

Singapore Management University

## Institutional Knowledge at Singapore Management University

---

Research Collection School Of Computing and Information Systems

School of Computing and Information Systems

---

1-2014

### Inferring the untold: Mining software engineering research publication networks

Santonu SARKAR  
*InfoSys Labs*

Subhajit DATTA  
*Singapore Management University, subhajitd@smu.edu.sg*

Follow this and additional works at: [https://ink.library.smu.edu.sg/sis\\_research](https://ink.library.smu.edu.sg/sis_research)



Part of the [Databases and Information Systems Commons](#), and the [Software Engineering Commons](#)

---

#### Citation

SARKAR, Santonu and DATTA, Subhajit. Inferring the untold: Mining software engineering research publication networks. (2014). *Infosys Lab Briefings*. 12, (1), 88-95.

Available at: [https://ink.library.smu.edu.sg/sis\\_research/6054](https://ink.library.smu.edu.sg/sis_research/6054)

This Journal Article is brought to you for free and open access by the School of Computing and Information Systems at Institutional Knowledge at Singapore Management University. It has been accepted for inclusion in Research Collection School Of Computing and Information Systems by an authorized administrator of Institutional Knowledge at Singapore Management University. For more information, please email [cherylds@smu.edu.sg](mailto:cherylds@smu.edu.sg).

# Inferring the Untold – Mining Software Engineering Research Publication Networks

*By Santonu Sarkar PhD and Subhajit Datta PhD*

*Do researchers collaborate in the  
software engineering ecosystem?*

Since the inception of organized research publication in software engineering in 1975, the discipline has gained maturity. This journey has been guided by the synergy of ideas and interactions of individuals. In this paper, we discuss a method for aggregating the corpus of 19,000+ papers and 21,000+ authors across 16 specialized software engineering venues. We focus on the approach of data collection, processing and storage. It can be used to address questions by the software engineering research community. We evaluate three questions: patterns of research topics with time, factors influencing the contribution of individual researchers, and the interaction among the most prolific researchers. Furthermore, we provide a brief overview of data analysis techniques that we performed to address questions. We also highlight how the outcome of the research questions can be made available as an online service.

## INTRODUCTION

The phrase Software Engineering (SE) was used for the first time at a NATO conference in 1968. In 1975, the first dedicated venue for publishing software engineering research was introduced– The IEEE Transactions on Software Engineering (TSE). Ever since, software engineering research has gained in scope and impact. It has coincided with the pervasiveness of software artifacts in our lives. It calls for a software engineering research collaboration network of researchers and practitioners for several reasons.

Firstly, in spite of the pervasive use of software, there is little consensus on the nature of software engineering. Parnas rued the lack of consummation in the marriage between software and engineering [1]; whether software will ever be an engineering discipline remains a cause of concern [2]; how SE relates to other computing disciplines is still confusing [3]; and why we do not learn adequately from our lessons continues to worry us. It indicates

that the identity of software engineering as a scientific discipline is still in limbo. To a large extent, a discipline's identity grows out of its research. So learning how researchers collaborate is of critical importance.

Secondly, there is a perception that software engineering is driven by trends (<http://www.semat.org>). Paradigms seem to arrive and depart in quick succession, and the next big thing is always said to be round the corner. Large-scale empirical scrutiny of such perceptions is imperative to establish the character of SE.

Thirdly, empirical research is significantly more collaborative than theoretical research. In mathematics and theoretical sciences, the number of joint authors for a paper is usually low, whereas in empirical sciences there are several authors working on a paper. Software engineering emerged as a sub-discipline of computer science - which has distinct mathematical roots - but SE research has become increasingly empirical [4]. How the most highly contributing SE researchers (the 'prolifics') collaborate and whether collaboration facilitates research reflects on research in SE.

In our paper, we present our ongoing work on the Microsoft method of construction and analysis of SE publication networks. We first describe the outline of the dataset, data collection method, and the creation of the model based on paper and co-authorship network. We then describe various possibilities of analysis involving people, articles and venues.

