Delft University of Technology

Impact of Software Engineering Research in Practice: A Patent and Author Survey Analysis

Kotti, Zoe; Gousios, Georgios; Spinellis, Diomidis

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Impact of Software Engineering Research in Practice: A Patent and Author Survey Analysis

Zoe Kotti [ID], Georgios Gousios [ID], and Diomidis Spinellis [ID], *Senior Member, IEEE*

Abstract—Existing work on the practical impact of software engineering (SE) research examines industrial relevance rather than adoption of study results, hence the question of how results have been practically applied remains open. To answer this and investigate the outcomes of impactful research, we performed a quantitative and qualitative analysis of 4 354 SE patents citing 1 690 SE papers published in four leading SE venues between 1975–2017. Moreover, we conducted a survey on 475 authors of 593 top-cited and awarded publications, achieving 26% response rate. Overall, researchers have equipped practitioners with various tools, processes, and methods, and improved many existing products. SE practice values knowledge-seeking research and is impacted by diverse cross-disciplinary SE areas. Practitioner-oriented publication venues appear more impactful than researcher-oriented ones, while industry-related tracks in conferences could enhance their impact. Some research works did not reach a wide footprint due to limited funding resources or unfavorable cost-benefit trade-off of the proposed solutions. The need for higher SE research funding could be corroborated through a dedicated empirical study. In general, the assessment of impact is subject to its definition. Therefore, academia and industry could jointly agree on a formal description to set a common ground for subsequent research on the topic.

Index Terms—Software engineering, practical impact, empirical study, survey, patent citations

---

1 INTRODUCTION

IN 2018, the field of software engineering (SE) marked the 50th anniversary of its first two-year conference series—the 1968–69 NATO Conferences on Software Engineering [1], [2]. Despite its relatively short period of existence, a lot of research has been performed in SE during these 50 years, composing a large body of information. In the meantime, numerous software and technology-related companies have emerged, partially as a result of hardware advancement and cloud computing [3], forming a multi-trillion dollar industry [4]. This growth both in terms of knowledge and market share raises the question of how these two relate, and to what extent research may have impacted industry. In this context, we define as *impact the direct or indirect incorporation of a software engineering study's output in an industrial setting, for example, in an industrial software development tool, process, marketable product, or service.*

In the scope of this study, we consider *software engineering* the discipline that systematically employs computer science knowledge and principles to develop new methods and tools to improve software development. The discipline's areas include software requirements, design, construction, testing, maintenance, configuration management, quality, SE management, SE models and methods, and SE process [5]. The application process is based on systematic, disciplined, and quantifiable SE approaches, and is influenced by cross-disciplinary areas, namely, mathematics, general management, project management, and systems engineering [5]. Note that our definition distinguishes foundational computer science research (*e.g.*, devising a new static analysis method, a test prioritization algorithm, or a requirements definition language) from that performed in SE contexts. For the described examples to be considered SE research, we expect them to be accompanied with empirical evaluation through, for example, repository mining, a developer survey, or a case study.

Existing work on the practical impact of SE research examines industrial relevance rather than adoption of study results. A variety of interviews and literature reviews have been conducted, mainly in domain-specific contexts such as the ACM SIGSOFT Impact Project [6], to assess the relation of research to industrial needs, highlight gaps between the two, and suggest best practices for collaborative projects. However, the question of how research results have been practically applied remains open.

To tackle this question and investigate the outcomes of impactful SE research, we performed a quantitative and qualitative analysis of SE patents citing SE research from four leading SE venues. Patents are by definition practical applications of technology, and are frequently employed as an estimator of the academic research impact (*e.g.*, in the works by Narin et al. [7], Estublier et al. [8], and the National

Academy of Engineering [9]). Software patents have increased rapidly in number, comprising 15% of all patents [10]. Most of them are acquired by large manufacturing firms from the computers, electronics, and machinery industries [10]. Furthermore, we conducted a survey on authors of highly recognized SE publications to examine impactful types, areas, methods, and outcomes of SE research as well as their footprint on information technology, society, and industry.

Our findings suggest that SE researchers have equipped practitioners with various tools, processes, and methods, and improved many existing products. SE practice seems to value knowledge-seeking research and is impacted by diverse cross-disciplinary SE areas. Practitioner-oriented publication venues appear more impactful than researcher-oriented ones, while industry-related tracks in conferences could enhance their impact. Two main obstacles to research adoption seem to be insufficient funding and the unfavorable cost-benefit trade-off of the produced solutions. The study's contributions are:

- the systematic collection of top-rated SE research,
- the systematic collection of SE patents citing SE research,
- the categorization of research according to its type, methods, SE area, and industrial application domain, and
- the synthesis of the preceding results into a taxonomy of the main practical impacts of SE research.

In the following Section we present an overview of existing work on the practical impact of SE research. We then describe the research questions and study methods in Section 3, and present the research results in Section 4. An extensive discussion of the study findings is included in Section 5. The study is complemented by the associated limitations in Section 6, followed by our conclusions in Section 7. Based on published guidelines [11], the code,[1] survey questionnaire, anonymized responses, and produced data[2] associated with this study are publicly available online, and can be used for replication or further empirical research.

2 RELATED WORK

We analyze and synthesize related work on the practical impact of SE research according to the study objective (*i.e.*, practical impact), and the two employed method axes (*i.e.*, patent and author survey analyses). For this, we employ the classification scheme by Lo et al. [12], and classify related work into three areas: research related to the ACM SIGSOFT Impact Project; literature reviews and surveys in SE evaluating the relationship between academia and industry; and ranking studies assessing the impact of SE researchers, institutions, or publication venues. To cover the patent axis, we extend this classification scheme with studies evaluating the impact of SE research based on patent metadata.

2.1 ACM SIGSOFT Impact Project

In the early 2000s, the Impact Project [6] was established, in an attempt "to help both the research and practitioner community to understand each other better", in order to strengthen their cooperation, and also avail funding agencies to maximize their return on investment in SE research. In general, the project aims to study the impact that SE research has had upon software development practice.

The project uncovers state-of-the-art software technologies in specific areas, and examines their influence by former research work, through literature searches and personal interviews [13], [14]. Specific areas include software configuration management [8], modern programming languages [15], software testing and analysis [16], middleware technology [17], inspections, reviews, and walkthroughs [18], and software resource estimation [19].

According to the area-specific studies, academic research tools and services have been adopted by major industrial projects and have influenced various fields. Although some original ideas require a long time (up to 15–20 years [17]), deep reworking, and re-engineering to apply in industrial practice, the constant flow of researchers between industry and academia can expedite adoption. By employing academic research techniques, laboratories have reported up to 95% increase in defect detection before testing, 50% cost reduction for newly developed source code lines, and up to 50% shortening of delivery times.

Conversely, companies have contributed to academia in estimation and mathematical approaches, advanced project planning, and flexible and realistic models [19]. In estimation and mathematics we distinguish the rise of analogy, expert judgment, hybrid, and Bayesian approximation methods. Motivated by corporate projects, academic ones have started balancing their deadlines and activities better, and prioritizing features according to schedule, cost, and quality requirements. Finally, SE models have become more adaptable to the diverse programming languages, new standards, and techniques such as rapid application development, and more realistic on the basis of ten evaluation criteria: *definition*, *accuracy*, *scope*, *objectivity*, *constructiveness*, *detail*, *stability*, *ease of use*, *prospectiveness*, and *parsimony* [19].

2.2 Literature Reviews and Surveys

In 2013, Misirli et al. [20] conducted in-depth interviews with twelve practitioners who were actively collaborating with them at that time in three industrial software analytics projects. These projects involved defect, effort, and quality prediction. Their aim was to explore practitioners' expectations, and ways to employ software analytics solutions in policy making. Respondents suggested enhancing the examined solutions with defect-severity or defect-type prediction, defect location, and phase– or requirement-level effort estimation. Furthermore, they stressed the need for collecting accurate and complete data through the provided solutions, and integrating these solutions into existing systems (*e.g.*, by combining defect prediction results with test interfaces to determine which interfaces to test first).

Around the same time, Beecham et al. [21] conducted interviews with practitioners from ten companies of various sizes to assess the impact of Global Software Engineering

---

1. https://doi.org/10.5281/zenodo.6780414
2. https://doi.org/10.5281/zenodo.7090818

(GSE) research papers in practice. GSE research is regarded as useful, and participants argue that studying the topic might improve performance. Still, none was found to have actually consulted the GSE literature. Practitioners mostly refer to books, blogs, forums, short reports, and their past experience to resolve problems in GSE. Some respondents associated GSE with general project management. The authors argue that GSE research should be relevant (*i.e.*, reflecting the needs of practice), documented in short, evidence-based, and readable papers with validated findings, disseminated more widely as grey literature, and advertised through social media.

Through another survey on 512 Microsoft practitioners, Lo et al. [12] examined the relevance of SE research to practice. Participants rated the relevance of 571 ICSE, ESEC/FSE, and FSE papers published between 2009–2014: 71% of all ratings were essential or worthwhile, while no correlation was observed between citation counts and relevance scores. Reasons behind research ideas rated as "unwise" include: unneeded tools; non-actionable empirical studies; generalizability issues; cost outweighing benefit; questionable assumptions; disbelief in the proposed solution; better alternatives or more crucial problems to handle; and side effects of the suggested solution. Nonetheless, practitioners seem generally positive to studies performed by the SE research community.

In 2017, Ivanov et al. [22] investigated the gaps between research and practice by surveying software engineers and comparing their answers to research topics covered by recent ICSE and FSE publications. Inconsistencies were detected between practitioners' needs and actions: while development productivity was deemed more important than software quality, the majority of the examined publications involve software verification and validation. In addition, practitioners struggle to find improved effort estimation methods.

To facilitate continuous and collaborative technology transfer, Mikkonen et al. [23] proposed a model for large consortia of companies and research institutes working on a common research topic. The model was applied in two national Finnish software research programs. To evaluate and refine it the authors conducted interviews with four participating companies. According to the findings, companies perform substantial SE research to create new business opportunities, and appear willing to provide data to academia for performing empirical SE research when they identify a company benefit. To improve industry-academia collaboration, technology transfer models should foster co-creation and co-learning rather than a linear one-way product transfer from academia to industry. Research institutes can improve their mindset by grasping long– and short-term company needs as well as emerging SE research trends, solutions, and results.

To address the relatively limited joint industry-academia collaborations in SE, Garousi et al. [24] provided a set of challenges and best practices for planning and conducting collaborative projects. The authors denote as challenges: difficulty in understanding the industry problems; differences in objectives, reward systems, and useful attributes; and difficulty in managing intellectual property rights. Among best practices we discern: organizing regular workshops and

seminars with the industry; assuring continuous learning from industry and academia sides; ensuring management engagement; grounding research on real-world problems; demonstrating explicit benefits to the industry partners; and maintaining agility during collaboration.

In a more recent work, Garousi et al. [25] explore and characterize the state of industry and academia collaborations in SE through an opinion survey among researchers and practitioners. Around a hundred collaborative projects from 21 countries were analyzed, revealing that the most frequent topics are testing, quality, process, and project management. The vast majority of collaborative projects result in more than one publication, while more than half have a positive impact on the industry parties, usually through a new approach, method, technique, or tool. To improve industry-academia collaborations, the authors recommend a set of good practices, including performing pilot tests in laboratory settings before industrial releases, cultivating trustful relationships with practitioners, investing in regular meetings to promote the team spirit, and, again, adopting iterative approaches such as agile methods.

2.3 Ranking Studies

To examine the health of SE conferences, Vasilescu et al. [26] used a metrics suite to measure the stability, openness, representativeness, availability, and scientific prestige of eleven conferences in a ten-year window, between 1993–2004. Although SE conferences are generally healthy and display high author turnover, there are considerable differences between wide– and narrow-scoped conferences with regard to the aforementioned measures. For instance, narrow-scoped SE conferences tend to be more introvert than wide-scoped, while maintaining more representative[3] program committees and lower author turnover.

In a 13-part study series spanning the years 1993–2008, Glass et al. [27], [28] assessed scholars and institutions based on the number of their publications in systems and SE. In general, top-ranked academic institutions outnumber industrial research centers. Although the USA was first in number of top-ranked institutions up to 2002, it has been surpassed by the Asia-Pacific institutions since 2003.

Similarly, Ren and Taylor [29] ranked individuals and organizations according to their publications in the ICSE, FSE, TSE, and TOSEM venues between 2000–2004. The majority of top-ranked scholars and institutions come from the USA, while a significant number originates from Europe. The authors argue that "rankings based on publications can supply useful data in a comprehensive assessment process", but Parnas [30] warns that "measuring productivity by counting the number of published papers slows scientific progress". Some organizations may be more conservative with publications. Consider, for example, Apple's limited publishing activity compared to its innovation [31].

A more recent assessment of top-cited SE researchers was conducted by Petersen and Ali [32] by analyzing a multi-field dataset of author citations provided by Ioannidis et al. [33]. The authors report that 37% of top researchers of

---

3. Representativeness of a program committee is measured through the ratio of its members that have never co-authored a paper in the conference's past editions [26].

the dataset were mistakenly assorted in the SE field, while Barry Boehm is the leading SE author. The majority of top SE researchers come from the USA, Canada, or the UK, and are affiliated with Microsoft. Along with SE, researchers are frequently involved in artificial intelligence, image processing, and human factors.

