull request model [22].

Empirical Foundations We have published a number of influential, well-cited papers that critique the assumptions and the methods used in empirical software engineering. Our work on bias issues in bug-fix data [25] and the ecological fallacy that can arise in hierarchically structured software products, teams and processes [26] have been durably influential.

In a separate work, during a visit to Microsoft, in collaboration with Microsoft Researchers, I studied the inconsistencies between the beliefs of developers, and the actual evidence arising out of the projects they work in [3].

2 Impact & Recognition

Thanks to great colleagues (D'Souza, Filkov, Rubio, Thakur), UC Davis is ranked within the top 10 in the Country in Software Engineering Research. Again thanks to wonderful collaborators, I became an ACM Fellow in 2018. The citation reads: *"For contributions to using software data and meta-data to improve software tools and processes"*. In addition, I won ACM SIGSOFT Outstanding Research Award (in 2021) *"for profoundly changing the way researchers think about software by exploring connections between source code and natural language"*. I also won the Alexander von Humboldt Research Award from the eponymous foundation in Germany in 2021. Other evidence of durable influence of our work is listed below.

1. A 2006 paper [23] (with my Ph.D student Chris Bird, now at Microsoft Research, and others) won the *10 year most influential paper award* (10 year MIP) at the Mining Software Repositories (MSR) conference in 2016. This award is given papers judged by the community to be most durably influential paper from 10 years ago, and is given each year.

2. A 2009 paper [4] (with Bird and several other authors) won the *10 year MIP award* at MSR 2019

3. A 2008 paper [24] at ESEC/FSE, (with Bird, and others) won the *Test-of-Time* award at ESEC/FSE 2018, which usually given to papers that have been most durably influential over the 10 year period.

4. A 2009 paper [25] at ESEC/FSE, (with Bird, and others) won the *Test-of-Time* award at ESEC/FSE 2019, which usually given to papers that have been most durably influential over the 10 year period.

5. A 2011 paper [21] at ESEC/FSE (with Bird, and others) won the *Test-of-Time* award at ESEC/FSE 2019, which usually given to papers that have been most durably influential over the 10 year period.

6. A 2012 paper [13] won the *Most Influential Ppaer* award at ICSE 2022.

7. A 2009 paper at ISSRE (with Bird, and others was listed a "Highlight from 30 years of ISSRE" in a retrospective collection published in 2019[1].

8. A paper at ICSE 2009 (with Bird and others) won the Distinguished Paper Award, and was invited to appear in *CACM Research Highlights* which showcases the best current work in Computer Science.

[1] `http://2019.issre.net/node/77`

9. A paper at ASE 2011 (with Posnett and others) won the Distinguished Paper Award
10. A paper at ICSE 2013 (with Posnett and others) won the Distinguished Paper Award
11. A paper in ICSE 2012 on Naturalness also appeared in the *CACM Research Highlights*.
12. A paper at ICSE 2015 NIER won the best paper award.
13. A paper from FSE 2014, (with Baishakhi Ray and others) was also appeared *CACM Research Highlights*.
14. My former PhD students and Postdocs were placed in permanent positions at well-regarded institutions: Carnegie-Mellon (Vasilescu[2], 2016, Hellendoorn 2019), Columbia (Ray, 2017), University College London (Barr, 2013), Alberta (Hindle, 2012), Microsoft Research (Bird, 2010), and UBC (Wohlstadter, 2004).

3 Teaching

Software engineering practice has evolved very quickly to reflect new demands of speed, cost and quality, and I felt that our teaching practice needed to reflect that. I spent a great deal of time in recent years trying to bring innovative content and practice to our SE curriculum. I believe our SE curriculum is among the best in the world. Some major efforts are listed below.

ECS 193 A/B was started as a capstone project course, by a colleague, Ken Joy. A few years ago, I redesigned 193 A/B to comprise an Agile process, with sprints, and greatly increased instructor contact. This course has been taken over by Prof. Xin Liu, and to my knowledge she has retained some of these ideas.