## DATA COLLECTION

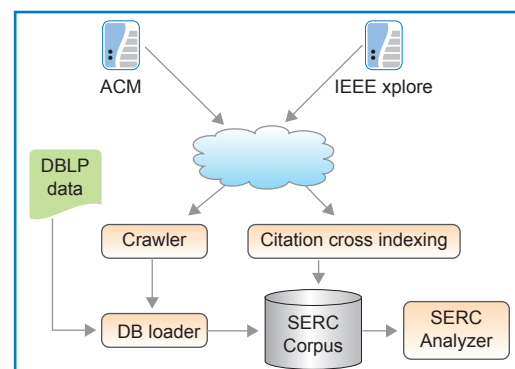
To address the research questions, we need some metrics that reflect on the parameters of our interest. For measuring the contribution of a researcher, we consider two basic measures - publication count and citation count. We assume that the former reflects the amount

of research published by a researcher, while the latter indicates the extent to which the researcher's work has been recognized.

These two measures provide distinct yet complementary points of view - the quantitative and qualitative aspects of research. We recognize that 'counting' the number of papers and citations as a way of measuring a researcher's contribution is not without controversy [5]. However, since the evaluation of academic and industrial research continues to rely on these measures, we believe that our approach is aligned with the status quo. So publication and citation counts are our measures of researcher's 'success.'

In this paper, terms/phrases such as success, a high level of contribution, and productivity are used interchangeably. We also use 'researcher' and 'author' as well as 'paper' and 'publication' interchangeably. To identify interactions among highly successful researchers, we consider their co-authorship information. Even though researchers can interact in other ways - for example, being friends or serving on the same committee - we assume that their interaction during research results in co-authored papers.

We have implemented a tool (Figure 1) that illustrates our approach to represent the SERC framework. We collected data published in the list of venues (Table 1).



**Figure 1:** Schematic diagram of SERC system  
**Source:** Industry-academia research

Venue	Main & companion conference	No. of papers	Year of inception
TSE	IEEE Transactions on Software Engineering	3,000	1975
ICSE	International Conference on Software Engineering, ICSE Companion, AST, FOSE, ICSE Workshop, SEAMS, SESS, SHARK, STRAW, WADS, and Software Education	3,338	1976
JSS	Journal of Systems and Software	2,895	1979
SOFTWARE	IEEE Software	2,541	1984
OOPSLA	Object-Oriented Programming, Systems, Languages & Applications	1,823	1986
ECOOP	European Conference on Object-Oriented Programming, FICS, RAM-SE and WRT	692	1987
ISSTA	International Symposium on Software Testing & Analysis, DEFECTS, PADTAD, PDATAD, ROSATEA, Random Testing, Symposium on Testing & Verification, TAV-WEB, WODA, and WTAOP	481	1989
KBSE	Knowledge-based Software Engineering Conference	1,062	1991
TOSEM	ACM Transactions on Software Engineering & Methodology	267	1992
FASE	International conference on Fundamental Approaches to Software Engineering	443	1993
ASE	International conference on Automated Software Engineering	305	1995
APSEC	Asia-Pacific Software Engineering Conference	1,083	1995
FASE	Intl. Conference on Fundamental Approaches to Software Engineering	396	1998
WICSA	Working Conference on Software Architecture, SOAR and ECSA	342	1999
ISSRE	International Symposium on Software Reliability Engineering	427	2000
CBSE	Component-based Software Engineering symposium	160	2004
<b>Total number of papers - 19,255</b>			
<b>Total number of authors - 21,282</b>			

**Table 1:** Publication venues and details

*Source:* Industry-academia research

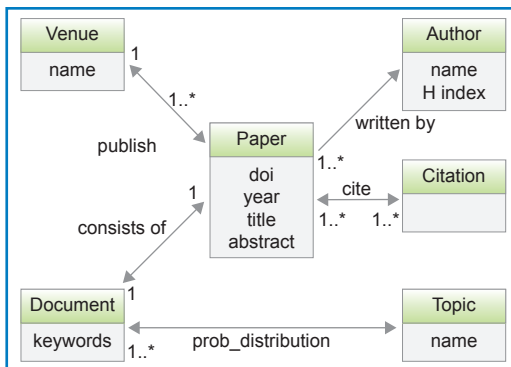
Information of papers published in these venues is available at DBLP [8]. Citation cross indexing module (Figure 3) builds a citation cross reference database between papers in SERC using publicly available information from ACM Digital Library and IEEE Xplore [9, 10].