2.4 Studies on Patent Metadata

Shortly after 2000, Agrawal and Henderson [34] evaluated the contribution of patents to knowledge transfer from universities to the industry by focusing on the MIT Departments of Mechanical Engineering, and Electrical Engineering and Computer Science. Patents account for less than 10% of knowledge transfer from the aforementioned Departments, while the majority of the faculty never patents at all. Although patent volume does not predict publication volume, it is positively correlated with paper citations, offering insight into the impact of university research. In general, patenting is a complementary rather than substitutional activity for fundamental research.

Through a systematic analysis of citation linkages between US patents and research papers, Narin et al. [7] assessed the contribution of public science to industrial technology. To collect all papers cited by patents, the authors employed a similar approach to ours (see Section 3.3) by extracting all non-patent references from around 400 000 US patents. In total, 73% of papers cited by patents originate from academic, governmental, and other public institutions, while only 27% were authored by industrial scientists. Patent-to-science linkage has a strong national component with US patents heavily citing US papers, while the linkage is subject-specific (*e.g.*, chemical patents cite chemistry papers). Furthermore, patent-cited engineering and technology papers are mainly published in electrical engineering journals, and IEEE is the top publisher.

3 METHODS

We framed our investigation on the impact of SE research in practice in terms of the following research questions.

RQ1 What types, areas, and methods of SE research are impactful? To answer this, we first collected a set of SE research papers published in leading venues and complemented them with their assigned topics, academic and SE patent citation counts, and awards. Through a survey on authors of most-cited and awarded publications, we identified impactful SE research types, areas, and methods. The set of impactful areas was enriched by extracting the topics of the papers cited by SE patents.

RQ2 What are the outcomes of impactful SE research? For this we retrieved the most-cited SE research by SE patents, and evaluated the citing patents based on their associated litigation cases and maintenance fee events. Furthermore, we examined the correlation between patent and academic citations, and patent citations and academic awards.

RQ3 What types of SE research outcomes are impactful? Through a survey on authors of top-notch SE publications we identified the practical footprint of SE research and how its results have been exploited by the industry. In addition, we detected potential obstacles in the practical adoption of SE research.

RQ4 What are the main practical impacts of SE research on information technology, society, and industry? Through the aforementioned survey we further investigated how premier SE research changed the state of practice.

RQ5 Which SE venues are more likely to publish papers that impact practice? We assessed the practical impact associated with top SE venues through their patent-based impact factors, which we obtained by dividing their publication counts with their patent citations.

RQ6 Is SE research funding sufficient for obtaining results that are relevant in practice? We approximated SE research expenditures (and thus funding) by examining the total number of existing SE publications and PhD dissertations, and compared them to those of the main engineering branches to evaluate sufficiency.

Methods' Overview An overview of the methods we employed to answer these six research questions is presented through a UML information flow diagram in Fig. 1. Their extended descriptions are introduced in the subsequent Sections. From the collected premier SE research (Section 3.1) we retrieved the academic citations and awards, and SE patent citations (Section 3.3). The authors of top-cited and awarded SE publications formed the sample of our survey (Section 3.6). The set of SE patents citing the premier SE research (Section 3.2) was evaluated in terms of litigation and maintenance fee events (Section 3.4). A correlation analysis was also conducted using the academic citations and awards, and SE patent citations (Section 3.5). From the patent-cited SE papers we extracted their research areas and top-cited papers. Through the correlation analysis, SE patent evaluation, and top-cited SE research by SE patents we answered RQ2.

Furthermore, the research areas of the patent-cited SE papers, and the research areas, methods, types, applied research outcomes, practical impacts, and adoption obstacles of the surveyed SE papers provided us insights for RQ1, RQ3, and RQ4. For RQ5 we identified the top SE venues, retrieved their publication and SE patent citation counts, and divided them to compute the venues' patent-based impact factors (Section 3.7). Finally, for RQ6 we extracted the publication and PhD dissertation counts of SE and the main engineering branches (*i.e.*, chemical, civil, electrical, and mechanical [35]), and compared them to assess the sufficiency of provided SE research funding (Section 3.8).

3.1 Premier SE Research

We created a dataset of research papers published in four top-notch SE venues, namely the *International Conference on Software Engineering (ICSE)*, the *IEEE Transactions on Software Engineering (TSE)*, the *ACM Transactions on Software Engineering and Methodology (TOSEM)*, and the *Empirical Software Engineering (EMSE)* journal. From these we retrieved 11 419 papers by downloading the complete DBLP computer science biography database (version April 4, 2017),[4] and filtering its XML records to retain those whose *inproceedings key* tag contained *conf/icse*, and those whose *article key* tag contained *journals/tse*, *journals/tosem*, or *journals/ese*. We excluded records with undefined *title* or *author*
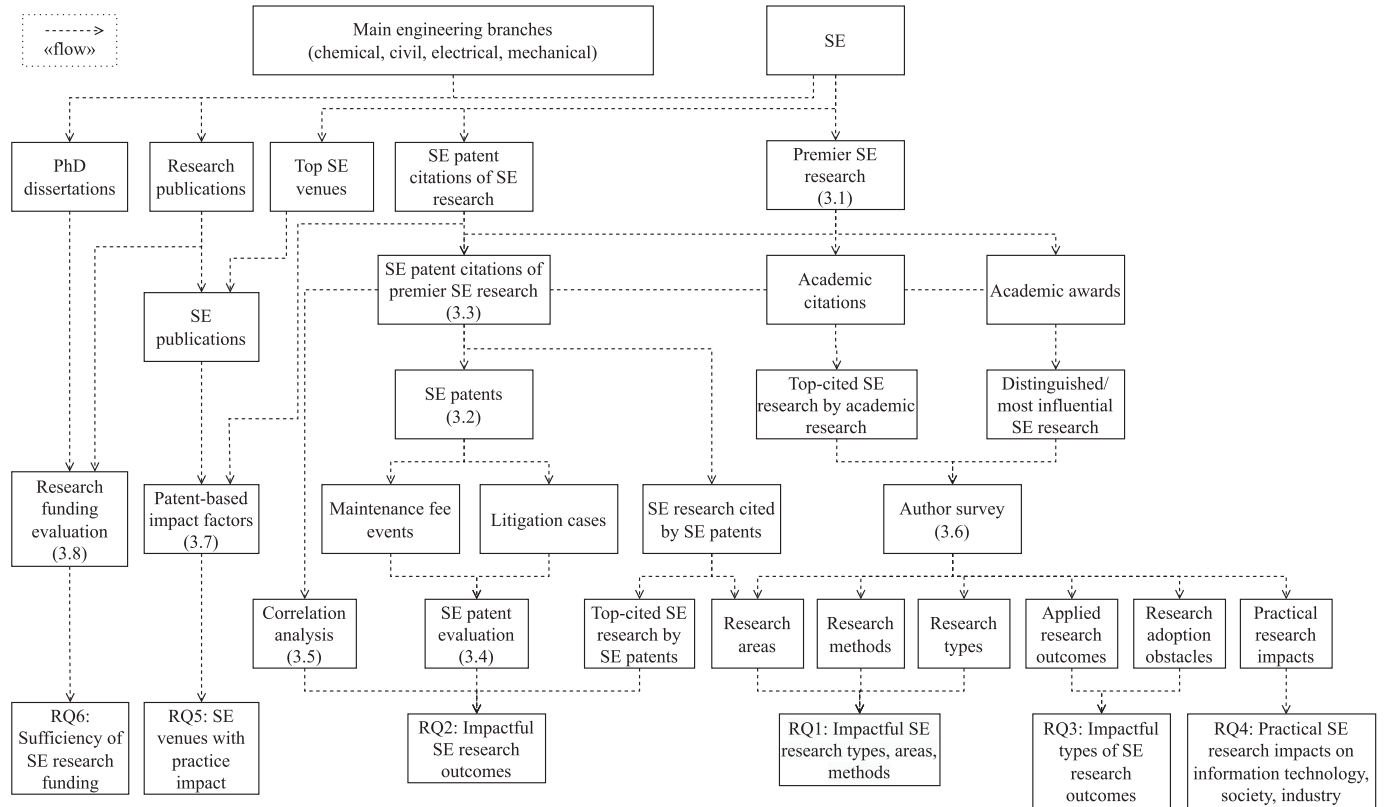
---

4. https://dblp.org/

Fig. 1. Information flow of study methods.

tags, and obtained 6 950 ICSE, 3 417 TSE, 428 TOSEM, and 624 EMSE publications spanning the years 1975–2017. For each selected study, we listed its publication year, first author, title, and digital object identifier (DOI).

We then obtained the topics of the selected studies as follows. For the TSE and the IEEE-published ICSE papers, we used the DBLP-extracted DOIs to download the corresponding IEEE Xplore HTML pages. From these we extracted the INSPEC controlled terms of each paper. In total, we retrieved 536 distinct topics of 1 106 TSE papers, and 547 distinct topics of 1 348 IEEE-published ICSE papers. (The overall distinct topics were 730.)

For the TOSEM and the ACM-published ICSE papers, we obtained the *ACM DL Abstracts and Titles for Research Purposes* database covering the period until 2017 from the ACM Publications Operations Manager (C. Rodkin, personal communication, October 21, 2020). We filtered its XML metadata files to retain those residing in directories starting with *TRANS-TSEM* or *PROC-ICSE*. We normalized the ICSE analysis to include only the regular main track, excluding Companion and Future proceedings as well as those of the following often co-published tracks: New Ideas and Emerging Results; Software Engineering Education and Training; Software Engineering in Practice; and Software Engineering in Society. We removed these tracks to preserve homogeneity of our ICSE dataset, which would otherwise be affected by the tracks' varying start years and substantial missing data in the ACM database. For each paper, we extracted its assigned 2012 ACM Computing Classification System (CCS) [36] concepts from the *concept_id* and *concept_desc* tags. Overall, we obtained 287 distinct topics of 429 TOSEM papers, and 439 distinct topics of 1 648 ACM-published ICSE papers.

For the EMSE papers, we did not use Springer's keywords because they do not have a defined structure. Instead, we retrieved their assigned 2012 ACM CCS concepts. EMSE is not included in the aforementioned ACM database, thus we web-scrapped the CCS concepts of the papers from their associated ACM Digital Library web pages by developing a Python script using the Beautiful Soup package [37]. In this way, we collected 216 distinct topics of 239 EMSE papers. (The overall distinct topics for the TOSEM, ACM-published ICSE, and EMSE papers were 545.)

We also retrieved the academic citations (*i.e.*, references by other research papers) and conference awards of the SE research papers. Since the DBLP data did not contain citation counts, we obtained them from Elsevier's Scopus database[5] by querying the field *Source title* using as input each of the four venue names combined with the publisher (*e.g.*, *ACM Transactions on Software Engineering and Methodology*). In this way, we retrieved 13 862 ICSE, 421 TOSEM, 3 464 TSE, and 694 EMSE records, again spanning the years 1975–2017. For the awards, we manually searched in the ICSE proceedings for all *distinguished* and *most influential* papers up to 2016, identifying a set of 74 and 29 papers, correspondingly.

3.2 SE Patents

To identify SE-related patents, we adopted one of the two approaches recommended by Griliches [38]: employing a patent classification system developed by a patent office. The alternative involves reading and manually classifying individual patents (*e.g.*, the work by Bessen and Hunt [10])—

5. https://scopus.com/

this would restrict the research scope due to our small number of human raters. Instead, we used the Cooperative Patent Classification (CPC) system (version 2020.08),[6] which has been jointly developed by the European Patent Office (EPO) and the US Patent and Trademark Office (USPTO) [39], and "is a further step towards a more general harmonization of the world's patent classification systems" [40].

The CPC system is divided into nine sections, which in turn are subdivided into classes, subclasses, groups, and subgroups—we manually looked for SE categories in all levels of the hierarchy. To ensure consistency of this manual process, guidelines recommended in the work by Brereton et al. [41] were followed: the first author of this paper performed the lookup, and the last author validated the identified SE categories. Specifically, the first author identified as relevant all subgroups under the group G06F8/00 *Arrangements for software engineering*, along with any subgroups mentioned in them that belonged to other groups. For instance, G06F8/451 *Code distribution* references G06F9/5083 *load rebalancing*, and G06F9/5083 contains G06F9/5088 *involving task migration*— the latter was also included in this case. In the end, 173 SE-related categories were identified by the first author.

The last author verified these categories taking into account the lower-level contents of the ACM CCS *Software and its engineering* concept. Categories associated with the following irrelevant CCS concepts were removed: *Hardware*; *Distributed computing methodologies*; *Concurrent computing methodologies*; *Security and privacy*; *Operations research*. Furthermore, the entire CPC subgroup G06Q10/06 *administrative, planning or organization aspects of software project management* mentioned by G06F8/00 was excluded, because although it seems relevant to SE, it effectively applies to any management context. The resulting set of SE-related CPC categories has 117 members.

To retrieve patents belonging to the 117 CPC categories, we queried the Google Patents Public Data (GPPD) dataset [42] on BigQuery. GPPD is a worldwide bibliographic and US full-text dataset of patent publications provided by IFI CLAIMS Patent Services and Google, and updated on a quarterly basis—for this study, the April 2020 version was used. From Table *patents.publications_202004* we extracted 304 368 distinct patents associated with at least one of the aforementioned CPC categories.

3.3 SE Research Cited by SE Patents

Scientific literature cited by patents (*i.e.*, science linkages) can provide insights into the impact of science on industry [43]. Science linkages are usually considered the state of the art and help in evaluating an invention's novelty and patentability. Companies whose patents contain many science linkages are regarded closer to science, basing their technology on scientific progress [43]. Moreover, science linkages may be used as predictors of a company's financial performance: high-tech companies typically surpass their competitors in science linkages [44].

