Statement on Research, Teaching and Service

Premkumar Devanbu

1 Research

"Naturalness" of Software. This is a recent, very exciting discovery, with substantial scientific and practical implications. We showed that software has striking similarities to human linguistic behavior, in that it is highly repetitive and *predictable* [1] using models adapted from Natural Language Processing. Indeed, we found that code is *8-16* times *more* predictable than English—this, together with the large mixed corpora of code and English that are widely available, opens up a large realm of research, rife with scientific questions and engineering promise. This work has led to a total of about $2,900,000 funding from the NSF (3 grants, including two EAGERs and one NSF CISE Large: $1,600,000 for UC Davis, shared between Computer Science and Linguistics, with the rest for our partners at CMU and Iowa State). This work has led to a spate of follow-on research and publications from Universities in the US (CMU, U. Mass-Amherst, Wisconsin-Madison, UIUC), Canada (Waterloo, Alberta), UK (Edinburgh, UCL), and Switzerland (ETH), with applications to code porting (Java to C#), coding standards, code idiom mining, etc. The original paper is now listed as the second most cited paper[1] from ICSE 2012. An NSF workshop in this area has been funded, and is being organized by Ben Snyder (a natural languages processing expert from Wisconson-Madison) and a proposal for Dagstuhl workshop is under way, organized by Charles Sutton (NLP person from Edinburgh). We are now engaged in a vigorous (but happy) competition with some very bright people at prestigious institutions, to stay at the forefront of this research area—despite the fact that we started it all!

During this period, this has led to two follow-up papers: one on improved language modeling that exploits local repetitiveness in code [13], as well as a study of repeating bug-fixes in a large corpus [15].

Defects: Prediction & Causes and Remedies. We have continued our empirical studies of defects and defect prediction. We studied the effects of bias on prediction accuracy [7] and found that bias matters less than training sample size. We studied the relative value of different types of metrics, and showed that process metrics have more useful variance, leading to better predictions [9]. We also found that focus, when modeled using an ecological measure relating to "entropy" of focus, explained defect occurrence [8].

We also had a couple of results that challenge existing presumptions in programming languages and software engineering. First, we found that statistical defect prediction (which is largely data-driven, and doesn't require extensive tool development) does essentially just as well static bug finders (which do require extensive language-specific tool development) at helping inspectors locate defects [11].

Second, large datasets available in GitHub permitted a careful study of the relative effects of programming language design choices on software quality. We found that static typing is somewhat better than dynamic typing, strong typing is better than weak typing, and built-in memory management is better; overall however, language designs played only a *minor* role in defect occurrence, when compared to process aspects [10].

[1] `http://arnetminer.org/conferencebestpapers`

New initiatives: We have recently initiated a collaboration with Raul Aranovich in Linguistics, currently funded through an EAGER grant; work has just begun; we hope to use statistical NLP methods to investigate parallels between "registers" in spoken & written language, and programming styles, as well as parallels between adult second-language learners and beginning programmers. In addition, we hope to start a new collaboration, along with Raul, with other faculty in Cognitive- and Neuro-linguistics to use large corpora to evaluate theories of human language behavior, as manifest both in code and natural language.

2 Impact & Recognition

Over the last two years, we have won three awards to support our new research into the Naturalness of Software. We received a $600,000 EAGER grant in 2012 ($300,000 for UC Davis) and more recently a $2,000,000 award (NSF SHF LARGE, $1,000,000 for UC Davis) with our partners CMU and Iowa State; we are the lead on both. SHF CISE LARGE is among the most competitive grants awarded in the CISE division, to my knowledge only one award is given per year. We have also won a $300,000 award from another EAGER program, led by Linguistics professor Raul Aranovich.

Our paper with Filkov, D'Souza & Post-doc Posnett at ICSE 2013 [4] won an *ACM SIGSOFT Distinguished paper award (DPA)*, which is given to about 1% of the submitted papers to the conference. We have also been nominated for a distinguished paper award at ACM SIGSOFT FSE 2014. At this FSE, UC Davis has 6 (!) papers. I believe this is an all-time record for *any* institution at this prestigious and competitive venue.

A few other relevant facts relating to impact:

1. In the *Arnetminer* (`http://arnetminer.org/conferencebestpapers`) listing of the best and most cited papers, out of several conferences in all areas of Computer Science; my papers are listed 7 times.

2. In the Google scholar venue impact listings for software systems (`http://goo.gl/qNp9uZ`) my papers are listed several times: one for ICSE [4], *three* for SIGSOFT FSE [6, 16, 14] (including the most cited paper over the last 5 years at FSE [6]), and one for Empirical Software Engineering Journal [12]

3. Including ICSE 2014, I have published more papers in ICSE than anyone else[2]. After this SIGOSFT FSE 2014, I expect to be be most published author considering the two top software engineering conferences (ICSE and FSE[3]).

3 Teaching

I have taught the senior-level and graduate courses in software engineering, the capstone project course, and the introductory C programming course.

Undergraduate Teaching: ECS 30, ECS 160 & ECS 193A/B The senior software engineering course *ECS 160*, is a capstone project course. More recently, *ECS 193A/B* has emerged as an alternative project course. As we move towards a more rational re-alignment of the two

[2] `http://goo.gl/FiHx7J` see author list on the left, in the middle of the page.

[3] `http://goo.gl/ua6pf8`

capstone courses, I conducted *ECS 193A/B* for the first time. Following the wonderful model set by Prof. Davis in Mechanical Engineering, I substantially revised the course. I greatly expanded the contact hours with students, and had weekly lectures *and* project review meetings with every team. I required students to follow an Agile development process, adopting a strong customer focus. Students were expected to complete project documents, and provide weekly sprint progress reports, co-ordinated with customers.

