

# Research in software engineering: an analysis of the literature

R.L. Glass<sup>a,\*</sup>, I. Vessey<sup>b,1</sup>, V. Ramesh<sup>b,2</sup>

<sup>a</sup>*Computing Trends, 1416 Sare Road, Bloomington, IN 47401, USA*

<sup>b</sup>*Kelley School of Business, Indiana University, Bloomington, IN 47401, USA*

Received 27 October 2001; revised 18 February 2002; accepted 20 March 2002

## Abstract

In this paper, we examine the state of software engineering (SE) research from the point of view of the following research questions:

1. What topics do SE researchers address?
2. What research approaches do SE researchers use?
3. What research methods do SE researchers use?
4. On what reference disciplines does SE research depend?
5. At what levels of analysis do SE researchers conduct research?

To answer those questions, we examined 369 papers in six leading research journals in the SE field, answering those research questions for each paper.

From that examination, we conclude that SE research is diverse regarding topic, narrow regarding research approach and method, inwardly-focused regarding reference discipline, and technically focused (as opposed to behaviorally focused) regarding level of analysis.

We pass no judgment on the SE field as a result of these findings. Instead, we present them as groundwork for future SE research efforts.  
© 2002 Published by Elsevier Science B.V.

*Keywords:* Topic: computing research; Research approach: evaluative-other; Research method: literature analysis; Reference discipline: not applicable; Level of analysis: profession

## 1. Introduction

Over the years, software engineering (SE) research has been criticized from several different points of view—that it is immature [26], that it lacks important elements such as evaluation [31,35], that it is unscientific in its approaches [7]. There have even been attacks on the very foundations of SE research—that it advocates more than it evaluates [24]; that it is, in fact, an endeavor in crisis [13].

Most of those criticisms and attacks have been supported by appropriate research. For example, claims of immaturity are accompanied by a deep analysis of the progress made by more mature research fields; claims of failure to evaluate are accompanied by analysis of the relevant literature to see if software engineering papers include an evaluative compo-

nent; and claims of advocacy are accompanied by at least quotations from papers that do precisely that.

But that research into software research has, ironically, been narrow in its focus, with its purpose more often being to explore and advocate a point of view than to investigate the depth and breadth of the field. That is, that research could be criticized for the very things for which it has criticized software engineering research in general! Perhaps it is time to step back from this melee and take an objective and unbiased view of what SE research actually is and does.

The SE field is arguably less than four decades old. Practitioners have been developing software for longer than that, of course. Land [21] traces that history back to the early 1950s, fully 50 years ago. But in academe, software engineering is a somewhat newer field. Its first conferences were held in the late 1960s, and its academic presence did not begin to separate off from computer science until the early 1980s.

Research into SE tends to track with the academic history of the field. There was SE research in the

\* Corresponding author. Tel.: +1-812-337-8047.

*E-mail addresses:* rlglass@acm.org (R.L. Glass), ivessey@indiana.edu (I. Vessey), venkat@indiana.edu (V. Ramesh).

<sup>1</sup> Tel.: +1-812-855-3485.

<sup>2</sup> Tel.: +1-812-855-2641.

early-practice days of the field, in the 1950s, but it tended to be ad hoc and there were few outlets for publishing the findings. Interestingly, an impressive number of findings were made. Data abstraction and information hiding were used in practice (by other names) in the 1950s [10], even though they were not described in the literature for 10 years afterwards. The traceable history of the field of software research effectively dates, however, to the late 1960s. It was then that journals began to appear that published SE research findings. For example, the early issues of *Communications of the ACM*, which most would say is primarily a computer science journal, carried articles on the subject of ‘Pracniques’, practical techniques of value to SE practitioners, and had departments specifically devoted to the common application domains of the time (e.g. Scientific Applications, Business Applications, Medical Applications, etc.).

