Software Engineering Evolution - The History Told by ICSE

Bruno L. Sousa
Computer Science Department
Federal University of Minas Gerais
Belo Horizonte, Minas Gerais, Brazil
bruno.luan.sousa@dcc.ufmg.br

Mívian M. Ferreira
Computer Science Department
Federal University of Minas Gerais
Belo Horizonte, Minas Gerais, Brazil
mivian.ferreira@dcc.ufmg.br

Kecia A. M. Ferreira
Department of Computing
Federal Center for Technological
Education of Minas Gerais
Belo Horizonte, Minas Gerais, Brazil
kecia@cefetmg.br

Mariza A. S. Bigonha
Computer Science Department
Federal University of Minas Gerais
Belo Horizonte, Minas Gerais, Brazil
mariza@dcc.ufmg.br

ABSTRACT

Software Engineering knowledge has continuously grown at a very high speed. In this work, we aim to provide a perspective on how the body of knowledge of Software Engineering has evolved. For this purpose, we analyzed data of 3,300 works published from 1988 to 2018 in one of the most important conferences on Software Engineering, the International Conference on Software Engineering (ICSE). We identified the main topics investigated in Software Engineering and how the investigation of those topics has evolved over the time. The results bring a compilation of Software Engineering evolution that may be of value to the software community.

KEYWORDS

systematic literature mapping, software engineering history, body of knowledge

ACM Reference Format:

Bruno L. Sousa, Mívian M. Ferreira, Kecia A. M. Ferreira, and Mariza A. S. Bigonha. 2019. Software Engineering Evolution - The History Told by ICSE. In *XXXIII Brazilian Symposium on Software Engineering (SBES 2019), September 23–27, 2019, Salvador, Brazil.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3350768.3350794

1 INTRODUCTION

In the last decade, some initiatives to compile the knowledge of Software Engineering and its evolution have been done [2, 3, 5]. An important contribution in this context is the Guide to the Software Engineering Body of Knowledge (SWEBOK), published by the IEEE Computer Society in 2004 [7]. Nevertheless, as the SWEBOK's authors highlight, it is a guide to the body of knowledge of Software Engineering, but it does not present the entire body of knowledge of the area. The document was revised and updated in 2014 [8].

SBES 2019, September 23–27, 2019, Salvador, Brazil

ACM ISBN 978-1-4503-7651-8/19/09…$15.00
https://doi.org/10.1145/3350768.3350794

We have carried out a research on the Software Engineering evolution. The present paper aims to bring an analysis of the evolution of Software Engineering, specially by identifying the key areas considered by researchers over the time.

In this paper, we focus in works published in the International Conference on Software Engineering (ICSE) due to the following main reasons: (i) in contrast with other conferences on Software Engineering, ICSE covers all the topics in the area; (ii) the history of ICSE and of Software Engineering are intertwined; (iii) and despite the importance of many other conferences, including their publications in our analysis may introduce bias in the results because many of them have specific scope. We analyzed the whole set of papers published in ICSE and available in IEEE Digital Library, i.e., from 1988 to 2018. In the total, we analyzed data of 3,300 works.

2 METHOD

To address the goal of our study, we defined three research questions (RQn), as follows.

RQ1: What are the top topics that have been studied in Software Engineering?
RQ2: How has the community interest in the top topics evolved over time?
RQ3: What are the contributions produced in Software Engineering?

We have investigated other three research questions based on the data analyzed in this work. However, due to limit space, we do not present them in this paper.

We carried out this research as a systematic literature mapping. In this section, we describe the planning of the research, involving the data source we used, as well as the inclusion and exclusion criteria. We also describe the execution of the research, that involves the search process, the selection of the documents, and the data processing.

2.1 Planning

2.1.1 Electronic Database. The electronic database chosen for collecting the primary studies was *IEEE Xplore*[1]. We chose *IEEE Xplore* to collect the primary studies because most of ICSE proceedings

[1] https://ieeexplore.ieee.org/Xplore/home.jsp

Table 1: Inclusion and exclusion criteria.