Once cross reference for all papers whose citation information can be accessed is constructed, the citation count for authors is computed. Paper abstracts were extracted from bibliographic repositories. We implemented specialized web crawlers to search each source and store the data in a MySQL database [11]. A set of Java-based components was developed

to further process and analyze the data. SPSS Statistics 18 was used for statistical analysis and some diagrams were generated using Microsoft Excel.

#### ADDITIONAL DATA

DBLP-generated data does not suffice for our analysis. To understand various topics of publication and how they evolve, how highly contributing researchers collaborate, and the importance of a particular paper or a topic, it is imperative to obtain additional information such as the abstract of the paper, citation details and the author's H-index information.



**Figure 2:** SERC metamodel  
**Source:** Industry-academia research

For the abstract of the software engineering paper, the special purpose crawler collects details from different sites (Figure 1) and creates additional database tables. Each paper has a unique internal id generated by DBLP. From the crawled data, we take the paper title and search in the DBLP table to obtain the unique id and establish referential integrity with DBLP data. Citation data collection for each paper happens in the following manner:

1. Use the cited paper title to crawl ACM and <http://academic.research.microsoft.com/> for the citation link of the paper. The link is first stored in a file.
2. Next, for each link corresponding to the citing paper, obtain the paper title.
3. Store the cited paper title, id and the paper title in a citation database table.

### STRUCTURE OF SERC

The metamodel for SERC is illustrated in Figure 2. The metamodel  $SERC = \langle V, P, A, Cref, I \rangle$  has five main elements. The central element is the set  $P$  of papers related to software engineering

published in different venues from 1975 till 2010, having attributes such as the *year of publication*, *paper title*, *doi* and so on as per the DBLP schema [9]. We have introduced an additional attribute called *abstract* that contains the abstract of the paper. The next element is the set software engineering publication venues denoted by  $V$  (Table 1).  $A$  denotes the set of authors of these papers. Like  $P$ , attributes of the author are defined in the DBLP schema. In addition, we define a new attribute called *H-index* for an author. As shown in the metamodel, one paper can be written by one or more authors.

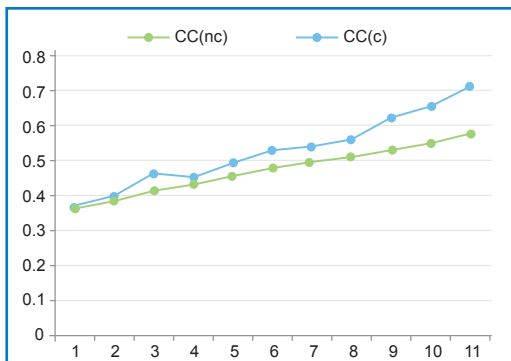
The next relation of the metamodel, the relation  $Cref \subseteq P \times P$  captures citation information between publications. Finally,  $\Gamma$  denotes the set of topics represented as a set of probability distributions over documents. A document is created from a paper by selecting a set of meaningful keywords from its *title* and *abstract*.

### SERC NETWORK MODEL

To address some aspects of our research questions, we constructed a co-authorship network  $N_{SERC} = \langle A, E \rangle$  from SERC, where vertices (nodes) are the authors ( $A$ ) and two vertices are connected by an undirected link ( $e = (a_i, a_j) \in E$ ) if the two corresponding authors  $a_i$  and  $a_j$  have co-authored at least one paper ( $p \in P$ ) in a given period of time. Based on SERC, we address research questions:

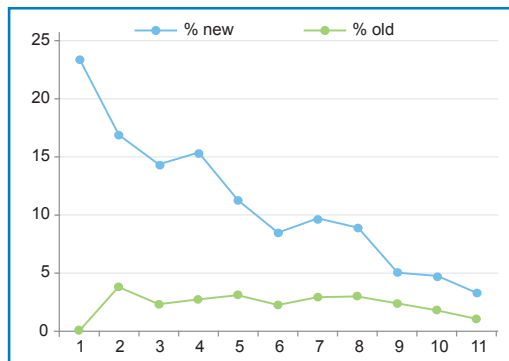
#### RQ-01: HAS RESEARCH COLLABORATION INCREASED WITH TIME?