The C course *ECS 30* has very large enrollments, usually 200 to over 300. Recently, one is faced with the challenge of keeping up instructional quality of difficult concepts, for non-majors, and English-language learners, while facing diminished resources (*e.g.,* TA support). I made use of automated grading, as well as volunteer tutors, and greatly expanded overall office hours to provide more students with one-on-one assistance at scale.

Graduate Teaching I continue to teach the Graduate Software Engineering Course ECS260, covering a mix of advanced programming (functional programming, advanced OO programming, with design patterns, aspect-orientation, etc) and empirical methods in software engineering.

Graduate & Post-Doc Supervision: In 2012, I took on 3 new students, all co-supervised with Filkov: S. Godhane, C. Casalnuevo, and D. Kavaler. In 2013, I recruited two post-docs, Z. Tu and B. Ray (female). Z. Tu has since moved on to a research position at Huawei Labs in Hong Kong. B. Ray will enter the faculty job market this year. I also graduated one M.S. (Project) student, Jui-Chung Wu.

4 Service

Departmental-level: I was on the department recruiting committee, and participated as a graduate advisor.

College & University-level: I served on the University Privilege & Tenure Investigative Committee

4.1 Professional

Journals: Associate Editor for *Springer-Verlag Empirical Software Engineering and Maintenance* and *Wiley Journal of Software Maintenance and Evolution.*

Extra-mural Service & Visibility I co-chaired the programs for the *SIGSOFT FSE 2014 Data & Artifact Tack* and the *Indian Software Engineering Conference 2014*, and served as General Chair for *International Conference on Mining Software Repositories, 2014.*

I served on the program committees of ICSE 2013, 2014, and SIGSOFT FSE 2014. These are the two top conferences in our field.

I keynoted at: *Indian Software Engineering Conference, 2013*, *Microsoft Software Engineering Innovation Fellowship Summit, 2003*, *International Workshop on Software Engineering Research and Industrial Practice 2014*, and the *Dagstuhl Seminar on Software Analytics 2014.*

I was an invited Distinguished Lecturer at: University of Massachusetts, Amherst (2013), University of Nebraska, Lincoln (2014), University of Maryland (B.C.) (2014).

Grant Reviews: NSF Panel service.

References

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

[2] Shroff, G., Agrawal, P., Devanbu, P., INSTANT MULTI-TIER WEB APPLICATIONS WITHOUT TEARS. Indian Software Engineering Conference 2009

[3] Filkov, V., Saul, Z.M., Roy, S., DSouza, R.M., and Devanbu, P. MODELING AND VERIFYING A BROAD ARRAY OF NETWORK PROPERTIES. Euro-Physics Letters, 86 (2009)

[4] Bird, C., Nagappan, N., Devanbu, P., Gall, H., and Murphy, B., DOES DISTRIBUTED DEVELOPMENT AFFECT SOFTWARE QUALITY? AN EMPIRICAL CASE STUDY OF WINDOWS VISTA, Research Highlight, Communications of the Association of Computing Machinery, August, 2009. Revised version of ICSE 2009 paper.

[5] Bird, C., Barr, E., Nash, A., Devanbu. P, Filkov, V., Su, Zhendong, STRUCTURE AND DYNAMICS OF RESEARCH COLLABORATION IN COMPUTER SCIENCE, Proceedings, SIAM International Conference on Data Mining 2009.

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

[7] Rahman, F., Posnett, P., Herraiz, I., Devanbu, P. SAMPLE SIZE VS. BIAS IN DEFECT PREDICTION. ESEC/FSE 2013

[8] Posnett,D., DSouza, R., Devanbu P., Filkov. V., DUAL ECOLOGICAL MEASURES OF FOCUS IN SOFTWARE DEVELOPMENT, ICSE 2013

[9] Rahman, F., and Devanbu, P. HOW AND WHY PROCESS METRICS ARE BETTER. ICSE 2013

[10] Ray, B., Posnett, D., Filkov, V., and Devanbu, P. A LARGE SCALE STUDY OF PROGRAMMING LANGUAGES AND CODE QUALITY IN GITHUB, ACM SIGSOFT FSE 2014.

[11] Rahman, F., Barr., E., Khatri, S., Devanbu, P. COMPARING STATIC BUG FINDERS AND STATISTICAL DEFECT PREDICTION, ICSE 2014.

[12] Rahman F., Bird, C., Devanbu, P., CLONES: WHAT IS THAT SMELL?, *EMPIRICAL SOFTWARE ENGINEERING JOURNAL*, *Springer-Verlag*, 2012.

[13] Tu, Z., Su, Z., Devanbu P., ON THE LOCALNESS OF SOFTWARE, Proceedings, ACM SIGSOFT FSE, 2014.

[14] C. Bird, N. Nagappan, B. Murphy, H. Gall, P. Devanbu. DON'T TOUCH MY CODE! EXAMINING THE EFFECTS OF OWNERSHIP ON SOFTWARE QUALITY (Formerly C4). ACM SIGSOFT ESEC/FSE 2014

[15] Barr,E., Harman, M., Devanbu, P., and Sarro, F., THE PLASTIC SURGERY HYPOTHESIS, Proceedings, SIGSOFT FSE 2014.

[16] Bachmann, A., Bird, C., Rahman F., Devanbu, P., Bernstein, A., THE MISSING LINKS: BUGS AND BUG FIX COMMITS, Proceedings, SIGSOFT FSE 2010