Inclusion Criteria
Papers published in English.
Complete, short papers, or research in progress.
Papers published only in ICSE.
Papers available in electronic format.
Exclusion Criteria
Tutorials, panels, lectures, or keynote talk.
Proceedings or round tables.
Call for workshops or symposium.
Presentation of sessions or tracks.

are indexed and organized in it. Moreover, the papers' metadata provided by *IEEE Xplore* is more detailed than other digital libraries.

2.1.2 Inclusion and Exclusion Criteria. We excluded documents that may not be considered research, such as proceedings, keynote talk, call for workshops, among others. Table 1 summarizes the inclusion and exclusion criteria defined for this study.

2.2 Execution

The execution phase consisted of collecting the ICSE papers via a web crawler and applying the inclusion and exclusion criteria to remove documents that are not relevant for this study. Moreover, after filtering the primary studies, we performed a data processing step to (1) recover keywords not returned during the search process, (2) analyze and standardize the papers keywords to remove synonyms, (3) and classify the results of the papers.

2.2.1 Search Process. We obtained 3,300 documents in this first step. We could not collect papers published before 1988 because they are not indexed into *IEEE Xplore*. The same problem occurred with documents from 2006 and 2014.

2.2.2 Selection Process. This step consisted of manual analysis of the title and the abstract of the documents to remove those that are not scientific research, by applying the inclusion and the exclusion criteria. In some cases, we had to read the introduction and conclusion of some studies to apply the inclusion and exclusion criteria. This step removed 549 documents, resulting in a total of 2,751 primary studies.

2.2.3 Data Processing. The data processing consisted of analyzing by manual inspection the selected studies at the selection process to: (i) recover authors' keywords not returned during the search process, (ii) standardize the authors' keywords of the documents, and (iii) classify the results of the papers.

Keyword Recovery. We identified a lot of papers, about 48%, without keywords attributed by their authors. This problem happened because these keywords were not indexed into *IEEE Xplore*. As this information is essential to our analysis, we performed a keyword recovery step to reduce the rate of papers without author keyword in the data.

This step consisted of a manual inspection of the primary studies to identify the keywords defined by their authors. In this step, we considered only those primary studies without authors keyword. Therefore, we opened the *PDF* files of the papers to verify if the authors defined keywords in these documents. This step was carried out by the first and the second authors of this paper and discussed among the four authors. At the end of this step, we reduced the rate of papers without authors keywords from 48% to 31%. In summary, we recovered keywords from 555 articles, resulting in a total of 1,707 documents with author keywords.

Keyword Standardization. After performing the authors' keyword recovery, we carried out a keyword standardization step to remove some inconsistencies in the data, such as synonymous, plural terms, terms incorrectly typed, among others. With this step, we have a real dimension of the number of papers corresponding to each keyword.

For this purpose, initially, we implemented a routine that counts the number of occurrences of each keyword in our metadata. After running this script, the found terms and their respective occurrences in our metadata were exported to a *CSV* file. This *CSV* was analyzed to identify the inconsistencies and to standardize synonyms and related keywords in a single keyword. We found a total of 2,523 keywords.

Papers Results Classification. To find the top kind of results produced by Software Engineering community, we carried out a paper results classification step. This step consisted of attributing a result category to each study by reading its abstract. The categories used to classify the primary studies results were defined by Mary Shaw [6] and they are reported in Table 2. This classification step was done by the first and the second authors of this present work. The results were discussed among the four authors to mitigate threats to this study.

3 RESULTS

In this section, we present the results regarding to the research questions investigated in this work.

RQ1: What are the top topics that have been studied in Software Engineering?

Table 3 presents the most used keywords in ICSE documents and the number of their occurrences. We show 11 keywords instead of 10 because one of them is *software engineering*, that is too generic. Together, the occurrences of these keywords represent 17.2% of all keywords occurrences. Although *test* stands out from the other keywords due to its recurrence, the high number of keywords and the way they are distributed show that the software engineering community is very diverse and develops works in several subjects. This fact indicates the community is not concentrated in a small set of problems.