It awaited the establishment of professional societies like the ACM and IEEE Computer Society to provide a platform for the publication of SE research results. Journals devoted to the subject, like ACM’s *Transactions on Software Engineering and Methodologies (TOSEM)*, and *Software Engineering Notes (SEN)*, and IEEE’s *Transactions on Software Engineering (TSE)*, and *Software*, opened the door to the publication of increasingly SE-focused research. The practice of SE continued to evolve during this time, and the academic aspects of SE and its research thrived.

With the advent of academic programs devoted to SE (e.g. Wang Institute and Seattle University, both of which established SE graduate programs in the early 1980s), SE research took another significant step forward. Arguably, today’s SE research literature is as rich in content and diversity as that of any other part of the computing field.

At the same time as SE research and practice were evolving, the computing usage world was becoming increasingly dependent on software systems for an enormous variety of tasks, ranging from information management to scientific calculation to entertainment/education. The great strides in both the theory and practice of SE facilitated this growth; today’s software systems are at least 50 times more complex than their early predecessors, and are arguably more reliable. There are few facets of human life in at least first-world countries that are not touched by software systems.

That is the historic background within which we conceived this research, and this paper.

The paper proceeds as follows. Section 2 presents a summary of related research, then describes our own research approach and methods. Section 3 presents the classification schemes used to assess each of our characteristics and details of our classification process. In Section 4, we present the findings for each of our research questions. The paper concludes with a discussion of our findings, the study’s limitations, and the implications of our findings.

## 2. Software engineering research studies

A number of studies have examined the existing state of the SE field and its research. We first examine those studies and then describe our own research approach.

### 2.1. Prior studies

The earliest attempts to define the field of SE were done primarily for pedagogic purposes. The previously-mentioned SE graduate programs conducted and published studies on the principal topics of the field. Wang Institute, Seattle University, and to some extent Southern Methodist University, made major contributions to the SE literature of the time as they struggled to define curriculum topics and content for their programs. Soon thereafter, the Software Engineering Institute of Carnegie Mellon University joined the fray, doing foundation work in not only characterizing the field, but in providing curriculum materials to help in its teaching.

As those efforts began to bear fruit, formal definitions of proposed curricula began to appear, not just for SE but for all three of the major computing fields—Computer Science (CS), Information Systems (IS), and Software Engineering (SE). As each field began to formalize its curriculum, an early paper comparing those curricula [14] found the differences to be significant, and the overlaps to be minimal, implying that the three fields were doing a good job of distinguishing their content.

But perhaps the most significant effort in trying to define the totality of the field of SE itself is the ongoing Software Engineering Body of Knowledge (SWEBOK) project [29]. Its Guide to the Software Engineering Body of Knowledge is still a work in progress. Its current version, as this paper is being written, is Version .9, meaning that it is not yet a final product, which is expected to be Version 1.0. This is a huge endeavor, utilizing the input of experts. Approximately 500 reviewers from 42 countries have critiqued its evolving content. In the end, its 10 Knowledge Areas (KAs) have each been written by one or a few experts in the field.

Because the KAs form a classification scheme for the field, and—as you will see—this paper is among other things about defining such schemes, we list them here for interest:

---

|                       |  |
|-----------------------|--|
| Software requirements | Software configuration management      |
| Software design       | Software engineering management        |
| Software construction | Software engineering process           |
| Software testing      | Software engineering tools and methods |
| Software maintenance  | Software quality                       |

---

Clearly, then, the SWEBOK KAs are primarily defined by the software life cycle.

Another interesting facet of the SWEBOK study is that it provides a classification scheme of what the study calls ‘Related disciplines,’ disciplines that in

one way or another have a relationship with the field of software engineering. They are:

|                                      |                    |
|--------------------------------------|--------------------|
| Cognitive sciences and human factors | Mathematics        |
| Computer engineering                 | Project management |
| Computer science                     | Engineering        |
| Management and management science    |                    |

These related disciplines are interestingly similar, at least in concept, to the notion of Reference disciplines, which we will present later in this paper.