There is a perception that the extent of research collaboration in various disciplines has increased in the last few decades. Several reasons are attributed: better facilities for storage, access and sharing of research results through digital libraries, major enhancements in electronic collaborative media, etc. Whether researchers are collaborating



**Figure 3:** Clustering coefficient: Cumulative and non-cumulative time-steps

Source: Industry-academia research



**Figure 4:** Percentage of new and old singletons across time-steps

Source: Industry-academia research

more than they did in the past is a moot question in the development of any discipline. In our first research question, we examine whether collaborative research in software engineering has increased over time. Let us evaluate collaboration characteristics from several perspectives.

It is observed in networks of individuals that two vertices linked to a third are more likely to be themselves linked. Intuitively, two of one's friends have a higher probability of being friends themselves. It is measured by the *Clustering Coefficient (CC)*.

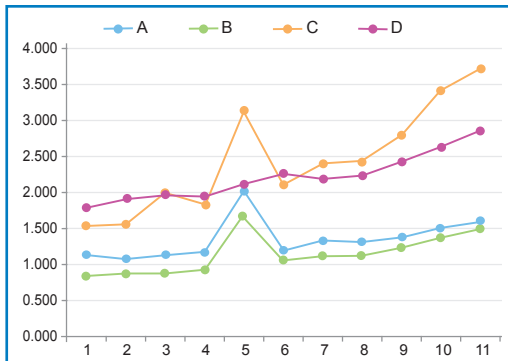
For research collaboration network based on co-authorship of papers, CC represents the probability that any two authors  $a_i$  and  $a_j$ , represented by two vertices, co-authored a paper if both of them *independently* co-authored paper(s) with a third author  $a_k$ . For a vertex  $a$  (which represents an author in our case) with a degree  $D_a$ , there are  $D_a$  neighbors of  $a$ . If all of these  $d_a$  neighbors are linked, there would be  $(D_a(D_a-1))/2$  links between them. Let  $E_a$  be the *actual* number of links between them. Then the clustering coefficient  $CC_a$  of node  $a$  is defined as the ratio of the actual number of links and the maximum possible number of links:

$$CC_a = (2 \cdot E_a) / (D_a \cdot (D_a - 1))$$

For the entire  $N_{SERC}$  network, the clustering coefficient  $CC(N)$  is the average of  $CC_a$  across all vertices [6].

With reference to Figure 3, we observe that the average clustering coefficient for  $N_{SERC}$  increases (almost monotonically) across both cumulative and non-cumulative time-steps. Increasing clustering coefficient for SE research signifies that it is more likely for two researchers to collaborate directly, if they have both collaborated separately with a common researcher. It is in complete contrast to trends in other disciplines such as mathematics and neuroscience, which have a decreasing clustering coefficient over time [7]. In our context, collaboration is manifested in the co-authoring of papers. Let us scrutinize co-authorship information in more detail.

Among papers in our dataset, 30% had a single author, 32% had two authors, 21% had three authors, 10% had four authors, 4% had five authors, and 3% had six or more authors. Single-authored papers represent zero collaboration; they do not add any edges to  $N_{SERC}$  in the time-step in which they are published. We refer to the authors who have written only single-authored paper(s) in a particular time-step as *singletons*.



**Figure 5:** Variation of average papers per author (A), average multi-author papers per author (B), average co-authors per author (C), and average authors per paper (D) across time-steps.