RQ2: How has the community interest in the top topics evolved over time?

With RQ2, we aim to depict a timeline for Software Engineering life from 1988 to 2018 that corresponds to the data we gathered from ICSE publications. To answer this question, we considered the top-10 keywords identified in RQ1. The graphic of Figure 1 shows the evolution of the occurrence of such keywords. One of them is *software engineering*, that is too generic. For this reason, we do not discuss it.

The results reveal that a few studies on *test* emerged in 1992, and since 1996, *test* have been the most investigated topic.

Table 2: Categories used to classify the papers results [6].

Type of Results	Description
Procedure or technique	New or better way to do some task, such as design, implementation, maintenance, measurement, evaluation, selection from alternatives; includes techniques for implementation, representation, management, and analysis; a technique must be operational, not advice or guidelines.
Qualitative or descriptive model	Structure or taxonomy for a problem area; architectural style, framework, or design pattern; non-formal analysis domain, well-grounded checklists, well-argued informal generalizations, guidance for integrating other results, well-organized interesting observations.
Empirical model	Empirical predictive model based on observed data.
Analytic model	A structural model that allows formal analysis or automatic manipulation.
Tool	Tool implemented that incorporates a technique; formal language to support a technique or model (should be a calculus, semantics, or another basis for computing or making inference).
Specific solution	Solution to application problems that shows the application of Software Engineering principles. Maybe design, prototype, or complete implementation.
Empirical study	Careful analysis of a system or its development; results of a specific investigation; evaluation or comparison.
Report	Interesting observations, rules of thumb, but not sufficiently general or systematic to rise to the level of a descriptive model.

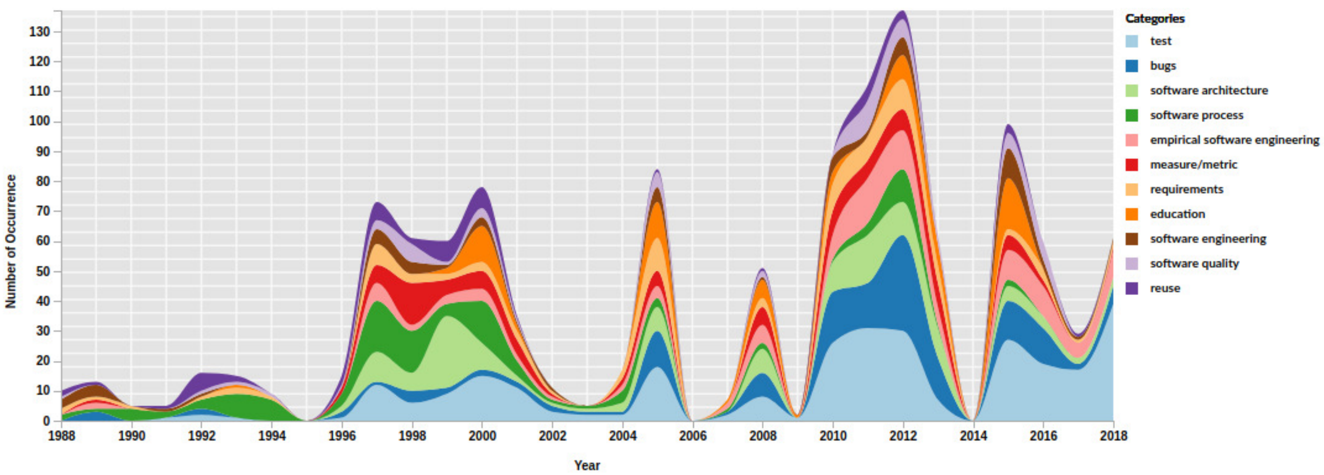

Figure 1: Community interest in the top topics over the time.