It is worthwhile to note that the SWEBOK effort, while important and comprehensive, is not specifically about the research of the field of SE, but rather about the totality of its body of knowledge. There have been other prior studies specifically about the nature of research in SE. For example, Shaw has conducted a series of studies of the history of SE in the context of the more traditional engineering fields. Addressing the field in general, and not just its research, she found that ‘software engineering is not yet a true engineering discipline, but it has the potential to become one’ [27]. She predicts that the maturity of SE will depend on a number of things, including evolving ‘professional specializations’ (presumably facilitated by focused research) and ‘improving the coupling between science (presumably in the form of research) and commercial practice.’

More recently, Shaw [26] speaks of the ‘coming of age’ of the field, particularly in the context of ‘software architecture research’ in which Shaw has been a pioneer. In that paper she provides several interesting classifications to help characterize the field, including the following:

|                       |   |
|-----------------------|---|
| Research setting      | Feasibility, characterization, method/means, generalization, and selection                          |
| Research product      | Qualitative or descriptive model, technique, system, empirical predictive model, and analytic model |
| Validation techniques | Persuasion, implementation (proof of concept), evaluation, analysis, and experience                 |

In her conclusion to that paper, she provides an interesting summary of the state of SE research, which she calls a ‘challenge to the whole of software engineering.’

Software engineering does not yet have a widely-recognized and widely-appreciated set of research paradigms in the way that other parts of computer science do. That is, we don’t recognize what our research strategies are and how they establish their results.

Our study attempts to provide some insights into the research strategies employed by SE researchers.

Zelkowitz and his colleagues in and around the University of Maryland Computer Science department have also performed interesting studies of SE research. Their particu-

lar concern has been the lack of ‘experimental validation’ in the SE research field. Zelkowitz and Wallace [35], for example, conclude, based on a study of the literature of the field, that ‘validation was generally insufficient’ and that ‘other fields do a better job of validating scientific claims.’ In the process of doing that work, they provide a ‘classification of software engineering papers’ that is similar to some of the notions of research method we have evolved in this paper:

|                  |                    |                   |
|------------------|--------------------|-------------------|
| Replicated       | Project monitoring | Literature search |
| Synthetic        | Case study         | Legacy data       |
| Dynamic analysis | Assertion          | Lessons learned   |
| Simulation       | Field study        | Static analysis   |

Following that thrust, further research by Zelkowitz et al. [34] explores the interaction between research into new SE concepts, and practitioners engaging in technology transfer of such concepts.

Like Zelkowitz, Tichy and his colleagues have done some important studies of the use of ‘experimental evaluation’ in the research of the SE field [31]. This pioneering article sounded the alarm regarding this particular deficit of SE research, and became the platform from which several similar studies have been undertaken. They attribute the problem as much to computer science research as to SE research, and conclude that continuing to underutilize experimental evaluation may be ‘extremely damaging’ to the field.

Another facet of SE research—its relationship with IS research—has been explored by Harrison and Wells [18], Morrison and George [22], and Gregg et al. [15]. The former paper focused on the research approaches and methods employed by the two disciplines, via a ‘meta-analysis’ of papers published in 19 SE and IS journals. They found that differences in terminology between the two fields made it ‘difficult to draw comparisons’ between them. However, they note that SE research ‘does not usually consider social and organizational aspects,’ whereas IS research does. The second SE/IS paper pursued the role played by SE research in IS research. To accomplish that, the authors examined papers in three journals generally considered to be among the most respected in the IS field: Communications of the ACM, which is, of course, also considered to be a leading CS and SE journal; Management Science; and MIS Quarterly. It found that nearly half (45%) of the IS articles in those journals over a six-year period involved ‘SE-related research.’ The latter paper proposes a framework for conducting SE Research within the context of Information Systems with the objective of issuing standards. The authors examined SE-related articles in IS journals and found that most articles measure up well with respect to their framework.