Motivated by this, we assessed the practical impact of the SE papers in terms of their citations by SE patents. First, we collected all non-patent literature cited by the patents, which

is stored in plain text format in Table *patents.publications_202004* (field *citation.npl_text*). Querying this Table we extracted 830 379 text references associated with 92 772 distinct patents. To identify any SE papers in the references, we followed two approaches: DOI crosschecking, and title and author mapping. For the first, we looked up the available 6 017 ICSE, 3 405 TSE, 426 TOSEM, and 509 EMSE DOIs in the references, and found 43 ICSE, 52 TSE, 12 TOSEM, and 3 EMSE papers cited by 121 patents. To include in the process papers and references without available DOIs, we also searched in the references all SE titles, combined with the last names of the first authors. In case both a title and a name were found in a reference, we considered this a match. From this process, 895 ICSE, 630 TSE, 104 TOSEM, and 20 EMSE papers (that were not identified through DOI crosschecking) were found cited by 4 248 patents. In total, 1 690 (912 ICSE, 649 TSE, 107 TOSEM, 22 EMSE) distinct papers are referenced in 4 354 distinct patents.

To evaluate our method, we estimated the accuracy using a random sample of the collected references. The sample size of a total of 6 469 references was calculated at around 363 using Cochran's sample size and correction formula for the proportion [45] (95% confidence, 5% precision). We manually verified the sample references and marked 19 (5%) of them as false positives. These fall into three categories: different paper version (mostly earlier)[7] cited by patent (63%); inherent dataset issue (*i.e.*, wrong paper title or DOI documented in dataset—32%); method insufficiency (*i.e.*, the searched DOI is a subset of the one referenced in patent—5%).

3.4 Evaluation of Citing SE Patents

As explained in Section 3, we estimated the value of the SE patents that cite SE research based on two indicators: patent maintenance fee events and litigation cases. These are recommended measures of patent quality due to the associated substantial monetary expenses [46].

We extracted the litigation cases associated with the citing patents from the USPTO Patent Litigation Docket Reports Data, which contain detailed patent litigation information on 81 350 unique district court cases filed during the period 1963–2016 [47], [48]. Specifically, we joined the files *cases* and *patents* (version 2016),[8] modifying patent publication numbers to follow the convention used in the GPPD dataset, and retrieved the number of cases as well as the corresponding aggregated demanded monetary damages per patent. For each SE paper cited by patents (Section 3.3), we listed the total number and damages of the corresponding litigation cases.

Maintenance fee events of patents granted since 1981 are provided on a weekly basis by USPTO;[9] we used the February 8, 2021 release which comprises 18 523 706 unique events, again adapting patent publication numbers to the GPPD standard. For each event, a fee code is listed, but not the actual monetary value; we manually extracted the

---

7. In patents, the established practice is to cite the earliest version of equally important documents [43].
8. https://bulkdata.uspto.gov/data/patent/litigation/2016/
9. https://bulkdata.uspto.gov/data/patent/maintenancefee/

6. https://www.cooperativepatentclassification.org/

related fee values from the USPTO Fee Schedule[10] (effective since January 2, 2021) as follows. A set of 157 codes are reported in the documentation file of the dataset. From these we excluded 37 codes that are irrelevant to payments or refunds, and four codes subject to the 37 Code of Federal Regulations, Paragraph 1.28, concerning debts occurring from errors in the small entity status—these are not fixed values. The remaining codes were mapped to their values based on the USPTO Fee Schedule, while deprecated codes were first associated with the replacing ones through the Fee Schedule Crosswalk, FY2002–2003.[11] Finally, a total of 112 codes regarding payments of maintenance fees, surcharges, and refunds were mapped to the current fee rates.

Similar to litigations, for each SE paper cited by patents, we extracted the number and aggregated monetary value of the corresponding patent maintenance fee events. To do this, we computed the monetary value of each distinct patent by summing its fee payments and surcharges, and subtracting any refunds. Refund cases were also excluded from the total number of maintenance fee events of each patent.

3.5 Correlation Analysis

We investigated whether patent citations are correlated with academic citations and academic awards. For the 1 690 papers cited by patents we joined their patent citations with the academic ones, and also marked the awarded papers. To select the appropriate correlation coefficient, we tested the two citation distributions for normality with D'Agostino and Pearson's omnibus test of normality [49], and found that they do not follow a normal distribution. Therefore, we used Spearman's rank correlation coefficient (ρ) [50], which summarizes the monotonic relationship between two variables that do not follow a normal distribution. Patent citations constituted the dependent variable.

3.6 Survey on SE Research Authors

We conducted a survey on authors of exceptional SE publications following the set of ten activities recommended in Kitchenham and Pfleeger's six-part series of survey research principles [51], [52].

Survey Design We adopted a cross-sectional, case-control, observational study design, which means that candidates were surveyed about their past experiences at a fixed point in time [53]. The goal of the survey was *to examine how landmark research has affected SE practice*, therefore we framed the objectives of the survey in terms of RQ1, RQ3, and RQ4 introduced in Section 3.

Survey Sample The sample was composed of the first authors of the most-cited studies published in ICSE, TSE, TOSEM, and EMSE as well as studies that received distinguished and most influential paper awards. We consider that first authors are the ones who have contributed the most to the associated research [32], and are therefore more familiar and knowledgeable about the investigated topics, and also the most appropriate to provide the required feedback. For each venue and year up to 2016, we selected the five

most-cited publications (Section 3.1), leading to a set of 613 studies. We complemented this with the 103 awarded distinguished and most influential ICSE papers. After removing duplicate studies (*i.e.*, ICSE publications that were subsequently extended in TSE, TOSEM, or EMSE) keeping the latest occurrences, our final set included 677 distinct papers, which were associated with 566 distinct first authors. From these, we managed to contact via e-mail 475 first authors of 593 papers (204 ICSE, 176 TSE, 121 TOSEM, 92 EMSE). These constituted the survey sample. Contact failures involved missing or defunct e-mail addresses, deceased or unavailable authors, and rejected e-mail deliveries.

A total of 50 (out of 475—10%) surveyed authors associated with 58 (out of 593—10%) papers were practitioners when their work was published. To compute this we retrieved the first author affiliations from Elsevier Scopus's API through the *pybliometrics* Python interface [54] using the publication DOIs as input. We excluded affiliations containing the (case-insensitive) string "univ", which would most likely involve universities, and manually looked in the remaining records for companies. In this way we identified 50 authors affiliated with 36 companies, including IBM, Nokia, Microsoft, Sun Microsystems, AT&T, General Electric, Intel, and Robert Bosch.

Survey Instrument Participants were provided with personalized questionnaires that mentioned at the beginning the author's name, the examined paper, the venue and year of publication, and the reason it was selected (*i.e.*, top-cited, distinguished, or most influential). (First authors of multiple papers were provided with multiple such questionnaires.) They could also include and review additional publications of theirs not included in the list, which they considered to have made significant impact on SE practice. A total of five responses were collected from this option from four distinct participants.

The questionnaire was composed of mandatory and optional open-ended, multiple choice, and Likert scale questions, accompanied by neutral and free-text options. Participants were initially requested to rate on a three-level Likert scale the extent of practical impact of their publication, and specify through a multiple choice question in what products or processes their work has been incorporated. In an open-ended question, they were subsequently invited to expand on the practical impact, along with ways in which their work changed the state of practice. In case of absence of practical impact, they were asked to comment on the reasons behind this. Furthermore, respondents were asked to select the research areas of their work from a list with the first– and second-level entries to the *Software and its engineering* concept of the ACM CCS [36]. Another question involved specifying the research types and methods employed in the publication from a list adapted from the work by Easterbrook et al. [55]. Finally, participants could list other impactful papers (of which they were not first authors), leave their e-mail address to receive a report with the survey results, and comment on the survey and its topic.

Survey Evaluation Two pilot studies were conducted on candidates of the survey sample to evaluate and refine the questionnaire. The first pilot was internal and was completed by three members of the laboratory associated with four publications. The second pilot was external and was

---

10. https://www.uspto.gov/learning-and-resources/fees-and-payment/uspto-fee-schedule
11. https://www.uspto.gov/learning-and-resources/fees-and-payment/fee-schedule/fee-schedule-crosswalk-fy2002-2003

distributed to a random subset of eleven candidates linked to 16 publications. This trial was held from September 5th to 30th, 2017, and we received six responses from six distinct participants (55% response rate in terms of authors, 37% in terms of papers).

Survey Operation The final survey ran from October 1st to November 5th, 2017, and from May 18th to 31st, 2022. The pilot and the first run of the final survey were hosted on the *SurveyGizmo* online survey platform,[12] while the second run was provided as a *Google form*. Both runs were distributed to the candidate participants through an invitational mail. E-mail addresses were manually fetched from the candidates' personal websites. The mailing process was automated but retained personalization, as explained before. Candidates were informed about the average time required for the questionnaire completion—around three minutes, and the survey objective. The final survey received in total 165 responses from 125 distinct authors (26% response rate in terms of authors, 28% in terms of papers). The anonymized responses are included in the provided dataset. Some answers were redacted upon respondents' request.

Survey Analysis We applied manual coding [56] to summarize the answers to the four open-ended questions and the free-text option of a multiple choice question. For each question, the second and third authors of this paper split the answers in two sets, and each individually applied codes to a half (in a shared online spreadsheet). At least one and up to six codes were applied to each answer. Next, the first author grouped together conceptually-related codes by generalizing or specializing them, following the Qualitative Content Analysis approach [57].

3.7 SE Venues

We assessed the practical impact of top SE venues by computing their patent-based impact factors as follows. We retrieved the Google Scholar Metrics list of top publications under *Categories > Engineering & Computer Science > Software Systems* (July 2021 index),[13] excluding the following entries that center on programming languages and algorithms: PLDI, POPL, Proceedings of the ACM on Programming Languages, TACAS, PPOPP. For the remaining venues we extracted their publication counts from Elsevier Scopus for a ten-year window between 2009–2019. This window was selected to approximate publications of all years, avoiding bias due to different venue start dates, while the particular range was chosen to align with the GPPD version. Therefore, we queried all venue names combined with the year range, restricting document types to reviews and articles for journals, and papers for conferences, similar to Clarivate's impact factor calculation [58]. To retrieve the patent citation counts of the SE venues, we searched their full names, abbreviations, and acronyms in the 830 379 non-patent references of patents (Section 3.3). Finally, citation counts were divided with publication counts to calculate the patent-based impact factors.

3.8 SE Research Funding

We also approximated the sufficiency of provided SE research funding by comparing it to that of the main engineering branches. We initially searched for existing empirical analyses in the literature, but did not obtain any fruitful results. As a workaround, we approximated funding based on the number of existing publications and PhD dissertations, given that both activities are usually grant-aided. We compared SE results to those of the four main engineering branches: civil, mechanical, electrical, and chemical [35]. For publications, we extracted from Scopus all English papers between 2010–2020 that belong to the subject areas *Engineering*, *Computer Science*, or *Chemical Engineering*, and contain the above engineering fields in the keyword list. For dissertations, we queried the Open Access Theses and Dissertations (OATD) database,[14] and retrieved all English PhD dissertations from the same period, whose subject and discipline are the corresponding engineering fields (except for SE, where we specified computer science as discipline).

We used OATD for the following reasons. The database contains more than six million electronic theses and dissertations (ETDs) published between 1973–2022 (retrieved April 20, 2022)[15] from about 1 100 universities, colleges, and research institutes [59]. OATD is considered one "of the finest resources to access ETDs worldwide" [59], and is a recommended international thesis resource for researchers by the Networked Digital Library of Theses and Dissertations (NDLTD) [60]. In a recent evaluation of four prominent search engines on retrieving ETDs through various search techniques including title, keyword, and author search, OATD was ranked second in overall performance (88.5%), closely following Google (89%), and surpassing Yahoo (78%), and Google Scholar (76%) [61]. Furthermore, OATD has been used in various research studies. These mainly evaluate the quality of the database and its repositories [59], [61], [62], investigate user behavior [63], assess and contrast cultural heritage [64], [65], and identify the benefits and impact of open access PhD e-theses [66].

4 RESULTS

In this section we present the study findings from the patent and survey analysis, in respect to the research questions described in Section 3. Survey percentages are calculated on the basis of responses (165) rather than distinct authors (125). Codes derived from the manual coding process of the open-ended answers are set in **bold**.

4.1 RQ1: Impactful Research Types, Methods, Areas

Among survey participants, 40% (66 participants) consider that the work described in their paper resulted in *some* practical impact and 29% (47) in *wide* practical impact, as opposed to 21% (35) who do not believe that their work had any practical impact, followed by 10% (17) who are unaware of any footprint. The variety in the responses could be explained by the entailed subjectivity, the fact that awards and citations are, by definition, not exclusively tied to

---

12. https://www.surveygizmo.com/
13. https://web.archive.org/web/20211118180610/https://scholar.google.gr/citations?
view_op=top_venues&hl=en&vq=eng_softwaresystems

14. https://oatd.org/
15. https://web.archive.org/web/20220420032348/https://oatd.org/oatd/search?q=*%3A*&sort=date

TABLE 1
Practical Impact of SE Research Types (According to Survey Respondents' Self-Evaluation)

Type	Papers	Impactful	%
Empirical	119	85	71
Design	72	53	74
Theoretical	34	26	76

TABLE 3
Practical Impact of SE Research Areas

ACM CCS SE Area [36]	Papers	Impactful	%
Software creation and management	142	104	73
Software organization and properties	106	77	73
Software notations and tools	93	71	76

practical impact, or the authors' different impact expectations. For instance, authors may be more ambitious than award-givers and researchers citing their work.