Software metrics were timidly studied in 1989, then the attention of research in metric appeared again in 1996, when *software architecture* started to be investigated too. *Software metric* was the main topic in 1998 and *software architecture* had central attention in 1999 and 2000. Since then, these themes have been widely discussed.

Bugs have always been the subject of research and since 2005 is the second most investigated topic. Studies on *requirement* have always been carried out.

A few studies in *education* appeared in 1993, reemerged in 1999 and have been investigated since then. *Empirical studies* and research on *software quality* have always been carried out but started to increase in 1997.

Software process was the most investigated topic from 1993 to 1995; the theme continued to be investigated until 2004 and, then, research in the area reduced a lot. Investigations on *software process* reappeared in large number in 2011, but have decreased again.

The apex of investigation on *reuse* was 1992. The topic was mainly investigated until 2001, reemerged in 2011 when the number of works in *reuse* started to decrease again.

Boehm [2] divides the history of Software Engineering into decades. The data for ICSE, however, reveals that the eras in Software Engineering run faster. In a linear and summarized manner, these data brings the history of Software Engineering as follows.

- 1980-1992 —Bug and Reuse Era: as stated by Boehm [2], in the 80's the community was concentrated in productivity and scalability, that is related to reuse. However, the ICSE data show that the community was also interested in studying bugs.

Table 3: The highest occurrences of keywords.

Keyword	#Occurrences
Test	290
Bugs	154
Software Architecture	135
Software Process	116
Empirical Software Engineering	110
Measure/Metric	90
Requirements	86
Education	73
Software Engineering	58
Software Quality	57
Reuse	52

- 1993-1996 —Process Era: the main example of the discussion in course in that time is the proposal of the Capability Maturity Model (CMM 1.1), in 1993.
- 1998-2002 —Holistic View of Software Engineering Era: Processes continued to be investigated, and the main event in this subject was the Agile Manifesto, in 2001. Education, software measurement, empirical studies, and software quality started to be considered in large scale. Reuse returned to be discussed, and the central subjects were test and software architecture.
- 2004-2009 —Non-process Era: research about process and reuse lost the main attention of the software community. A possible reason might be that Agile Methods had a central role in the 2000's.
- since 2010 —Bug and Test Era: research in process and reuse reappeared timidly in 2012, but the central subjects are bug and test. Boehm [2] argue that robust solutions for global collaborative processes are a necessity in the 2010's. This fact might be the reason why the topics related to bug and test have gained increasing attention when compared to other subjects in Software Engineering.

RQ3: What are the contributions produced in Software Engineering?

This research question investigates the kind of results that have been produced by researchers in Software Engineering. As described in Section 2.2.3, we classified the results of the primary studies according to the the categories defined by Mary Shaw [6] to answer this research question.

Figure 2 shows the distribution of the kinds of studies carried out from 1998 to 2018. By far, the most common type of results that ICSE papers have reported are *procedures or techniques*. *Empirical studies* were timidly reported in 1988, but have increasingly gained attention and became the second most common kind of study in Software Engineering. *Tool* is the type of result that has always been considered by the community. This behavior may occur because, in general, tools are developed to automate previously proposed techniques and procedures. Moreover, the existence of a special track in ICSE to publish tools certainly contribute to promote this type of study.

Descriptive or qualitative models are results that have been expressively produced in Software Engineering. In general, the works that provide this type of result have reported novel theory or formalism, and description of new frameworks. In contrast, *empirical and analytic models* are not widespread in Software Engineering according to data of ICSE.

Specific solutions are contributions that have been produced since 1988. By its distribution, it is noticed that the ICSE community has not been very concerned on this type of results since there is a low quantity of studies that have been produced over the years. The year which presented the highest production of this type of result is 2004. Finally, contributions regarding *reports* were widespread until 2008. After 2008, the number of studies producing reports declined considerably compared to previous years.