I pushed to redesign ECS 160, *Software Engineering*, into two parts, 160A and 160B. 160A will emphasize advanced programming techniques: design patterns, functional programming, Java Streams, *etc.* 160B will be focused on the innovative project approach developed by colleague Chris Nitta. With exclusive focus on advanced programming, 160A could align homeworks to reinforce classroom pedagogy. 160A was very well received by students, and the split has been approved.

I have introduced a new Sophomore Level course, ECS 161, *Modern Software Tools* . The impetus for this course arose from two recent trends: a) burgeoning adoption of the agile-derived DevOps process framework, which integrates the traditionally distinct development and operations processes, and b) the widespread use of distributed/open development processes incorporating version control, continuous integration, code review, and the pull-request model. Several protracted discussions with recent alums and colleagues at Facebook, Google, etc., increased our awareness of the pervasive impact of these ideas, and the need to train our students in the underlying concepts and tools. It was clear that none of the other courses in the curriculum was teaching students about version control, modern testing and code-review tools, de-centralized build systems, continuous integration, virtualized/continuous deployment, and such.

The importance of internships for our students is well-recognized, and being well-versed in these concepts and tools greatly increases their viability in the job market for internships. This course adopts a lab-style format, with 2 hours of lectures, and 2 hours of labs–the instructor is present for all four hours. During the lab sessions, we present assignments and support student learning:

[2] Vasilescu was co-supervised with Filkov

scaffolding is provided, and students are assisted in completed the first part of homeworks. They are encouraged to collaborate, and learn from each other. The course is very well-subscribed, and in demand.

Finally, working with Gerardo Con Diaz, a colleague from the Science & Technology Studies Department, I designed a new *Conscientious Computing* minor, primarily intended for our Computer Science majors. This minor largely comprising courses outside of Computer Science, and is designed to train students who can think about Computing in broad terms: it's effect on society, politics, freedom, oppression, *etc*. This initiative has been approved by our Undergraduate Committee, and is in consideration at the College of Engineering. The Science & Technology studies department has already approved it; our hope is to have it ready for incoming students in Fall 2021. Students graduating from this program will emerge ready to tackle difficult issues of ethics, public policy *etc*, in the context of computing applications.

Graduate Teaching I continue to teach the Graduate Software Engineering Course ECS260, focusing heavily on empirical software engineering. I also have run Graduate Seminars (289) on Empirical & Formal Software Engineering, as well as one on Machine Learning applications to Software Engineering.

Graduate & Post-Doc Supervision: I am now supervising 3 PhD students, Kevin Jesse, Toufique Ahmed, and David Gros, and one postdoc, Anand Sawant. I have graduated several PhD students, including: Foyzur Rahman, Daryl Posnett, Christian Bird, Vincent Hellendoorn, Casey Casalnuovo. I have also supervised several postdocs, including Abram Hindle (now faculty at U Alberta), Niraj Kumar (Samsung Research, India), Jennifer D'Souza (Leibniz Informationzentrum, Germany), Earl Barr (faculty at University College London), Baishakhi Ray (faculty, at Columbia) and Bogdan Vasilescu (now faculty at CMU, co-supervised with Filkov)

4 Service

Intramural I've served on several intra-mural committees, at various levels; I am also on a faculty advisory board for the Shrem Museum of Art.

Extra-mural Service & Visibility I am frequent PC member of ICSE, ESEC/FSE, and MSR. I serve as general chair of SIGSOFT FSE 2020, which is major, high-visibility task. I also co-organized several successive of NSF Sponsored Workshops on "Naturalness", held at Microsoft Research, in 2015, and in conjunction with SIGSOFT FSE (in 2016), and then at AAAI in 2018.

I gave several Keynotes and Plenary Talks, most notably: ESEM (2010), ICSE Doctoral Symposium (2012), ISEC (2013), Brazilian Symposium on Software Engineering (2015), Chilean Computer Science Conference (2015), ICSE (2022)

I was an invited Distinguished Lecturer at UMBC (Sept 2013) University of Maryland (2014), University of Nebraska (2014) University of California, Irvine (2016), George Mason (2018) and University of Houston (2020, cancelled due to the Pandemic).