Table 1 displays the practical impact of SE research types declared in the collected responses. Types are sorted in descending order of appearance frequency. Empirical research (*e.g.*, investigating the adoption of engineering methods, developing new tools) appears the most common type among the examined publications, followed by design research (*e.g.*, developing new methods) and theoretical research (*e.g.*, proving properties of systems axiomatically). Papers that were characterized by some or wide impact were grouped together as *impactful*. All types proved considerably impactful.

Table 2 presents the practical impact of the sample's employed research methods in descending frequency order. Action research (*e.g.*, being embedded in the development team) and exploratory research (*e.g.*, describing characteristics of a population or phenomenon under investigation) are the most impactful ones. Case study (*e.g.*, applying a new technique on existing systems) is the most frequent method but less impactful. In addition to our survey's predefined set, the following methods were also reported for the examined publications: **survey**, **secondary research** (*e.g.*, systematic review, literature review, meta-analysis), **replication study**, **content analysis**, **econometric analysis**, **data collection and analysis**, **formal theory**, and **design and evaluation**. Both research types and methods can overlap, because papers may employ many of them.

The impact of SE research areas and subareas based on the survey findings can be deduced from Tables 3 and 4, correspondingly. Areas and subareas correspond to the first– and second-level entries to the *Software and its engineering* concept of the ACM CCS, respectively. From Table 3, the overall most impactful area is *Software notations and tools*, but the most frequent one is *Software creation and management*. The most impactful subarea (Table 4) is *Development frameworks and environments*, followed by *Contextual software domains* (*e.g.*,

operating systems) and *Compilers* with the fewest publications. The most frequent subareas are *Software verification and validation* and *Extra-functional properties* with moderate impact. Again, both areas and subareas can overlap.

We further retrieved the most frequent IEEE INSPEC controlled terms and ACM CCS concepts of the SE papers that are cited by SE patents (Section 3.1). We retrieved the topics of 277 ACM– and 292 IEEE-published ICSE papers, 107 TOSEM, 227 TSE, and 14 EMSE papers. The ten most common INSPEC terms are *program testing* (24%—124), *software maintenance* (16%—81), *object-oriented programming* (13%—69), *program debugging* (13%—68), *program diagnostics* (12%—65), *software engineering* (11%—57), *Java* (11%—56), *formal specification* (10%—54), *software tools* (9%—47), and *software quality* (8%—42). Similarly, the ten most frequent CCS concepts are *Software testing and debugging* (29%—116), *Software development process management* (19%—77), *Software management* (14%—55), *Designing software* (12%—49), *Software maintenance* (12%—49), *Formal software verification* (11%—43), *Program verification* (11%—43), *Software design techniques* (%—42), *Development frameworks and environments* (10%—40), and *Software creation and management* (10%—39).

4.2 RQ2: Outcomes of Impactful SE Research

A total of 1 690 SE papers have been cited by 4 354 SE patents; the 20 most-cited ones are summarized in Table 5. The majority were published in ICSE between 1984–2013. Through a correlation analysis between patent and academic citations, and patent citations and academic awards (Section 3.5), Spearman's ρ was calculated at 0.25 and 0.07, respectively, suggesting a weak positive correlation in both cases.

From the patents' evaluation in terms of litigations (Section 3.4), twelve patents citing 15 papers are associated with 20 litigation cases concerning patent infringements, and five of these papers are included in the 20 most-cited ones (Table 5). No damages are documented for the 20 cases in the USPTO Patent Litigation Docket Reports Data, but we manually looked them up online and retrieved results for 14 of them: six cases' damages are undisclosed, while the remaining eight cases' range from six million to eight billion dollars. Looking at the involved parties, the majority are/were big corporations, including Apple, Kodak, Ericsson, Facebook, Google, Lenovo, Microsoft, Oracle, Yahoo!, Sun Microsystems, and Radware.

Regarding renewals, 1 953 (45%) patents citing 1 159 (69%) papers are linked with maintenance fee events. Eight papers of the top ten in fee expenses are also among the 20 most-cited. The number of maintenance fee events of patents citing papers with an equivalent number of patent citations is higher for older ones, which is reasonable as the related patents may have been renewed more times. Although, in general, the number of maintenance fee events seems to be associated with the

TABLE 2
Practical Impact of SE Research Methods

Method	Papers	Impactful	%
Case study	85	63	74
Controlled/Natural experiment	48	32	67
Exploratory research	31	25	81
Action research	18	16	89
Ethnography	15	10	67
Simulation	10	8	80
Other	39	21	54

TABLE 4
Practical Impact of SE Research Subareas

ACM CCS SE Area [36]	Subarea	Papers	Impactful	%
Software creation and management	Software verification and validation	62	47	76
	Software development techniques	58	44	76
	Designing software	48	37	77
	Software development process management	38	29	76
	Software post-development issues	32	26	81
	Collaboration in software development	20	14	70
	Search-based software engineering	11	9	82
Software organization and properties	Extra-functional properties	61	45	74
	Software functional properties	49	38	78
	Software system structures	38	26	68
	Contextual software domains	6	5	83
Software notations and tools	Software maintenance tools	44	36	82
	Development frameworks and environments	38	36	95
	System description languages	29	23	79
	Software configuration management and version control systems	20	15	75
	General programming languages	19	12	63
	Formal language definitions	14	9	64
	Software libraries and repositories	14	10	71
	Context specific languages	11	9	82
	Compilers	6	5	83

aggregated fee value, there are substantial differences in some cases. For example, although the number of maintenance fee events is equivalent for the works by Zhang and Cheng [69], and Moher [82], the aggregated fees of the latter are almost double that of the former. Such discrepancies could be explained by the different entity status of the citing patents: fees for large-entity patents are twice that of small-entity, and quadruple that of micro-entity.[16]

4.3 RQ3: Impactful Types of SE Research Outcomes

The results of 44% (72) of the examined publications of the survey were incorporated in software industry processes, practices, or methods, 41% (68) in software development tools, 21% (35) in marketable products, 13% (22) in marketable services, and 19% (31) in other areas. In Fig. 2 we present an overview of the practical impact of SE research grouped by type of outcome, as expressed in the responses to the complementary open-ended question. For each category we show the number of involved responses as weights. An extensive review is included below, enhanced with example quotes to support our interpretation of the responses [87]. Example quotes are marked with an [RX] notation, where *X* refers to the respondent's identification number.

From the open-ended responses we discern the improvement of software product attributes, particularly **design**, **testing**, and **quality**. In their studies, researchers introduced methods and tools for resolving conflicting requirements [R190], and documenting and understanding a system's design evolution through flexible solutions (*e.g.*, [R48,R195, R400]), supported assertion checking [R178], test approach and testing technique selection [R244,R246,R320], model-

based testing and prioritization [R214], and applied techniques such as combinatorial testing [R157] and symbolic execution [R154]. Quality was enhanced through fault injection and clone detection tools (*e.g.*, [R164,R193]), and by analyzing software **faults and failures**. In this way, researchers proposed refined experiment and testing techniques (*e.g.*, [R179,R199]), proved that combinatorial testing "*can provide assurance effectively equivalent*" to exhaustive testing [R157], and also accelerated fault localization [R100].

Along with product attributes, we also distinguish the advancement of software processes, which provide support throughout a software product life cycle [5]. **Maintenance** was improved by "*increasing the visibility of software refactoring research*" [R52], "*providing candidate patches to defects*" [R153], and by warning developers about test smell issues [R317]. **Development collaboration** was enhanced through the advancement of code review tools [R222] and the assessment of global software team configurations [R76]. **Analysis**-wise, a study's empirical results "*motivated many teams to adopt static analysis*" [R170], while other efforts found application in meta-programming, process and software analysis (*e.g.*, [R1,R49, R218,R223,R251]). To estimate **cost, risk, and effort**, researchers published effective models and metrics [R133,R211], and enhanced analogy-based reasoning tools [R173]. Meanwhile, the objective of some studies was to **demonstrate value or raise awareness** about a topic or existing work. For instance, [R112] aimed "*to alert practitioners for the need for assessing complexity*", [R115] performed an empirical validation on quality measures, which "*are now incorporated into static analysis tools, quality assurance practices, and as quality level agreements in software contracts*", and [R136] "*raised awareness of software development and evolution as an economic activity*".

Significant impact was made on existing or new **tools**. The most notable contributions concern tools for static program analysis [R49], logic model checking [R90], software

16. https://www.uspto.gov/learning-and-resources/fees-and-payment/uspto-fee-schedule

TABLE 5
Most-Cited SE Papers by SE Patents

#	Title	Authors	Venue	Year	SE Patent Citations	Patent Lit. Cases	Patent Maint. Fee Events	Patent Maint. Fees ($)
1	Tracking down Software Bugs Using Automatic Anomaly Detection	Hangal and Lam [67]	ICSE	2002	45	-	34	64 830
2	EDMAS: A Locally Distributed Mail System	Almes et al. [68]	ICSE	1984	43	3	67	**260 360**
3	Model-based Development of Dynamically Adaptive Software	Zhang and Cheng [69]	ICSE	2006	43	-	37	75 760
4	Software Deployment, Past, Present and Future	Dearle [70]	ICSE	2007	40	-	21	43 760
5	Aspect-oriented Programming	Kiczales [71]	ICSE	2005	39	-	42	114 290
6	Program Slicing	Weiser [72]	TSE	1984	35	-	30	62 680
7	The Eden System: A Technical Review	Almes et al. [73]	TSE	1985	34	2	79	**312 580**
8	Software Engineering Economics	Boehm [74]	TSE	1984	33	-	25	70 340
9	Distribution and Abstract Types in Emerald	Black et al. [75]	TSE	1987	33	2	77	**306 820**
10	A Cooperative Approach to Support Software Deployment Using the Software Dock	Hall et al. [76]	ICSE	1999	32	1	44	**150 360**
11	Automated Software Test Data Generation	Korel [77]	TSE	1990	32	-	35	**130 470**
12	Safe Software Updates via Multi-version Execution	Hosek and Cadar [78]	ICSE	2013	31	-	14	26 500
13	Predicting Source Code Changes by Mining Change History	Ying et al. [79]	TSE	2003	29	-	14	31 780
14	Call Path Profiling	Hall [80]	ICSE	1992	27	-	54	**154 180**
15	Hipikat: Recommending Pertinent Software Development Artifacts	Čubranić and Murphy [81]	ICSE	2003	27	-	22	46 020
16	PROVIDE: A Process Visualization and Debugging Environment	Moher [82]	TSE	1988	27	-	38	**134 230**
17	Refactoring	Fowler [83]	ICSE	2002	27	-	18	50 500
18	The *Pan* Language-Based Editing System	Ballance et al. [84]	TOSEM	1992	27	-	46	**164 000**
19	Recovering Traceability Links in Software Artifact Management Systems Using Information Retrieval Methods	Lucia et al. [85]	TOSEM	2007	26	-	8	16 000
20	An Intrusion-Detection Model	Denning [86]	TSE	1987	25	1	29	70 140

Papers with **bold** fees are among the top ten with the highest fees.

fault and defect prediction (*e.g.*, [R111,R148,R301,R302, R306]), symbolic simulation [R154], and clone detection [R164]. Some of these tools underwent **industrial and commercial adoption** from startups [R51,R106,R156,R207] to established corporations, such as Microsoft [R222], ABB Corporation Research [R196], BlackBerry [R211], Huawei [R188], Rational Software [R224], and IBM [R251]. **Acquisition and licensing** of tools, techniques, and even startups,

Fig. 2. Practical impact of SE research.

decreased the acquirers' software development costs, and increased substantially the inventors' revenue. As [R224] declares, they "*eventually had almost one billion in annual revenue*". Moreover, the acquired work is now used globally by numerous people (*e.g.*, [R106]'s startup was acquired by Facebook, leading to an approximate monthly impact on "*more than two billion people worldwide*").

Some efforts directly impacted a particular domain. To assist domain-specific language (**DSL**) **and IDE development**, researchers produced a visual language for software modeling and specification [R7], "*encouraged more research on preprocessors*", and experienced adoption of their work by open source frameworks for feature-oriented software development [R215]. Exploiting **formal methods**, ad hoc pragmatic reuse tasks were simplified [R228], and bounded model checking of multi-threaded software was improved, amplifying the analysis of larger problems and "*reducing the verification time over state-of-the-art techniques*" [R97].

In the field of **systems development**, we distinguish contributions to version control [R57], concurrent [R90], hybrid [R154], service-oriented [R250], and embedded software systems [R206]. A frequently occurring method is model-based systems engineering (**MBSE**). Contributions to this involve component models [R68,R242], consolidated process models "*offering the ability to streamline process analysis and redesign work*" [R223], and modeling language semantics of hybrid systems [R154].

