Big(ger) Data in Software Engineering

Data has always been used in every company irrespective of its domain to improve the operational efficiency and the products themselves. However, analyzing and extracting information from "*Big Data*" is the next revolution in technology [1], since previously unknown nuggets of information are now made visible. In fact, over 90% of the data available in the world has been generated in the last two years [2]. "*Big Data*" analytics has become the next hot topic for most companies - from financial institutions to technology companies to service providers. Likewise in software engineering, data collected about the development of software, the operation of the software in the field, and the users feedback on software have been used before. However, collecting and analyzing this information across hundreds of thousands or millions of software projects gives us the unique ability to reason about the ecosystem at large, and software in general. At no time in history has there been easier access to extremely powerful computational resources as it is today, thanks to the advances in cloud computing, both from the technology and business perspectives. Therefore, it is easier today than ever before to analyze big data.

In this technical briefing, we will present the state-of-the-art with respect to the research carried out in the area of big data analytics in software engineering research. We will present the research along three dimensions:

1) *What are the software engineering problems being solved?* Examples of problems include: How much source code is newly written and how much is reused from past projects? Can we recommend best practices to developers by observing the development of software among hundreds of thousands of software projects?
2) *What are the datasets that are being used?* Examples of my datasets include: all the mobile apps in the Google Play store, all of the world's Open Source projects, and hundreds of gigabytes of execution logs. Such large datasets provide us with a unique view into the SE field.
3) *What are the tools and techniques available to analyze the large datasets?* We intend to present generic software solutions that have been applied to big datasets in other areas of research, and the tools and techniques created by software engineering researchers.

In the end we will present the challenges inherently present in large datasets - volume, variety, velocity, and veracity. Such challenges often complicate the analysis of the data and can invalidate the interpretation of the results. We will conclude with the future opportunities that are present in big data analytics for software engineering research.

Relevance to Software Engineering Community

Software Engineering (SE) research over the past few decades has identified and attempted to solve a variety of issues pertaining to the development of software. Such studies are able to shed light on current development, and maintenance practices of software engineers. However, these studies often face the threat of Generalizability since only a limited set of projects has been examined. Therefore, the conclusions of these studies may only be applicable to a limited set of software projects. Hence even though studies on a small set of cases are important in any research area, complementary studies that look at entire ecosystems of software projects are needed as well.

This technical briefing will offer an opportunity to become familiar with the non-traditional techniques for mining software repositories. The outcomes of the technical briefing are expected to be an improved understanding of the current-state-of-the-art in the problems solved, datasets used, tools and techniques available, and future challenges in the area of *Big Data Analytics for Software Engineering*.

General Discussions

This technical briefing will be highly interactive in order to encourage discussion between participants. The session stars with an introduction to the notion of big(er) data in SE, then this is followed by brief

presentation of state of art and practice on this topic from three complimentary perspectives of (i) software engineering problems addressed (ii) data characteristics, accessibility and examples (iii) Tools and techniques previously developed and use. Presentation of these topics will be structured to provide plenty of time for discussion. It will feature “one minute madness” talks from participants to share their experience and initial thoughts on the topic. The idea is that the participants will be given one minute to talk for the audience about their interests, challenges or questions. We have already used this format at "Mining New Patterns" technical briefing at PLoP 2014 and several workshops TwinPeaks@ICSE 2014, TwinPeaks@ICSE 2013 and TwinPeaks@RE13 and it was very well received by participants. Finally we will host an activity in which participants will work collaboratively on designing experiments for the bigdata examples provided as part of technical briefing.

Participant Selection

The target audience includes researchers interested in data driven approach to software engineering and scientists who are developing/using methods, techniques and tools to mine ultra large scale software repositories. We also target practitioners who face the challenge of finding empirically validated solutions for the software engineering problems they are addressing. Furthermore, our target audience includes SE educators interested in developing course materials in this area. No specific background will be required, participation in the technical briefing will be open to all ICSE 2015 participants.

References

[1] http://www.mckinsey.com/insights/business_technology/big_data_the_next_frontier_for_innovation
[2] http://www.sciencedaily.com/releases/2013/05/130522085217.htm

Bios for the Presenters

Meiyappan Nagappan is an Assistant Professor in the Software Engineering Department of Rochester Institute of Technology. His research is centered on the use of large-scale Software Engineering (SE) data to address the concerns of the various stakeholders (e.g., developers, operators, and managers). He received a PhD in computer science from North Carolina State University. Previously he was a postdoctoral fellow in the Software Analysis and Intelligence Lab (SAIL) at Queen's University, Canada.
Dr. Nagappan has published extensively in various top SE venues such as TSE, FSE, EMSE, and IEEE Software. He has also received a best paper award at the International Working Conference on Mining Software Repositories (MSR 12). He continues to collaborate with both industrial and academic researchers from the US, Canada, Japan, Germany, Chile, and India. You can find more about him at http://sailhome.cs.queensu.ca/~mei/.

Mehdi Mirakhorli is an assistant professor at Rochester Institute of Technology. His research interest includes empirical software engineering with an emphasize on the application of large scale data mining and information retrieval techniques to solve software architecture design problems. He has conducted several research projects in the area of mining large scale software repositories to build knowledge discovery techniques, recommender systems and advanced architecture traceability methods. Previously, he worked for 7 years as a software architect on large data-intensive software systems in banking, health care and meteorological domains. He has co-organized a technical briefing on a similar topic of “Discovering new patterns by mining large scale code repositories” at Pattern Languages of Programs Conference (PLoP 2014). Mehdi has served as Guest Editor for a special edition of IEEE Software and organizer, committee member and reviewer for several software engineering workshops, conferences and journals. Furthermore he has spoken in several technical venues, and served as ALTA Distinguished Speaker at Alcatel-Lucent. Mehdi has received two ACM SIGSOFT Distinguished Paper Awards at the International Conference on Software Engineering and has been actively engaged in data-intensive research projects with the US Department of Homeland Security (DHS).