I was on the Editorial Board of ACM TOSEM (2002-2005), IEEE Transactions on Software Engineering (2006-2010), Springer-Verlag EMSE (2009-2015) and Wiley Journal of Software Maintenance & Evolution (2009-2015). I have recently appointed co-Chair (along with prof. m.c. schraefel) of the Contributed Articles track, on the editorial board of Communications of the ACM.

References

[1] Hellendoorn, V.J., and Devanbu, P., ARE DEEP NEURAL NETWORKS THE BEST CHOICE FOR MODELING SOURCE CODE? *SIGSOFT FSE 2017*

[2] Vasilescu, B., Casalnuovo, C., and Devanbu, P. RECOVERING CLEAR, NATURAL IDENTIFIERS FROM OBFUSCATED JAVASCRIPT NAMES, *SIGSOFT FSE 2017*

[3] Devanbu, P., Zimmermann T., and Bird, C., BELIEF & EVIDENCE IN EMPIRICAL SOFTWARE ENGINEERING. *ICSE 2016*

[4] Bird, C., Rigby, P.C., Barr, E.T., Hamilton, D.J., German, D.M. and Devanbu, P., THE PROMISES AND PERILS OF MINING GIT. *MSR* 2009.

[5] Vasilescu, B., Blincoe, K., Xuan, Q., Casalnuovo, C., Damian, D., Devanbu, P. and Filkov, V., THE SKY IS NOT THE LIMITE: MULTITASKING ACROSS GITHUB PROJECTS. *ICSE 2016*

[6] Ray B., Hellendoorn, V., Godhane, S., Tu, Z., Bacchelli, A., and Devanbu, P., ON THE NATURALNESS OF BUGGY CODE. *ICSE 2016*

[7] Xuan Q., Devanbu, P., and Filkov V., CONVERGING WORK-TALK PATTERNS IN ONLINE TASK-ORIENTED COMMUNITIES, *PloS ONE Vol 11, No. 5*

[8] Allamanis, M., Barr, E.T., Devanbu, P. and Sutton, C., A SURVEY OF MACHINE LEARNING FOR BIG CODE AND NATURALNESS. *ACM Computing Surveys* 2018

[9] Tu., Z., Su., Z., Devanbu, P, ON THE LOCALNESS OF SOFWARE, *ESEC/FSE 2014*

[10] Casalnuovo, C., Barr, E.T., Dash, S.K., Devanbu, P. and Morgan, E., A THEORY OF DUAL CHANNEL CONSTRAINTS. *ICSE NIER* 2020.

[11] Casalnuovo, C., Devanbu, P., Morgan, E., DOES SURPRISAL PREDICT CODE COMPREHENSION DIFFICULTY? *Cognitive Science Society Conference*, 2020

[12] Vasilescu, B., Posnett, D., Ray, B., van den Brand, M., Serebrenik, A., Devanbu P., and Filkov, V., GENDER AND TENURE DIVERSITY IN GITHUB TEAMS, *CHI 2015*

[13] Hindle, A., Barr., E., Gabel, M., Su, Z., Devanb, P., ON THE “NATURALNESS” OF SOFTWARE, *ICSE 2012*

[14] Casalnuovo, Casey, Kenji Sagae, and Prem Devanbu. STUDYING THE DIFFERENCE BETWEEN NATURAL AND PROGRAMMING LANGUAGE CORPORA *Springer-Verlag Journal Empirical Software Engineering* 24.4 (2019): 1823-1868.

[15] LeGoues, C., Holtschulte, N., Smith, E., Brun, Y., Devanbu, P., Forrest, S., and Weimer W., THE MANYBUGS AND INTROCLASS BENCHMARKS FOR AUTOMATED REPAIR OF C PROGRAMS *IEEE Transactions on Software Engineering 41(12), 2015*,

[16] Thung, F., Lo, D., Jiang, L., Rahman, F., and Devanbu, P., TO WHAT EXTENT COULD WE DETECT FIELD DEFECTS? AN EXTENDED EMPIRICAL STUDY OF FALSE NEGATIVES IN STATIC BUG-FINDING TOOLS, *Automated Software Engineering Journal, 22(4), 2015*