Various studies affected subsequent research. Some formed the **groundwork for further research**, obtaining
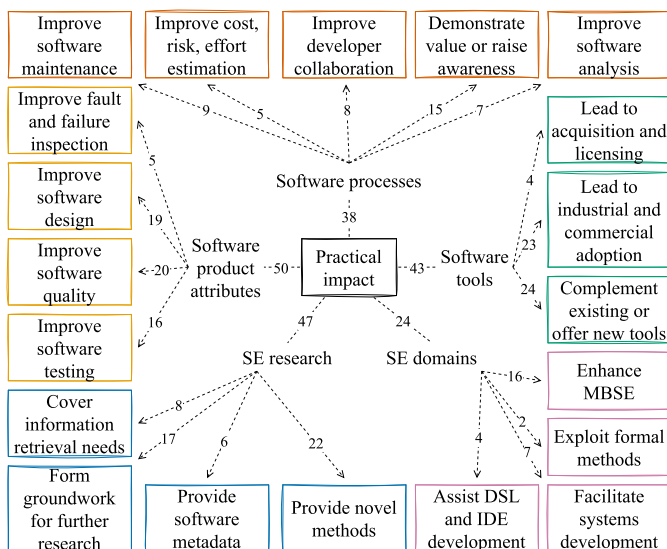
multiple academic references [R5,R6,R7,R197]. As [R116] states, their work on software process models *"blew up the foundation of clean top-down models and forced consideration of real world issues"*. Some **novel methods** concern qualitative research methods aiming to *"help researchers immerse themselves more fully in practice"* [R163], goal orientation in requirements engineering [R189], experience base automation [R243], and standardized methods (*e.g.*, drawing binary trees [R70], collecting data [R234], performing manual inspections [R316]). By providing **software metadata**, researchers managed to *"reduce the cost of widely used software practices"* [R54] and to increase analysts' efficiency [R1,R227, R309]. To cover **information retrieval needs**, researchers proposed new research agendas [R166], influenced source code search solutions [R196], and identified substantial mismatches between IDE designs and information seeking [R102,R103].

Lack of practical impact: Respondents who reported no practical impact of their work were subsequently encouraged to determine the reasons for this inefficiency. Some efforts that could have been impactful did not succeed due to an **immature phase**, **inefficiencies**, or additional required **effort and resources**. In some cases, a longer time horizon was needed to cause an impact [R213,R216]. As [R213] explains, their paper *"is at the intersection of programming language (PL) and SE research, and PL research has a longer time horizon than typical SE research"*. Concerning inefficiencies, some papers contained wrong assumptions [R47], tooling issues [R64], undocumented practical uses [R149], incomplete resolution of the addressed problem [R226], and rarely employed research methods [R118]. Furthermore, some research outcomes have not been used by practice due to high implementation and maintenance costs [R226], risk aversion for technology commercialization [R81], and paucity of maintenance by their creators [R64].

Additional causes involve **undetected needs** and lack of **support by SE practice**. In [R225]'s words, *"most projects feel they are doing 'good enough' with their existing processes"*, and *"a project will not invest the time and effort [into integrating a process such as triage recommendation] unless there is a perceived significant benefit"*. Regarding support, some participants expressed their concerns about SE practice neglecting research, for example, in requirements engineering [R141] and defect density estimation [R321], while others recognized high barriers of work adoption in programming languages [R213], and tool integration challenges in the current SE community [R144]. It comes as no surprise, therefore, that trending software development tools in worldwide practice have industrial provenance [88], [89]. The academic and industrial research lab foundations of a few popular version control and programming language technologies are by now many decades old [90], [91], [92], [93].

Although some were **unaware** of any incorporation of their results into development tools, products, or services [R120,R304], the majority of responses referred to an **indirect research impact**. This includes fundamental research contributions [R79,R80,R94], for instance, to distributed systems [R167], surveys [R91,R96,R304], research methods for conducting computational experiments [R75,R172], guidelines for empirical research [R161,R300,R314], design concepts [R212], as well as contributions that *"do not work on*

real systems" [R185,R186,R315]. As [R216] mentions, having no practical impact is not necessarily *"a bad thing"*, especially when a work influences other researchers, or facilitates the exploration of different fields and the development of techniques.

4.4 RQ4: Practical Impacts of SE Research on Information Technology, Society, and Industry

The authors' responses to the related open-ended question revealed various changes of SE research in the state of practice. Through their **prototypes**, practitioners advanced **production technology** and facilitated the **open source community**. Specifically, they improved practices involving mining design patterns from source code [R48], code reviewing [R222], clone detection [R169], reverse engineering and program understanding [R171], systems development [R90], quality and robustness [R175,R193], testing and analysis [R157,R209,R210], model checking [R159], remodularization optimization [R62], and analogy-based reasoning [R173,R174]. Various prototypes were quickly adopted by open source projects [R164,R195]. Software development companies incorporated researchers' approaches into their processes [R173,R174,R209] and product lines (*e.g.*, [R210]). We also observe adoption by competitors. Although this is *"not exactly desirable from a company perspective, at least the research had an impact"* [R170]. Collaborations with companies benefited both parties, regardless; companies improved their solutions, while individuals promoted their open source tools gaining numerous downloads [R173,R174].

Several attempts were made towards extending the **functionality** of existing open source and proprietary products. Researchers provided solutions to code recommendation projects [R229], algorithms to graph layout tools [R303], support for free text search to code search engines [R166,R196], conflict modeling techniques to commercial tool sets [R190], bidirectional streaming to HTTP/2 and relevant frameworks [R207], multi-threaded software verification methods to state-of-the-art tools [R97], and online tools for visualizing map representations of GitHub code clones [R164]. Version control systems were incorporated into Unix system distributions, eliciting their widespread adoption by universities and technology-leading corporations—*"the trend setters"* [R57]. Moreover, the identification of performance bug patterns and the development of static performance checkers allowed Android developers to *"generate real-time warnings when they are writing code"* [R127], while the construction of rules matrices helped companies *"to differentiate under which legal conditions a dataset can be used for analysis"* [R156].

Along with functionality, practitioners improved products' **design and modeling**, **quality**, and **performance**. We observe new architecture concepts [R206], software modeling languages [R152], hybrid system modeling plug-ins [R154], and project management frameworks [R76]. Support was provided for identifying countermeasure requirements by modeling and analyzing threats at the application level [R187], and for encompassing visual formalisms, including statecharts, in modeling processes [R220]. Quality-wise, some products upgraded the robustness of cloud management platforms [R193], the measurement and decrease of attack surfaces in software systems [R162], the long-term evaluation of regression testing

TABLE 6
Practical Impact of SE Venues

Type	Name	Publisher	Papers	SE Patent Citations	Impact Factor
J	IEEE Trans. Softw. Eng.	IEEE	681	1 988	2.919
C	MSR	IEEE/ACM	469	556	1.185
C	ISSTA	ACM SIGSOFT	435	473	1.087
J	Softw. Pract. Exp.	Wiley	666	709	1.064
C	ICSE	ACM/IEEE	3 367	2 497	0.742
J	IEEE Software	IEEE	1 136	833	0.733
C	FSE	ACM SIGSOFT	1 061	614	0.579
C	ICSME	IEEE	940	511	0.544
C	ASE	IEEE/ACM	1 120	596	0.532
C	SANER	IEEE	746	355	0.476
J	J. Syst. Softw.	Elsevier	1 979	364	0.184
J	Inf. Softw. Technol.	Elsevier	1 136	207	0.182
C	RE	IEEE	683	73	0.107
J	Softw. Syst. Model.	Springer	549	53	0.096
J	Empir. Softw. Eng.	Springer	579	25	0.043

techniques [R155], and the independent verification and validation of proprietary tools [R227]. Performance-wise, researchers accelerated testing [R210] and root cause analysis [R251], and suggested solutions for extending the lifetime of products while reducing their maintenance costs [R197].

Some studies aimed to advance **development processes**, release **new methods**, or educate researchers and improve their **professional practices**. For instance, factor-covering array generation tools impacted statistical testing and analysis [R157], model checking techniques affected code analysis [R159], object-oriented design metrics improved software development feedback [R239], data analysis was enhanced by linking code reviews to commits [R222], analogy-based reasoning improved prediction and estimation [R173], and goal-oriented risk analysis impacted requirements engineering [R188,R189,R190]. In short, new methods entail approaches for developing and refining software development processes [R116,R246,R250,R301,R302,R306] and domain-specific languages [R218], model-based requirements [R184], model checking [R159], formalism-oriented abstraction levels [R138], conflict modeling [R190], syntactic preprocessors for implementing variability [R215], configuration management for component models [R152], fault injection [R193], and accident models for safer systems [R6]. Proposed practices involve abandoning "*information retrieval-based traceability link recovery approaches due to low performance*" [R69] and copy-pasting [R51], favoring "*minimization of coupling*" [R126], carefully testing code segments that are predicted to contain faults [R111], using software metrics to predict programmers' performance [R115], being careful with attack surfaces of software systems [R162], preferring alternative metrics to test coverage [R221], and detecting and mitigating "*architectural mismatch*" [R99].

Although some studies may not have directly changed the state of practice through prototypes and product add-ons, they still influenced **follow-up work** and had a **research impact**. Second-order mutation algorithms affected mutation testing tools [R142], a work "*served as a guideline for reusing components*" [R242], various doctoral dissertations were inspired by research on reverse engineering and program comprehension [R171], invalid techniques deployed in safety-critical systems were revealed [R183], and the effect of design decisions on inspection

performance was evaluated [R316]. Experiments were used as a baseline for subsequent research in aspect-oriented programming [R124], specification models led to further academic research and funding [R214], and some works were "*instrumental in ushering in the current era of data-driven thinking in SE*" [R54]. Finally, some publications led to author **promotion** [R116].

4.5 RQ5: SE Venues with Practice Impact

The patent-based impact factors of the top SE venues are presented in Table 6 in descending order, complemented with the corresponding number of publications and SE patent citations. Journals are denoted with *J* and conferences with *C*. The most impactful venue appears to be the IEEE TSE, followed by the International Conference on Mining Software Repositories (MSR) and the ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA).

4.6 RQ6: Sufficiency of SE Research Funding

For each engineering branch, we obtained the total number of publications and PhD dissertations through the process described in Section 3.8. In this way we retrieved a total of 56 679 software, 28 514 civil, 23 969 mechanical, 22 851 electrical, and 3 138 chemical engineering publications. With regard to dissertations, we identified 55 SE, 302 civil, 2 018 mechanical, 1 861 electrical, and 1 147 chemical engineering records. Overall, SE appears first in publications but last in dissertations.

5 DISCUSSION

More than fifty years after the launch of the first SE conference series, the SE research discipline can be proud of numerous tangible contributions to practice. From the survey analysis it appears that researchers have equipped their industrial partners with a swarm of new open source software tools, novel development processes and methods, and advanced professional practices. In addition, they managed to expand the quality and scope of existing proprietary products, showcasing the importance of maintaining strong academia-industry ties. However, the quantitative results of the patent analysis contradict the self-assessed survey results, since only a limited number of the collected patents cite any

SE paper of the examined venues. Still, almost half (45%) of the patents citing 69% of the SE papers have been renewed, paying out substantial corresponding fees. (The maintenance fees in the US alone range from \$500 to \$7 700.[17]) These facts demonstrate the influence of SE papers on valuable patents.

Introspective SE Research Looking at the practical impact of SE research types, areas, and methods (Section 4.1), we deduce that in a narrow sense the SE field is introspective in its nature. Researchers mostly generate knowledge, tools, and reviews about SE practices and methods to address their own needs [94]. This, by definition, limits the potential footprint of SE research only within its own boundaries, which is typically not the case for other fields such as artificial intelligence and machine learning [95]. Consider, for example, the application of the latter in the petroleum industry [96], biotechnology [97], Internet of Things [98], public administration [99], healthcare [100], and earthquake engineering [95]. As a result, we would expect SE research to reach a narrower industrial audience compared to other science and engineering disciplines.

Yet, in a wider sense SE, by taming the complexity of software development, has allowed the exponential growth [101] of the sophisticated software intensive systems that underpin the modern economy, science, and way of living. This still leaves open the question of the role of SE *research* in this progress.

Cross-disciplinary SE Areas SE practice is impacted by cross-disciplinary SE areas. This is reasonable considering that, according to the Guide to the Software Engineering Body of Knowledge (SWEBOK) [5], SE intersects with diverse computer science areas and other disciplines. A considerable portion of impactful SE research pertains to areas related to programming languages, compilers, and management (Section 4.1). Furthermore, looking at the CPC categories of SE patents citing SE research (Section 3.2) we observe that some of them (G06F8/31, G06F8/37, G06F8/41, G06F8/53) are associated with the aforementioned areas. Previous work has also demonstrated that SE researchers are very interested in human factors [32]. Consequently, it might be worth investigating in isolation the impact of cross-disciplinary and specialized SE areas to SE practice by studying more specialized venues, such as the Conference on Programming Language Design and Implementation, the International Conference on Object-Oriented Programming Systems, Languages, and Applications, the International Conference on Functional Programming, and the Symposium on Principles of Programming Languages.

SE Venues With regard to SE venues, one might wonder whether practitioner-oriented ones are more impactful. We assessed the top SE venues by computing their impact factors on the basis of SE patent citations (Table 6). We observe that practitioner-oriented venues, such as *Software: Practice and Experience* and *IEEE Software*, are higher in the ranking compared to researcher-oriented ones, such as *Empirical Software Engineering*. Moreover, we see a large number of conferences ranked considerably high. This could be an effect of the practitioner-oriented tracks of these conferences including ICSE's *Software Engineering in Practice*, MSR's

Mining Challenge, ISSTA, FSE and ICSME's *Tool Demonstrations*, and FSE and ICSME's *Industry Showcase* [102]. To increase their industrial appeal, less practitioner-oriented journals and conferences might want to consider including dedicated industrial topics, calls, and tracks for contributions, similar to the aforementioned examples.