This analysis reveals that most works in Software Engineering result in new procedures or techniques, carry out empirical study or develop new tools. This behavior is observed along the whole time of ICSE publications.

4 RELATED WORK

Some studies have performed literature reviews to classify the types of papers, works, and subjects that have been used in Software Engineering, as we describe following.

Glass et al. [5] reviewed the Software Engineering literature to identify: the topics researchers address, the approaches and methods they apply, the Software Engineering reference disciplines the works refer to, and the level of analysis performed in the studies. They considered 396 paper published in six journals from 1995 to 1999. The report by Cai and Card [3] identifies the main topics investigated in Software Engineering. They considered data from 691 papers published in seven journals and seven conferences in 2006.

The analysis of Garousi and Fernandes [4] considered the citation count to identify the top-100 most influential papers in the area of Software Engineering. Bertolino et al. [1] carried out a study about the types of papers that have been accepted in the main track of ICSE from 2012 to 2016. They identified the eight topics that have been mostly investigated per year, the types of problems that have been mainly addressed (development methods, analysis method, specific instance, generalization or characterization, and feasibility study or exploration), the main types of contribution (theoretical, technological, empirical, and perspectival), and the main types of validation applied to the studies (analysis, evaluation, example, experience, and no validation).

Our work also is based on literature review. However, it differs from previous work in many aspects. Cai and Card analyzed 691 studies, from 14 distinct sources. Glass et al. considered even fewer papers and is based on data from 1995 to 1999. Garousi and Fernandes have concentrated in the aspect of paper influence, whereas we consider other aspects.

Similarly to our study, Bertolino et al. have focused on ICSE publications. However, the aim and the method of our study are different from theirs. Our purpose is to study the Software Engineering evolution by analyzing data from works published in ICSE, and not to provide an analysis of paper acceptance at ICSE. Besides, we analyzed data from documents published in any track of ICSE,

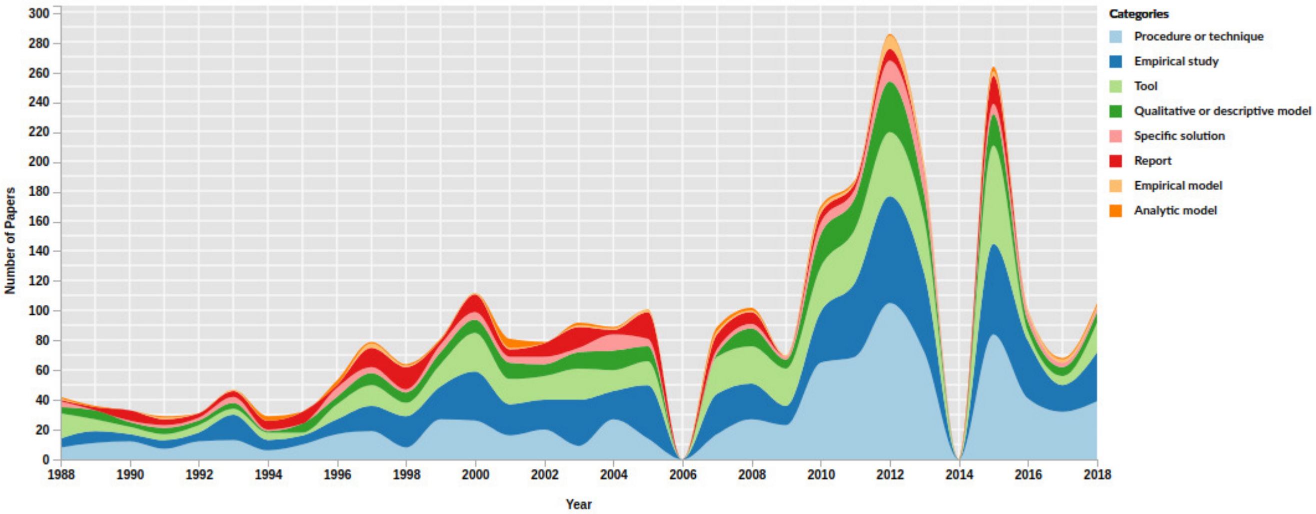

Figure 2: Kinds of contributions in Software Engineering over the time.

considering the editions of the conference from 1988 to 2018, in a total of 3,300 documents.