**Source:** Industry-academia research

A singleton who publishes a paper in a time-step without ever having published a paper in the past is a 'new' singleton, and a singleton publishing a paper in a time-step who has also published in earlier time-step(s) is an 'old' singleton. With reference to Figure 4, the percentage of new singletons from the pool of authors in each time-step has decreased from 23% to 3.35%, while the percentage of old singletons from the pool of authors has remained more or less static - around 3%. We also observed that the percentage of single-author papers from the total number of papers in each time-step has decreased (though non-monotonically) from 47% to 20%. In addition, with reference to Figure 5, the average number of papers per author, average number of multi-author papers per author, and average number of co-authors per author, all increased with time, with a sharp peak at time-step 5 (1989-1991); the average number of authors per paper also increased with time, without sharp peaks.

The results indicate increasing collaboration in software engineering research over time. The outcome may have

been facilitated by enhanced storage and communication media. However, we posit that the very nature of the software engineering discipline has also influenced the outcome. It has been suggested that as software engineering has progressively become more aligned with conventional engineering principles, it has also gravitated toward a more empirical pursuit [8]. The ubiquity of the Internet has made software systems increasingly network-centric, distributed and ultra large. Consequently, a majority of software engineering research revolves around the design, development and maintenance of software systems. These subjects are often studied through joint programs between academia and industry, facilitating better collaboration.

In summary, empirical evidence indicates that the extent of collaboration in software engineering research has increased from 1975 to 2010. Let us examine how individual researchers collaborate with one another.

RQ-02: Do similar researchers collaborate more? As the body of research literature grows in a discipline, researchers discover more opportunities for collaboration with existing authors. With time, authors develop profiles based on the number of papers, number of co-authors, number of venues, the span of publishing (in years), and topics of research interest. In our second research question, we examine the factors that influence authors to collaborate with one another.

Similarity between researchers can be perceived at many levels. Let us begin with a simple way to measure similarity: we consider two authors to be similar if both of them have written similar number of papers, or have similar number of co-authors, or have published in similar number of venues, or have published

in a similar time span. We will refine this naive notion of similarity in a subsequent discussion.

Although co-authorship networks are casually referred as 'social networks,' Newman and Park pointed out that social networks differ from other networks in two ways: i) non-trivial clustering (or *network transitivity*), and ii) positive correlations between the degrees of adjacent vertices (or *assortative mixing*) [7]. In the discussion of the first research question, we saw how  $CC(N)$  for  $N_{SERC}$  network shows an increasing trend. The other distinctive feature of social networks, *assortative mixing* reflects the tendency of higher degree vertices connecting with other higher degree vertices (and vice versa). Our experience relates to this observation: a gregarious person is more likely to have gregarious friends. In  $N_{SERC}$  assortative mixing should translate to highly connected authors collaborating with other highly connected authors.

To understand if similar authors collaborate more in software engineering research, we first tested whether assortative mixing holds in  $N_{SERC}$ . As recommended in [7], we calculated the Pearson correlation coefficient between the degrees of vertices at the ends of each of the 25,511 edges of  $N_{SERC}$ . The resulting value is 0.28. It suggests a low positive correlation between the numbers of co-authors of two collaborating authors, and cannot be taken as reasonable evidence of degree assortativity in the network. We were curious to find out the levels of correlation between the other attributes of collaborating authors. For each pair of authors at two ends of the 25,511 edges in our network, we observed the Pearson correlation coefficients to be 0.23 for the number of papers, 0.35 for the number of venues, and 0.23 for the timespan of publishing (in years). The values indicate a mildly positive correlation at best, and do not

indicate that authors with similar publication profiles collaborate appreciably in software engineering research.

## FURTHER ANALYSIS

Our insights are based on the network analysis of SE publication data. In addition, fascinating insights can be derived from this infrastructure. Based on our ongoing work and from this dataset, we can seek answers to these questions:

- **Genesis of software engineering research**  
In software engineering, research and practice go hand-in-hand. Given how software artifacts have influenced our lives, it is worthwhile to investigate the genesis of software engineering research. We found evidence that the extent of collaboration in software engineering has increased from 1976 to 2010.
- **Factors influencing researcher productivity**  
Software engineering has attracted researchers from diverse backgrounds and interests. Some researchers have been very successful in terms of publication volume as well as the number of citations received. Exploring factors that influence the productivity of individual researchers can be facilitated by the SERC infrastructure.
- **Collaboration profiles between researchers**  
The nature and direction of research in a discipline depends to a large extent on how researchers collaborate among themselves. An empirical understanding of what encourages researchers to collaborate can go a long way in fostering collaboration across individuals and organizations.