Correlation Analysis From the patent analysis we obtained no concrete evidence of relationship between patent citations and impact of SE research in practice. Although previous work has demonstrated positive correlation between patent and academic citations [34], our correlation analysis (Section 4.2) did not show a sufficient association between the two. In addition, looking at the most-cited SE papers by SE patents (Table 5) we infer that many (*e.g.*, [70], [74]) are knowledge– rather than solution-seeking [103]. Their main use as citations in patent documents is most likely the provision of background information. Solution-seeking studies usually produce algorithms, models, and tools to cope with *practical problems* [104], whereas knowledge-seeking ones employ cross-disciplinary research methods (*e.g.*, case studies, surveys) to explain *knowledge problems* [104] of SE practice (*e.g.*, to evaluate and compare different tools, or to study developers' collaboration) [105].

Knowledge-seeking SE Research The strong appearance of knowledge-seeking studies in patent citations could be justified by the fact that SE (beyond the study scope—Section 1) is a complex discipline consisting of various dimensions (also *knowledge areas* in SWEBOK [5]). Apart from product-related areas (*e.g.*, Software Construction and Testing), there are also areas associated with the processes, methods, and models employed in the software construction (SE Process, SE Models and Methods), the management and cooperation of the development teams (SE Professional Practice), and the project management of the software development (SE Economics). These topics are highly relevant to the industry [88], and this could be the reason for their high citation numbers. Therefore, the interpretation of the association between patent citations and practical impact depends on the impact definition.

Impact Definition The dependence of our analysis on the impact definition, along with the dissent of some survey respondents from our perceived sense of impact, constitutes a need for a formal term description. From the answers to RQ3 and RQ4 (Sections 4.3, 4.4) we infer that the influence of subsequent research was rated both as practical and nonpractical impact by respondents. To allow future studies building on our work to quantify and assess the impact of SE research in practice, academia and industry need to jointly agree on what constitutes impact based on empirically validated research. In this regard, we propose as term description our impact definition in Section 1.

SE Research Funding One prominent hindrance to the development and commercialization of academic SE research products appears to be insufficient funding (Section 4.3). Although some research ideas may be promising for the industry, they seem to struggle to evolve due to a lack of financial resources. These deficiencies concern both project-based and long-term funding, which is required for a software product's maintenance after the project completion [106]. In addition, trending industrial research topics, such as the metaverse, self-driving cars, space, robotics, and quantum

---

17. https://www.uspto.gov/learning-and-resources/fees-and-payment/uspto-fee-schedule#Patent%20Maintenance%20Fee

computing [31], are often financially unbearable for academia. In 2021, the R&D budget of America's top five tech companies (Amazon, Alphabet, Meta, Apple, Microsoft) added up to 149 billion dollars, almost a quarter of America's 2020 public and private R&D investment (713 billion) [31]. To this end, SE researchers have repeatedly stressed the need for higher financial support by society to be able to perform more realistic studies for the industry (*e.g.*, [107], [108], [109]). Particularly, Sjøberg et al. [109] argued that, given the recognized value of software in business [110], the discipline should not fall back in funding compared to other fields, including natural sciences and medicine. But there is also a matter of priorities resulting from the finite amount of available resources: research is the steering wheel of innovation, but it also needs to cater to financial and societal needs [111].

Due to the lack of empirical data supporting the aforementioned remarks, further analysis is needed to assess the sufficiency of SE funding. To set the path, we approximated funding based on the number of existing SE publications and PhD dissertations, and compared them to those of the main engineering branches (Section 4.6). The resulting data are inconclusive: SE seems to outnumber the other branches in publications, while trailing in dissertations. This can be an effect of the employed search method, leading to false positives or missed records. Or it could be the case that SE researchers can get away with publishing studies of low-hanging fruit, with carefully developed methods and impeccable text, but having limited practical impact—a practice that may not be widely tolerated in more mature engineering disciplines. A comprehensive study of SE research funding could shed light on this issue, and provide well-grounded insights regarding the funding's adequacy.

Tech Transfer Challenges Beyond funding, a set of technology transfer challenges arise from the related work (Section 2) and the survey answers to RQ3 (Section 4.3). These can be categorized into two broad areas: industrial obstacles (*i.e.*, adoption barriers by practice) and research challenges (*i.e.*, limitations of the proposed work). Concerning the first, we observe disregard and questioning of research progress by practice. In a recent (non peer-reviewed) study Koziolek reports that practitioners appear willing to participate in case studies and experiment with new methods from academia, regardless of the return-on-investment [112]. However, according to two recent large-scale developer surveys by JetBrains [88] and Stack Overflow [89], the majority of most-used software development tools in practice are developed in the industry or by practitioners by a vast margin, and any research results they appear to incorporate were developed many decades ago [90], [91], [92], [93]. The reasons for the industrial provenance of tools used by practitioners could be the products' technology readiness [113], stability, usability, trustworthiness, feature customization, exclusive customer support and training, and collaboration ties between companies and software vendors [114], [115]. Furthermore, companies developing their own software products may be reluctant to work with open source software coming outside their organization—also known as the *not invented here syndrome* [116]. These factors hinder the adoption of academic technology by practice.

Regarding research challenges, we notice an unfavorable cost-benefit trade-off of some proposed solutions, despite their high citation numbers and awards. We summarize the following cases: useful ideas that are technology immature [113], offer conceptual software design solutions [117], and require a long time span and re-engineering to apply in the industry [8]; redundant ideas that do not meet industrial needs or quality standards [12]; research methods suffering from confirmation bias and unreported implications [112]; and promising ideas documented in long, inscrutable papers that require hours of reading [21]. Reasons leading to these deficiencies could be the *publish or perish* culture [118] "forcing scientists to produce *publishable* results at all costs" [119], excessive competition over collaboration among researchers [120], and reductionist thinking hindering scaling of complex software systems [55], [121].

Best Practices To address the outlined challenges, several best practices have been proposed in the related literature, aiming to bridge the gap between academia and industry (*e. g.*, [6], [24], [25]). Notable recommendations from Section 2.2 include: grounding research on real-world problems and reducing generalizability issues; publishing more accessible content (*e.g.*, through blogs or videos) and advertising it through social media; producing more actionable empirical studies, for example, studies on development productivity and effort estimation techniques; ensuring open-access availability of research studies; conducting pilot laboratory tests before industrial releases; and organizing regular workshops with the industry. In addition, adopting the open science and ACM Artifact Badging approaches can facilitate the adoption and application of research products by independent, unguided users [112]. Collaboration between academia and software vendors, who already have an established business model, could also ease the integration of research products into existing tools through plug-ins [112]. From a company perspective, practitioners should systematically evaluate and improve methods they have not produced themselves [112]. It might be worth investigating empirically the extent to which such recommended practices have been adopted by industry and academia as well as any factors impeding their application.

6 LIMITATIONS

Here we present the risks resulting from the patent and survey analyses. The survey was designed with the stated goal of examining how SE research has impacted practice. For this purpose, we followed recommended guidelines for survey research [51], [52]. Response options of two out of three multiple choice questions were adapted from established literature [36], [55], and the complete questionnaire was validated through two pilot runs. Although the survey was mainly characterized as **interesting** and potentially **impactful**, its results are limited by the reasons detailed below.

Internal Validity Some risks to the internal validity of the study stem from the manual processes involving subjective judgment. These include the identification of the SE-related CPC categories (Section 3.2), the mapping of patent maintenance fee codes to their fee values (Section 3.4), and the manual coding of the open-ended survey responses. Although biases related to human judgment cannot be completely eradicated [122], we aimed to reduce this threat by employing established methods [41], [56]. Manual

coding also entails loss of accuracy of responses, due to the extreme level of their categorization; we addressed this by assigning multiple (rather than only one) codes to each answer.

Risks also rise from some automated processes we employed. The identification of SE papers cited by SE patents through DOI crosschecking, and title and author mapping is not completely accurate, according to our evaluation (Section 3.3). The identification of surveyed practitioners (Section 3.6) does not account for 118 (20%) papers lacking DOIs. Similarly, the identification of SE venues in patent citations based on their full names, abbreviations, and acronyms (Section 3.7) may have led to some false positives in Table 6. In the same Table, the number of MSR papers does not include records for the years 2010, 2011, and 2018, because these are not tracked by Scopus. Publication and PhD dissertation counts for the engineering disciplines may also include false positives, as confirmed in Section 5. Also, the research funding evaluation (Section 4.6) does not make a distinction between academic and industrial research funding. One might expect the former to be less impactful than the latter. Further analysis is required to assess this difference.

A few limitations are associated with the analysis of SE patent citations of SE papers (Section 3.3). The analysis does not account for instances where an author of a patent may also be involved in the cited publication either as a direct author or as a collaborator of the research team. There may also be cases where authors employ both publication means (scientific publication and patent) and one cites the other to increase the publication's value. Moreover, patent citations were not screened to determine the actual use of SE research. Consequently, citations that may reference a study as an example, comparison, statement source, or related work [123] may have skewed the associated quantitative results in Sections 4.1, 4.2, and discussion comments (Section 5). Although we evaluated patents based on their litigation and maintenance fee events (Section 4.2), this does not guarantee that all patented products are impactful. These cases require special treatment to verify the practical impact.

Further concerns may result from the patent-to-science citation linkages regarding novelty. Novelty can be a deal breaker when preparing a patent application: turning an already published idea into a patent is usually very hard, in many jurisdictions a showstopper [124]. However, to cope with this challenge, it is a recommended practice to adopt the *patent first, publish later* approach [124], according to which researchers file a patent application before they publish it in a research paper. This approach would result in patent-publication chains, where, for example, *patent A* is cited by *related-publication A*, *publication A* is cited by *improvement-patent B*, *patent B* is cited by *related-publication B*, and so forth. The publication-to-patent part of this chain (*i.e.*, *publication A is cited by patent B*) is the center of our patent citation analysis (Section 4.2). As a result, the cited publication and the citing patent can refer to different—yet possibly similar—research ideas. Still, there may also be cases of citations diverging from this approach.

The survey questionnaire is associated with two inherent biases. Social desirability bias [125] (*i.e.*, a respondent's potential tendency to appear in a positive light, for example, by showing they are fair or rational) is a risk associated with the survey responses to RQ3 and RQ4. For instance, one should not over-interpret that the majority (69%) of respondents consider their surveyed work to have impacted either partially or widely SE practice. To mitigate this issue, participants were informed that their responses would be published anonymized. Furthermore, question-order effect [126] (*e.g.*, one question may have provided context for the subsequent one) may have influenced respondents' answers. Although this effect could have been reduced, for example, by shuffling questions, we opted to order them in a rational sequence for participants to recall and comprehend the context of the asked questions.

The collected survey feedback also revealed the following limitations. The definition of practical impact that was provided to candidates was an earlier version of the one introduced in Section 1. This did not successfully clarify that software development tools impacted by research should only be industrial, hence answers to RQ3 and RQ4 may also concern non-industrial tools. In addition, the **subjective** and **limited definition** of the term along with its **quantification process** roused concerns to some respondents who emphasized that impact "*can almost only be assessed by people working in closely related research areas*" [R80]. The classification process using the 2012 ACM CCS was deemed **difficult** and **irrelevant** by some participants. Others advised against using institutional e-mail accounts in surveys, because these are often overloaded with unsolicited messages, and may thus limit response rate. To address this, in our manual search we looked for active e-mail addresses in researchers' personal web pages (Section 3.6). Finally, the **author selection process** was not sufficiently documented in the invitational mail causing confusion to some respondents.

External Validity Generalizability concerns arise from the survey sample selection process (Section 3.6), which is limited to the first authors and the venues of ICSE, TSE, TOSEM, and EMSE, only includes the most-cited, awarded distinguished, and most influential works, and involves a small number of practitioners (10%). Still, these practitioners are affiliated with 36 diverse companies. To counterbalance this shortcoming, we aimed to cover all publication years, motivated survey candidates to include and review additional impactful publications of theirs, and also invited them to list any notable papers of other researchers that may have come to their attention. Although these concerns prevent us from generalizing the survey findings, meaningful insights emerged, which could be amplified through study replication in other research outlets and author sets. Furthermore, the computation of the patent-based impact factors of the top SE venues (Section 3.7) is affected by the restricted year range (2009–2019). As explained, the rationale was to approximate publication counts of all years, preventing any risk that could occur from different venue start dates.

7 CONCLUDING REMARKS

We investigated the impact of SE research in practice through a systematic analysis of science linkages between SE research and SE patents, and a survey on authors of top-notch publications. Specifically, we identified impactful types, areas, and methods of SE research, the outcomes of impactful research, and its main practical impacts on

information technology, society, and industry. We further assessed the impact of SE venues and the sufficiency of SE research funding by comparing it to the main engineering branches'. To address these, we collected 11 419 papers from ICSE, TSE, TOSEM, and EMSE between 1975–2017, and complemented them with their assigned topics, citation counts, and awards. We also retrieved a set of 304 368 SE patents, and found 1 690 papers cited by 4 354 of them. To assess patents' value, we analyzed their litigation cases and maintenance fee events. Through a survey on 475 authors of 593 top-cited and awarded papers (26% response rate), we complemented our study results with quantitative and qualitative insights. For the venues' impact we computed their patent-based impact factors, while for the adequacy of SE research funding we retrieved the research publication and dissertation counts of SE and the main engineering branches. The study's key findings are summarized below.

- SE researchers have equipped practitioners with various tools, processes, and methods, and improved many existing solutions. Moreover, practitioners seem to value knowledge-seeking studies.
- SE practice is impacted by cross-disciplinary SE areas, hence it could be of value to assess this influence by studying in more depth some specialized venues.
- Practitioner-oriented tracks in conferences may enhance their impact. A dedicated study of these tracks could provide more insights as well as useful recommendations to organization and program committees.
- Academia and industry could jointly agree on a formal impact term description based on empirically validated research and backed by key performance indicators to set a common ground for subsequent research on the topic.
- There is a claim for higher funding in SE research, which we cannot corroborate through our analysis on engineering dissertations and publications, or literature search. A comprehensive empirical study could shed light on the matter.