[17] Vasilescu B., Yu, Y., Devanbu, P., and Filkov, V., QUALITY AND PRODUCTIVITY OUTCOMES RELATING TO CONTINUOUS INTEGRATION IN GITHUB *SIGSOFT FSE 2015*

[18] Casalnuovo, C., Vasilescu B., Devanbu, P., and Filkov V., DEVELOPER ONBOARDING IN GITHUB: THE ROLE OF PRIOR SOCIAL LINKS AND LANGUAGE EXPERIENCE *SIGSOFT FSE 2015*

[19] Hellendoorn, V., Devanbu, P., and Bacchelli, A., WILL THEY LIKE THIS?: EVALUATING CODE CONTRIBUTIONS WITH LANGUAGE MODELS, *MSR 2015*

[20] Hindle,A., Barr, E., Gabel, M., Su, Z., and Devanbu, P. On the Naturalness of Software *CACM Research Highlights, Vol 59, No.5*, 2016

[21] Bird, C., Nagappan B., Murphy, B., Gall, H., Devanbu, P., Don’t touch my code!: examining the effects of ownership on software quality *ESEC/FSE* 2011

[22] Yu, Y., Wang, H., Filkov, V., Devanbu, P., and Vasilescu, B., WAIT FOR IT: DETERMINANTS OF PULL REQUEST EVALUATION LATENCY ON GITHUB *MSR 2015*

[23] Bird, C., Gourley, A., Devanbu,P., Gertz, M., Swaminathan, A. MINING EMAIL SOCIAL NETWORKS. *MSR* 2006

[24] Bird, C., Pattison, D., D’Souza, R., Filkov, V., and Devanbu., P. LATENT SOCIAL STRUCTURE IN OPEN SOURCE PROJECTS. *ESEC/FSE* 2008

[25] Bird, C., Bachmann, A., Aune, E., Duffy, J., Bernstein, A., Filkov, V., Devanbu, P. FAIR AND BALANCED? BIAS IN BUG-FIX DATASETS. *ESEC/FSE* 2009

[26] Posnett, D., Filkov, V., Devanbu, P. Ecological inference in empirical software engineering. *ASE* 2011

[27] Devanbu, P., NEW INITIATIVE: THE NATURALNESS OF SOFTWARE *ICSE NIER 2015*

[28] FRANKS, C. , Tu, Z., Devanbu P., and Hellendoorn, V., CACHECA: A CACHE LANGUGE MODEL BASED CODE SUGGESTION TOOL, *ICSE 2015 Demonstration Track*

[29] Casalnuovo, C., Sagae, K., and Devanbu, P. , STUDYING THE DIFFERENCE BETWEEN NATURAL AND PROGRAMMING LANGUAGE CORPORA *Empirical Software Engineering*, 2019

[30] Casalnuovo, C., Devanbu, P., Oliveira, A., Filkov V., Ray, B., ASSERT USE IN GITHUB PROJECTS *ICSE 2015*

[31] Gros, D., Sezhiyan, H., Devanbu, P., Yu, Z., CODE TO COMMENT ”TRANSLATION”: DATA, METRICS, BASELINING & EVALUATION *ASE 2020*

[32] Ding, R., Ray, B., Devanbu, P., and Hellendoorn, V. PATCHING AS TRANSLATION: THE DATA VS. THE METAPHOR *ASE 2020*

[33] Casalnuovo, C., Devanbu P., Morgan, E., DOES SURPRISAL PREDICT CODE COMPREHENSION DIFFICULTY? *Cognitive Science Society Conference* 2020

[34] Liao, J., Yang, G., Kavaler, D., Filkov, V., Devanbu , P. STATUS, IDENTITY, AND LANGUAGE: A STUDY OF ISSUE DISCUSSIONS IN GITHUB *PLOS One Journal* 2019

[35] Kavaler, D., Devanbu, P., Filkov,V., WHOM ARE YOU GOING TO CALL? DETERMINANTS OF @-MENTIONS IN GITHUB DISCUSSIONS *Empirical Software Engineering Journal*, 2019

[36] Zhai, H., Casalnuovo, C., Devanbu, P TEST COVERAGE IN PYTHON PROGRAMS *MSR* 2019