The SERC framework offers a test bed for undertaking such analysis.

## CONCLUSION

The advent of web-based archival systems for scientific publications has stoked interest in the ecosystem of scientific research. While other fields within and beyond computer science have made significant advances, there has not been a notable effort in understanding aspects of software engineering research. In this paper, we studied the characteristics of research collaboration in software engineering using publication data from 1975 to 2010. We found empirical evidence that the extent of collaboration has increased with time, authors with similar publication backgrounds do not collaborate with one another appreciably, and when a research topic attracts more researchers there is lesser collaboration. We also found evidence that there are distinct phases within our measurement period, and a simple model can predict whether two researchers will collaborate in the future. The results of this paper can serve as a foundation for a better understanding of the software engineering research ecosystem.

## REFERENCES

1. Parnas, D., L. (1997), *Software Engineering: an Unconsummated Marriage*, In the communication of the ACM, 40 (9), pp. 128.
2. Davis, M. (2011), *Will software engineering ever be engineering?* In the communication of the ACM 54 (11), pp. 32-34.
3. Zelkowitz, M. V. (2012). *What have we learned about software engineering?* In the communication of the ACM 55 (2), pp. 38-39.
4. Shaw, M. (2009), *Continuing Prospects for an Engineering Discipline of Software*, In IEEE Software, 2009.
5. Parnas, D., L. (2007), *Stop the numbers game*, In the communication of the ACM, 50 (11), pp. 19-21.
6. Barabasi, A., Jeong, H., Neda, Z., Ravasz, E., Schubert, A. and Vicsek, T. (2002), *Evolution of the social network of scientific collaborations*, In Elseviers' Physica A 311, 2001, pp. 590-614.
7. Newman, M. and Park, J. (2003), *Why social networks are different from other types of networks*, In Phys. Rev. E 68, 036122.
8. DBLP, Available at <http://www.informatik.uni-trier.de/~ley/db/>.
9. ACM Digital Library. Available at <http://dl.acm.org>.
10. IEEE Digital Library. Available at <http://ieeexplore.ieee.org>.
11. Microsoft Academic Research. Available at <http://academic.research.microsoft.com>. 

## *Author's Profile*

---

SANTONU SARKAR PhD is leading next generation computing research at Infosys Labs. He can be reached at [Santonu\\_Sarkar01@infosys.com](mailto:Santonu_Sarkar01@infosys.com).

SUBHAJIT DATTA PhD is a teaching faculty at the Singapore University of Technology and Design (SUTD). He can be contacted at [subhajit.datta@acm.org](mailto:subhajit.datta@acm.org).

*For information on obtaining additional copies, reprinting or translating articles, and all other correspondence, please contact:*

Email: [InfosyslabsBriefings@infosys.com](mailto:InfosyslabsBriefings@infosys.com)

© Infosys Limited, 2014

Infosys acknowledges the proprietary rights of the trademarks and product names of the other companies mentioned in this issue of Infosys Labs Briefings. The information provided in this document is intended for the sole use of the recipient and for educational purposes only. Infosys makes no express or implied warranties relating to the information contained in this document or to any derived results obtained by the recipient from the use of the information in the document. Infosys further does not guarantee the sequence, timeliness, accuracy or completeness of the information and will not be liable in any way to the recipient for any delays, inaccuracies, errors in, or omissions of, any of the information or in the transmission thereof, or for any damages arising there from. Opinions and forecasts constitute our judgment at the time of release and are subject to change without notice. This document does not contain information provided to us in confidence by our clients.

**Infosys**<sup>®</sup>

POWERED BY INTELLECT  
DRIVEN BY VALUES