In summary, we see then that the SE research field has been studied in the past, but only by examining certain facets of it (e.g. architecture, experimental evaluation, or its relationship with IS research). More comprehensive

studies of the SE field have focused more on the content and pedagogy of the field, rather than its research. As a result, we believe that the work presented in this paper, which examines and characterizes the **whole of SE research**, is important and timely.

## 2.2. The current study

Our goal in this study is to report facts pertaining to the SE research field. Our examination is, therefore, intended to be descriptive rather than normative.

While many studies have addressed various aspects of SE research, there has been no comprehensive evaluation of the field. To conduct this evaluation, here is what we did; each of these bullets will be discussed in more depth in what follows:

1. Our plan was to define the nature of SE research by the research papers published in the SE field.
2. We performed that study over a five-year period, 1995–1999.
3. We identified the key journals of the SE field, those in which we would examine research papers.
4. We defined categories into which we would classify each paper.
5. In order to utilize these categories, we defined classification schemes for each of them.
6. We decided how many papers we would examine in the five year period.
7. We examined those papers, classifying them according to the categories defined.
8. We analyzed the data so obtained, and wrote the paper.

### 2.2.1. Research questions

To define our view of the field of software engineering research, we addressed the following research questions:

1. What topics do SE researchers address?
2. What research approaches do SE researchers use?
3. What research methods do SE researchers use?
4. What reference disciplines do SE researchers use as the theoretical basis for their publications?
5. At what levels of analysis do SE researchers conduct research?

### 2.2.2. Selection of papers

To get a significant and current snapshot of the field of software engineering research, we determined to conduct the study over a five-year period from 1995 to 1999. We examined every fifth paper in each journal during the time period in question resulting in our classifying 369 papers. We considered the examination of every fifth paper an appropriate randomization technique, designed to reduce the magnitude of the number of papers examined while still using a set of papers that suitably represented the

field in a bias-free manner. Other randomization techniques might have been used, but there is no reason to believe that they would have been more appropriate.

We examined the papers published in key journals of the software engineering literature during that five-year period. The software engineering journals chosen were selected because they have been used over the years for another study regarding the Top Scholars and Top Institutions in the field of systems/software engineering (see, for example, Ref. [9]). They are:

|       |  |
|-------|--|
| IST   | Information and Software Technology                      |
| JSS   | Journal of Systems and Software                          |
| SPE   | Software Practice and Experience                         |
| SW    | IEEE Software  |
| TOSEM | ACM Transactions on Software Engineering and Methodology |
| TSE   | IEEE Transactions on Software Engineering                |

It would have been possible to include additional journals. However, this set of six journals was originally selected by a vote of senior software engineering academics and practitioners, who could nominate any journals they wished to represent the field; its usage over an eight-year period since then represents a stable view of the field; and there are no newer journals that are generally considered to have been of sufficient stature to replace or supplement the list.

### 2.2.3. Categories and classification schemes

We felt it was important to define in advance the categories into which we would classify the papers. We viewed this as a top–down approach. The other alternative would have been to use a bottom–up classification driven by the papers themselves as they were examined. Our fear was that a bottom–up approach would lead to a haphazard classification, especially given our previous negative experiences with computing classification schemes [11].

Considerable time was spent early in our research process defining those categories, and the resulting classification schemes. That work is described in depth in Section 3 of this paper. In addition to the primary goal of this paper of describing the nature of software engineering research, the provision of these taxonomies is a valuable contribution of this research.

### 2.2.4. Examining the papers

Once the appropriate decisions had been made regarding the duration of the study, the journals to be examined, and the classification scheme to be used, we began examining the papers themselves. Sometimes a paper could be classified on the basis of its title, abstract, introduction, and key words. More often, the full paper had to be examined. The bulk of the research time of this study was spent in this examination.