Research Agenda There are various directions for extending the investigation of the SE research impact in practice—some are listed in the preceding paragraphs. Other ideas include: studying the patenting behavior by SE research area, and investigating the practical impact of less patent-focused areas (*e.g.*, agile/lean methods, test automation); conducting in-depth systematic analyses of SE patent citations to delve into the use of academic research by patents; and comparing reports of industry impact (*e.g.*, company statements, software development metrics) with venue, funding, and citation metrics. With these final remarks we aim to steer academia's attention towards some research topics requiring further investigation, and begin a discussion on how we, the researchers, can increase our footprint on practice.

ACKNOWLEDGMENTS

REFERENCES

[1] P. Naur and B. Randell, *Software Engineering: Report of a Conference Sponsored by the NATO Science Committee, Garmisch, Germany, 7–11 Oct. 1968*. Brussels, Scientific Affairs Division, NATO, 1969. Accessed: Jun. 2022. [Online]. Available: http://homepages.cs.ncl.ac.uk/brian.randell/NATO/nato1968.PDF

[2] J. N. Buxton and B. Randell, *Software Engineering Techniques: Report of a Conference Sponsored by the NATO Science Committee, Rome, Italy, 27–31 Oct. 1969*. Brussels, Scientific Affairs Division, NATO, 1970. Accessed: Jun. 2022. [Online]. Available: http://homepages.cs.ncl.ac.uk/brian.randell/NATO/nato1969.PDF

[3] P. Mell and T. Grance, "The NIST definition of cloud computing, NIST SP 800-145," Comput. Secur. Div., Inform. Technol. Lab., Nat. Inst. Stand. Technol., Gaithersburg, MD, USA, 2011. [Online]. Available: https://doi.org/10.6028/NIST.SP.800-145

[4] Statista: Industries: Technology & Telecommunications, "Statistics and market data on software," 2018. Accessed: Jun. 2022. [Online]. Available: https://www.statista.com/markets/418/topic/484/software/

[5] P. Bourque and R. E. Fairley, Eds., *Guide to the Software Engineering Body of Knowledge, Version 3.0*. IEEE Computer Society, 2014. [Online]. Available: www.swebok.org

[6] L. J. Osterweil, L. A. Clarke, M. Evangelist, J. Kramer, D. Rombach, and A. L. Wolf, "The Impact Project (panel session): Determining the impact of software engineering research upon practice," *SIGSOFT Softw. Eng. Notes*, vol. 25, no. 6, pp. 108–109, 2000.

[7] F. Narin, K. S. Hamilton, and D. Olivastro, "The increasing linkage between U.S. technology and public science," *Res. Policy*, vol. 26, no. 3, pp. 317–330, 1997.

[8] J. Estublier et al., "Impact of software engineering research on the practice of software configuration management," *ACM Trans. Softw. Eng. Methodol.*, vol. 14, no. 4, pp. 383–430, 2005.

[9] National Academy ofEngineering, *The Impact of Academic Research on Industrial Performance*. Washington, D.C., USA: National Academies Press, 2003.

[10] J. Bessen and R. M. Hunt, "An empirical look at software patents," *J. Econ. Manag. Strategy*, vol. 16, no. 1, pp. 157–189, 2007.

[11] D. C. Ince, L. Hatton, and J. Graham-Cumming, "The case for open computer programs," *Nature*, vol. 482, no. 7386, pp. 485–488, 2012.

[12] D. Lo, N. Nagappan, and T. Zimmermann, "How practitioners perceive the relevance of software engineering research," in *Proc. 10th Joint Meeting Found. Softw. Eng.*, 2015, pp. 415–425.

[13] L. Osterweil, C. Ghezzi, J. Kramer, and A. Wolf, "Editorial," *ACM Trans. Softw. Eng. Methodol.*, vol. 14, no. 4, pp. 381–382, 2005.

[14] L. Osterweil, C. Ghezzi, J. Kramer, and A. Wolf, "Determining the impact of software engineering research on practice," *Computer*, vol. 41, no. 3, pp. 39–49, 2008.

[15] B. G. Ryder, M. L. Soffa, and M. Burnett, "The impact of software engineering research on modern progamming languages," *ACM Trans. Softw. Eng. Methodol.*, vol. 14, no. 4, pp. 431–477, 2005.

[16] L. A. Clarke and D. S. Rosenblum, "A historical perspective on runtime assertion checking in software development," *SIGSOFT Softw. Eng. Notes*, vol. 31, no. 3, pp. 25–37, 2006.

[17] W. Emmerich, M. Aoyama, and J. Sventek, "The impact of research on the development of middleware technology," *ACM Trans. Softw. Eng. Methodol.*, vol. 17, no. 4, pp. 1–48, 2008.

[18] D. Rombach, M. Ciolkowski, R. Jeffery, O. Laitenberger, F. McGarry, and F. Shull, "Impact of research on practice in the field of inspections, reviews and walkthroughs: Learning from successful industrial uses," *SIGSOFT Softw. Eng. Notes*, vol. 33, no. 6, pp. 26–35, 2008.

[19] B. Boehm and R. Valerdi, "Impact of software resource estimation research on practice: A preliminary report on achievements, synergies, and challenges," in *Proc. 33rd Int. Conf. Softw. Eng.*, 2011, pp. 1057–1065.

[20] A. T. Misirli, B. Caglayan, A. Bener, and B. Turhan, "A retrospective study of software analytics projects: In-depth interviews with practitioners," *IEEE Softw.*, vol. 30, no. 5, pp. 54–61, Sep./Oct. 2013.

[21] S. Beecham, P. OLeary, I. Richardson, S. Baker, and J. Noll, "Who are we doing global software engineering research for?," in *Proc. 8th Int. Conf. Glob. Softw. Eng.*, 2013, pp. 41–50.

[22] V. Ivanov, A. Rogers, G. Succi, J. Yi, and V. Zorin, "What do software engineers care about? Gaps between research and practice," in *Proc. 11th Joint Meeting Found. Softw. Eng.*, 2017, pp. 890–895.

[23] T. Mikkonen, C. Lassenius, T. Männistö, M. Oivo, and J. Järvinen, "Continuous and collaborative technology transfer: Software engineering research with real-time industry impact," *Inf. Softw. Technol.*, vol. 95, pp. 34–45, 2018.

[24] V. Garousi, K. Petersen, and B. Ozkan, "Challenges and best practices in industry-academia collaborations in software engineering: A systematic literature review," *Inf. Softw. Technol.*, vol. 79, pp. 106–127, 2016.

[25] V. Garousi et al., "Characterizing industry-academia collaborations in software engineering: Evidence from 101 projects," *Empir. Softw. Eng.*, vol. 24, no. 4, pp. 2540–2602, 2019.

[26] B. Vasilescu, A. Serebrenik, T. Mens, M. G. van den Brand, and E. Pek, "How healthy are software engineering conferences?," *Sci. Comput. Prog..*, vol. 89, pp. 251–272, 2014.

[27] R. L. Glass, "An assessment of systems and software engineering scholars and institutions," *J. Syst. Softw.*, vol. 27, no. 1, pp. 63–67, 1994.

[28] W. E. Wong, T. Tse, R. L. Glass, V. R. Basili, and T. Y. Chen, "An assessment of systems and software engineering scholars and institutions (2003–2007 and 2004–2008)," *J. Syst. Softw.*, vol. 84, no. 1, pp. 162–168, 2011.

[29] J. Ren and R. N. Taylor, "Automatic and versatile publications ranking for research institutions and scholars," *Commun. ACM*, vol. 50, no. 6, pp. 81–85, 2007.

[30] D. Parnas, "Stop the numbers game," *Commun. ACM*, vol. 50, pp. 19–21, 2007.

[31] *What America's Largest Technology Firms are Investing in: Their Focus is on the Metaverse, Cars and Health Care*. London, U.K.: The Economist, 2022.

[32] K. Petersen and N. B. Ali, "An analysis of top author citations in software engineering and a comparison with other fields," *Scientometrics*, vol. 126, no. 11, pp. 9147–9183, 2021.

[33] J. P. A. Ioannidis, K. W. Boyack, and J. Baas, "Updated science-wide author databases of standardized citation indicators," *PLOS Biol.*, vol. 18, no. 10, pp. 1–3, 2020.

[34] A. Agrawal and R. Henderson, "Putting patents in context: Exploring knowledge transfer from MIT," *Manag. Sci.*, vol. 48, no. 1, pp. 44–60, 2002.

[35] J. Hamilton, *The engineering profession*. London, U.K.: UK Engineering Council, 2000.

[36] B. Rous, "Major update to ACM's Computing Classification System," *Commun. ACM*, vol. 55, no. 11, pp. 12–12, 2012.

[37] C. Zheng, G. He, and Z. Peng, "A study of web information extraction technology based on Beautiful Soup," *J. Comput.*, vol. 10, no. 6, pp. 381–387, 2015.

[38] Z. Griliches, "Patent statistics as economic indicators: A survey," *J. Econ. Lit.*, vol. 28, no. 4, pp. 1661–1707, 1990.

[39] M. Blackman, "News from the USPTO," *World Pat. Informat.*, vol. 34, no. 4, pp. 333–335, 2012.

[40] European Patent Office, "The cooperative patent classification: Introduction to the CPC," 2013. Accessed: Jun. 2022. [Online]. Available: https://e-courses.epo.org/wbts/cpc_general/index.html

[41] P. Brereton, B. A. Kitchenham, D. Budgen, M. Turner, and M. Khalil, "Lessons from applying the systematic literature review process within the software engineering domain," *J. Syst. Softw.*, vol. 80, no. 4, pp. 571–583, 2007.

[42] IFI CLAIMS Patent Services and Google, "Google patents public data," 2017. Accessed: Jun. 2022. [Online]. Available: https://console.cloud.google.com/marketplace/product/google_patents_public_datasets/google-patents-public-data

[43] Organisation for Economic Co-operation and Development, "OECD patent statistics manual, chapter 6: The use and analysis of citations in patents," OECD Publishing, Paris, 2009. [Online]. Available: https://doi.org/10.1787/9789264056442-7-en

[44] S. Nagaoka, "Assessing the R&D management of a firm in terms of speed and science linkage: Evidence from the US patents," *J. Econ. Manag. Strategy*, vol. 16, no. 1, pp. 129–156, 2007.

[45] W. G. Cochran, *Sampling Techniques*, 3rd ed. Hoboken, NJ, USA: Wiley, 1977.

[46] J. Bessen, "The value of U.S. patents by owner and patent characteristics," *Res. Policy*, vol. 37, no. 5, pp. 932–945, 2008.

[47] A. C. Marco, A. Tesfayesus, and A. A. Toole, "Patent litigation data from US district court electronic records (1963–2015)," USPTO Economic Working Paper No. 2017-06, 2017. [Online]. Available: https://doi.org/10.2139/ssrn.2942295

[48] D. L. Schwartz, T. M. Sichelman, and R. Miller, "USPTO patent number and case code file dataset documentation," USPTO Economic Working Paper No. 2019-05, Northwestern Law & Econ Research Paper No. 19-12, San Diego Legal Studies Paper No. 20-435, 2019. [Online]. Available: https://doi.org/10.2139/ssrn.3507607

[49] R. D'Agostino and E. S. Pearson, "Tests for departure from normality. Empirical results for the distributions of b_2 and $\sqrt{b_1}$," *Biometrika*, vol. 60, no. 3, pp. 613–622, 1973.

[50] C. Spearman, "The proof and measurement of association between two things," *Am. J. Psychol.*, vol. 15, no. 1, pp. 72–101, 1904.

[51] S. L. Pfleeger and B. A. Kitchenham, "Principles of survey research: Part 1: Turning lemons into lemonade," *SIGSOFT Softw. Eng. Notes*, vol. 26, no. 6, pp. 16–18, 2001.

[52] B. Kitchenham and S. L. Pfleeger, "Principles of survey research: Part 6: Data analysis," *SIGSOFT Softw. Eng. Notes*, vol. 28, no. 2, pp. 24–27, 2003.

[53] B. A. Kitchenham and S. L. Pfleeger, "Principles of survey research: Part 2: Designing a survey," *SIGSOFT Softw. Eng. Notes*, vol. 27, no. 1, pp. 18–20, 2002.

[54] M. E. Rose and J. R. Kitchin, "pybliometrics: Scriptable bibliometrics using a Python interface to Scopus," *SoftwareX*, vol. 10, 2019, Art. no. 100263.

[55] S. Easterbrook, J. Singer, M.-A. Storey, and D. Damian, *Selecting Empirical Methods for Software Engineering Research*. Berlin, Germany: Springer, 2008, pp. 285–311.

[56] J. M. Corbin and A. Strauss, "Grounded theory research: Procedures, canons, and evaluative criteria," *Qual. Sociol.*, vol. 13, no. 1, pp. 3–21, 1990.

[57] J. Linåker, S. M. Sulaman, R. Maiani de Mello, and M. Höst, *Guidelines for Conducting Surveys in Software Engineering*, Dep. Comput. Sci., Lund Univ., Lund, Sweden, 2015.

[58] Clarivate, "Journal citation reports: Document types included in the impact factor calculation," *Sci. Acad. Res.*, 2018. Accessed: Jun. 2022. [Online]. Available: https://support.clarivate.com/ScientificandAcademicResearch/s/article/Journal-Citation-Reports-Document-Types-Included-in-the-Impact-Factor-Calculation

[59] S. Bains, *Electronic Theses and Dissertations (ETDs) Resources*. Lucknow, Uttar Pradesh: BFC Publications, 2022, pp. 213–222.