5 FINAL REMARKS

Software Engineering is a young field of human experience if compared with others. However, the knowledge in this field has evolved at a very high speed, that is a characteristic of Computer Science in general. In this paper, we present an analysis of Software Engineering knowledge evolution by analyzing ICSE publications. We chose to consider ICSE because it is the main and the oldest conference in the area. Moreover, in contrast with other foruns, ICSE covers Software Engineering as a whole.

We analyzed data from works published in ICSE from 1988 to 2018. In the total, data of 3,300 documents and 4,916 authors were analyzed in the present paper. Our study has brought the following conclusions. (1) Since the beginning (1988, in this case), the main type of research carried out by the community is the development of new procedures and techniques, followed by experimental studies. (2) The main topics investigated by researchers are, in this order: tests, bugs, software architecture, measurement, requirement, education, empirical study, software quality, software process, and reuse. (3) The Software Engineering evolution may be divided into eras in which a set of those topic gains more attention by the community. For instance, the apex of studies in software processes was between 1992 and 2002. Nowadays, although most works are focuses in test and bugs, Software Engineering is studied holistically, i.e., the attention of the community is also shared among many subjects, specially software architecture, measurement, requirement, education, empirical study, and software quality.

This study brings a compilation of Software Engineering evolution that may be of value to the software community. The results aid the comprehension of how the body of knowledge in the field has been considered over the time. As ICSE is the main forum of Software Engineering discussion, we believe that our analysis is a reliable portrait of Software Engineering evolution. Nevertheless, it will also be of value to study data before 1988 to comprehend the beginning of the area. It will also be interesting to evaluate how the discussion of the main key topics have evolved, to aid the comprehension of the history of specific subjects. Analyzing the publications of the main contributors in Software Engineering will be of interest of the academic community to identify aspect such as hints to write a good paper and successful research methods in Software Engineering. The content of this paper is part of our research on Software Engineering. We have analyzed the data we collected in order to investigate other questions.

ACKNOWLEDGES

This work was supported by Capes, CNPq and CEFET-MG.

REFERENCES

[1] Antonia Bertolino, Antonello Calabrò, Francesca Lonetti, Eda Marchetti, and Breno Miranda. 2017. What Paper Types Are Accepted at the International Conference on Software Engineering?. In *39th International Conference on Software Engineering (ICSE)*. teste, teste, 399–401.
[2] Barry Boehm. 2006. A View of 20th and 21st Century Software Engineering. In *28th International Conference on Software Engineering (ICSE)*. 12–29.
[3] Kai-Yuan Cai and David Card. 2008. An analysis of research topics in software engineering – 2006. *Journal of Systems and Software* 81, 6 (2008), 1051–1058.
[4] Vahid Garousi and João M. Fernandes. 2016. Highly-cited Papers in Software Engineering. *Inf. Softw. Technol.* 71 (2016), 108–128.
[5] R.L. Glass, I. Vessey, and V. Ramesh. 2002. Research in software engineering: an analysis of the literature. *Information and Software Technology* 44, 8 (2002), 491–506.
[6] Mary Shaw. 2003. Writing Good Software Engineering Research Papers: Minitutorial. In *25th International Conference on Software Engineering (ICSE)*. 726–736.
[7] IEEE Computer Society, Pierre Bourque, and Richard E. Fairley. 2004. *Guide to the Software Engineering Body of Knowledge (SWEBOK)*. IEEE Computer Society Press.
[8] IEEE Computer Society, Pierre Bourque, and Richard E. Fairley. 2014. *Guide to the Software Engineering Body of Knowledge (SWEBOK(R)): Version 3.0*. IEEE Computer Society Press.