[60] Networked Digital Library of Theses and Dissertations (NDLTD), Thesis resources: Find ETDs. Accessed: Jun. 2022. [Online]. Available: https://ndltd.org/thesis-resources/find-etds/

[61] F. A. Loan, U. Y. Parray, and A. M. Khan, "Performance of search engines in harvesting grey literature," *Libr. Hi Tech News*, vol. 39, no. 2, pp. 21–23, 2022.

[62] J. A. Wani, "Open access electronic thesis and dissertation repositories: An assessment," *Libr. Philos. Pract. e-J.*, no. 2528, pp. 1–11, 2019.

[63] M. Coates, "Electronic theses and dissertations: Differences in behavior for local and non-local users," *Libr. Hi Tech*, vol. 32, no. 2, pp. 285–299, 2014.

[64] C. J. Greenberg, "Opening cultural heritage in the age of OAI-PMH: Finding Armenia in the OATD discovery service," *Libr. Manag.*, vol. 35, no. 4/5, pp. 320–328, 2014.

[65] A. Ali and S. Jan, "Contribution of UK and USA towards open access theses and dissertations (OATD)," in *Proc. Int. Conf. Knowl. Manage. Higher Educ. Inst.*, 2021, pp. 309–316.

[66] T. Ferreras-Fernández, F. García-Penalvo, J. A. Merlo-Vega, and H. Martín-Rodero, "Providing open access to PhD theses: Visibility and citation benefits," *Program*, vol. 50, no. 4, pp. 399–416, 2016.

[67] S. Hangal and M. S. Lam, "Tracking down software bugs using automatic anomaly detection," in *Proc. 24th Int. Conf. Softw. Eng.*, 2002, pp. 291–301.

[68] G. Almes, A. Black, C. Bunje, and D. Wiebe, "EDMAS: A locally distributed mail system," in *Proc. 7th Int. Conf. Softw. Eng.*, 1984, pp. 56–66.

[69] J. Zhang and B. H. C. Cheng, "Model-based development of dynamically adaptive software," in *Proc. 28th Int. Conf. Softw. Eng.*, 2006, pp. 371–380.

[70] A. Dearle, "Software deployment, past, present and future," in *Proc. 29th Int. Conf. Softw. Eng.*, 2007, pp. 269–284.

[71] G. Kiczales, "Aspect-oriented programming," in *Proc. 27th Int. Conf. Softw. Eng.*, 2005, pp. 730–730.

[72] M. Weiser, "Program slicing," *IEEE Trans. Softw. Eng.*, vol. SE-10, no. 4, pp. 352–357, Jul. 1984.

[73] G. T. Almes, A. P. Black, E. D. Lazowska, and J. D. Noe, "The Eden system: A technical review," *IEEE Trans. Softw. Eng.*, vol. 11, no. 1, pp. 43–59, Jan. 1985.

[74] B. W. Boehm, "Software engineering economics," *IEEE Trans. Softw. Eng.*, vol. 10, no. 1, pp. 4–21, Jan. 1984.

[75] A. Black, N. Hutchinson, E. Jul, H. Levy, and L. Carter, "Distribution and abstract types in Emerald," *IEEE Trans. Softw. Eng.*, vol. SE-13, no. 1, pp. 65–76, Jan. 1987.

[76] R. S. Hall, D. Heimbigner, and A. L. Wolf, "A cooperative approach to support software deployment using the software dock," in *Proc. 21st Int. Conf. Softw. Eng.*, 1999, pp. 174–183.

[77] B. Korel, "Automated software test data generation," *IEEE Trans. Softw. Eng.*, vol. 16, no. 8, pp. 870–879, Aug. 1990.

[78] P. Hosek and C. Cadar, "Safe software updates via multi-version execution," in *Proc. 35th Int. Conf. Softw. Eng.*, 2013, pp. 612–621.

[79] A. T. T. Ying, G. C. Murphy, R. Ng, and M. C. Chu-Carroll, "Predicting source code changes by mining change history," *IEEE Trans. Softw. Eng.*, vol. 30, no. 9, pp. 574–586, Sep. 2004.

[80] R. J. Hall, "Call path profiling," in *Proc. 14th Int. Conf. Softw. Eng.*, 1992, pp. 296–306.

[81] D. Cubranić and G. C. Murphy, "Hipikat: Recommending pertinent software development artifacts," in *Proc. 25th Int. Conf. Softw. Eng.*, 2003, pp. 408–418.

[82] T. G. Moher, "PROVIDE: A process visualization and debugging environment," *IEEE Trans. Softw. Eng.*, vol. 14, no. 6, pp. 849–857, Jun. 1988.

[83] M. Fowler, "Refactoring," in *Proc. 24th Int. Conf. Softw. Eng.*, 2002, pp. 701–701.

[84] R. A. Ballance, S. L. Graham, and M. L. Van De Vanter, "The pan language-based editing system," *ACM Trans. Softw. Eng. Methodol.*, vol. 1, no. 1, pp. 95–127, 1992.

[85] A. D. Lucia, F. Fasano, R. Oliveto, and G. Tortora, "Recovering traceability links in software artifact management systems using information retrieval methods," *ACM Trans. Softw. Eng. Methodol.*, vol. 16, no. 4, pp. 13–es, 2007.

[86] D. E. Denning, "An intrusion-detection model," *IEEE Trans. Softw. Eng.*, vol. 13, no. 2, pp. 222–232, Feb. 1987.

[87] K. Krippendorff, *Content Analysis: An Introduction to Its Methodology*, 3rd ed. Newbury Park, CA, USA: SAGE Publications, 2012.

[88] JetBrains, The state of developer ecosystem 2021: Team tools. Accessed: Jun. 2022. [Online]. Available: https://www.jetbrains.com/lp/devecosystem-2021/team-tools/

[89] Stack Overflow, Stack overflow developer survey 2021. Accessed: Jun. 2022. [Online]. Available: https://insights.stackoverflow.com/survey/2021/

[90] M. J. Rochkind, "The source code control system," *IEEE Trans. Softw. Eng.*, vol. SE-1, no. 4, pp. 255–265, Dec. 1975.

[91] W. F. Tichy, "Design, implementation, and evaluation of a revision control system," in *Proc. 6th Int. Conf. Softw. Eng.*, 1982, pp. 58–67.

[92] D. M. Ritchie, S. C. Johnson, M. E. Lesk, and B. W. Kernighan, "The C programming language," *Bell Syst. Tech. J.*, vol. 57, no. 6, pp. 1991–2019, 1978.

[93] B. Stroustrup, "Adding classes to the C language: An exercise in language evolution," *Softw. Pract. Exp.*, vol. 13, no. 2, pp. 139–161, 1983.

[94] E. Schwitzgebel, "Introspection," in *Stanford Encycl. of Philos.*, E. N. Zalta Ed., Metaphysics Research Lab, Stanford University, 2019. Accessed: Jun. 2022. [Online]. Available: https://plato.stanford.edu/archives/win2019/entries/introspection/

[95] Y. Xie, M. E. Sichani, J. E. Padgett, and R. DesRoches, "The promise of implementing machine learning in earthquake engineering: A state-of-the-art review," *Earthq. Spectra*, vol. 36, no. 4, pp. 1769–1801, 2020.

[96] H. Rahmanifard and T. Plaksina, "Application of artificial intelligence techniques in the petroleum industry: A review," *Artif. Intell. Rev.*, vol. 52, no. 4, pp. 2295–2318, 2018.

[97] A. L. Oliveira, "Biotechnology, big data and artificial intelligence," *Biotechnol. J.*, vol. 14, no. 8, 2019, Art. no. 1800613.

[98] F. Al-Turjman, Ed., *Artificial Intelligence in IoT*. Berlin, Germany: Springer, 2019.

[99] G. N. Kouziokas, "The application of artificial intelligence in public administration for forecasting high crime risk transportation areas in urban environment," *Transp. Res. Procedia*, vol. 24, pp. 467–473, 2017.

[100] K.-H. Yu, A. L. Beam, and I. S. Kohane, "Artificial intelligence in healthcare," *Nat. Biomed. Eng.*, vol. 2, no. 10, pp. 719–731, 2018.

[101] L. Hatton, D. Spinellis, and M. van Genuchten, "The long-term growth rate of evolving software: Empirical results and implications," *J. Softw.: Evol. Process*, vol. 29, no. 5, 2017, Art. no. e1847.

[102] V. Vyatkin, "Software engineering in industrial automation: State-of-the-art review," *IEEE Trans. Industr. Inform.*, vol. 9, no. 3, pp. 1234–1249, Aug. 2013.

[103] K.-J. Stol, M. Goedicke, and I. Jacobson, "Introduction to the special section—General theories of software engineering: New advances and implications for research," *Inf. Softw. Technol.*, vol. 70, no. C, pp. 176–180, 2016.

[104] R. Wieringa, "Design science as nested problem solving," in *Proc. 4th Int. Conf. Des. Sci. Res. Informat. Syst. Technol.*, 2009, pp. 1–12.

[105] K.-J. Stol and B. Fitzgerald, "The ABC of software engineering research," *ACM Trans. Softw. Eng. Methodol.*, vol. 27, no. 3, pp. 1–51, 2018.

[106] C. Haupt, T. Schlauch, and M. Meinel, "The software engineering initiative of DLR: Overcome the obstacles and develop sustainable software," in *Proc. 13th Int. Workshop Softw. Eng. Sci.*, 2018, pp. 16–19.

[107] D. I. K. Sjøberg et al., "Conducting realistic experiments in software engineering," in *Proc. Int. Symp. Empir. Softw. Eng.*, 2002, pp. 17–26.

[108] B. Kitchenham, T. Dyba, and M. Jorgensen, "Evidence-based software engineering," in *Proc. 26th Int. Conf. Softw. Eng.*, 2004, pp. 273–281.

[109] D. I. K. Sjøberg, T. Dyba, and M. Jorgensen, "The future of empirical methods in software engineering research," in *Proc. Future Softw. Eng.*, 2007, pp. 358–378.

[110] G. Booch, "Developing the future," *Commun. ACM*, vol. 44, no. 3, pp. 118–121, 2001.

[111] C. Bloch and M. P. Sørensen, "The size of research funding: Trends and implications," *Sci. Public Policy*, vol. 42, no. 1, pp. 30–43, 2014.

[112] H. Koziolek, "Tracing the practical impact of software architecture research," 2022. Accessed: Jun. 2022. [Online]. Available: https://medium.com/@heiko.koziolek/tracing-the-practical-impact-of-software-architecture-research-a2b91136455

[113] M. Héder, "From NASA to EU: The evolution of the TRL scale in Public Sector Innovation," *The Innov. J.*, vol. 22, no. 2, pp. 1–23, 2017.

[114] A. Singh, R. Bansal, and N. Jha, "Open source software versus proprietary software," *Int. J. Comput. Appl.*, vol. 114, no. 18, pp. 26–31, 2015.

[115] S. Dhir and S. Dhir, "Adoption of open-source software versus proprietary software: An exploratory study," *Strateg. Change*, vol. 26, no. 4, pp. 363–371, 2017.

[116] H. Piezunka and L. Dahlander, "Distant search, narrow attention: How crowding alters organizations' filtering of suggestions in crowdsourcing," *Acad. Manag. J.*, vol. 58, no. 3, pp. 856–880, 2015.

[117] European Parliament, Council of the European Union, "Directive 2009/24/EC of the european parliament and of the council of 23 april 2009 on the legal protection of computer programs," *Official J. Eur. Union*, vol. L 111/16, 2009. [Online]. Available: https://eur-lex.europa.eu/eli/dir/2009/24/oj

[118] "Publish or perish," *Nature*, vol. 467, no. 7313, pp. 252–252, 2010.

[119] D. Fanelli, "Do pressures to publish increase scientists' bias? An empirical support from us states data," *PLOS ONE*, vol. 5, no. 4, pp. 1–7, 2010.

[120] P. van den Besselaar, S. Hemlin, and I. van der Weijden, "Collaboration and competition in research," *High. Educ. Policy*, vol. 25, no. 3, pp. 263–266, 2012.

[121] J. Rikkila, P. Abrahamsson, and X. Wang, "The implications of a complexity perspective for software engineering practice and research," *J. Comput. Eng. Inf. Technol.*, vol. 1, no. 1, 2012, Art. no. 2.

[122] K. Petersen, S. Vakkalanka, and L. Kuzniarz, "Guidelines for conducting systematic mapping studies in software engineering: An update," *Inf. Softw. Technol.*, vol. 64, pp. 1–18, 2015.

[123] Z. Kotti, K. Kravvaritis, K. Dritsa, and D. Spinellis, "Standing on shoulders or feet? An extended study on the usage of the MSR data papers," *Empir. Softw. Eng.*, vol. 25, no. 5, pp. 3288–3322, 2020.

[124] V. Mohan-Ram, "Patent first, publish later: How not to ruin your chances of winning a patent," *Science*, 2001. Accessed: Sep. 2022. [Online]. Available: https://www.science.org/content/article/patent-first-publish-later-how-not-ruin-your-chances-winning-patent

[125] A. Furnham, "Response bias, social desirability and dissimulation," *Pers. Individ. Differ.*, vol. 7, no. 3, pp. 385–400, 1986.

[126] L. Sigelaman, "Question-order effects on presidential popularity," *Public Opin. Q.*, vol. 45, no. 2, pp. 199–207, 1981.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/csdl.