Table 1  
Computing topics

|     |  |      |   |
|-----|--|------|---|
| 1.0 | Problem-solving concepts   | 5.3  | Systems programming   |
| 1.1 | Algorithms   | 5.4  | Real-time (incl. robotics)  |
| 1.2 | Mathematics/computational science  | 5.5  | Edutainment (incl. graphics)  |
| 1.3 | Methodologies (object, function/process, information/data, event, business rules, etc.)            | 6.0  | System/software management concepts                                     |
| 1.4 | Artificial intelligence  | 6.1  | Project/product management (incl. risk management)                      |
| 2.0 | Computer concepts  | 6.2  | Process management  |
| 2.1 | Computer/hardware principles/architecture  | 6.3  | Measurement/metrics (development and use)                               |
| 2.2 | Inter-computer communication (networks, distributed systems)                                       | 6.4  | Personnel issues  |
| 2.3 | Operating systems (as an augmentation of hardware)   | 7.0  | Organizational concepts   |
| 2.4 | Machine/assembler-level data/instructions  | 7.1  | Organizational structure  |
| 3.0 | Systems/software concepts  | 7.2  | Strategy  |
| 3.1 | System architecture/engineering  | 7.3  | Alignment (incl. business process reengineering)                        |
| 3.2 | Software life-cycle/engineering (incl. requirements, design, coding, testing, maintenance)         | 7.4  | Organizational learning/knowledge management                            |
| 3.3 | Programming languages  | 7.5  | Technology transfer (incl. innovation, acceptance, adoption, diffusion) |
| 3.4 | Methods/techniques (incl. reuse, patterns, parallel processing, process models, data models, etc.) | 7.6  | Change management   |
| 3.5 | Tools (incl. compilers, debuggers)   | 7.7  | Information technology implementation                                   |
| 3.6 | Product quality (incl. performance, fault tolerance)   | 7.8  | Information technology usage/operation                                  |
| 3.7 | Human-computer interaction   | 7.9  | Management of 'computing' function                                      |
| 3.8 | System security  | 7.10 | IT impact   |
| 4.0 | Data/information concepts  | 7.11 | Computing/information as a business                                     |
| 4.1 | Data/file structures   | 7.12 | Legal/ethical/cultural/political (organizational) implications          |
| 4.2 | Data base/warehouse/mart organization  | 8.0  | Societal concepts   |
| 4.3 | Information retrieval  | 8.1  | Cultural implications   |
| 4.4 | Data analysis  | 8.2  | Legal implications  |
| 4.5 | Data security  | 8.3  | Ethical implications  |
| 5.0 | Problem domain specific concepts   | 8.4  | Political implications  |
| 5.1 | Scientific/engineering (incl. bio-informatics)   | 9.0  | Disciplinary issues   |
| 5.2 | Information systems (incl. decision support, group support systems, expert systems)                | 9.1  | 'Computing' research  |
|     |  | 9.2  | 'Computing' curriculum/teaching   |

### 3. Classification

We introduced the topic of classification of the papers in question in Section 2. Here, we discuss that classification scheme in more detail and also discuss the process we used to code the papers.

#### 3.1. Classification scheme

The fundamental intent of our classification scheme was to cover the breadth of, not just the field of SE, but the computing field as a whole. This would allow us to identify, for example, not just what topics SE research addresses as a function of its internal interests, but what topics SE research addresses out of the total spectrum of computing topics.

We chose our five characteristics based on the research questions described earlier: the topic of each paper, its research approach and method, the reference disciplines it utilized, and the level at which it addressed the issues. We defined a classification scheme for each of these research characteristics, keeping in mind that the scheme should cover the entire computing field, while permitting the classification of SE research papers.

An in-depth discussion of that classification scheme may be found in Ref. [32]. Here, we include sufficient discussion of its content to make this paper self-contained. The classifications themselves are presented in this paper as Tables 1–5.

To promote consistent classification and analysis it was important that for each paper examined we select one dominant category for most of the five classifications. For example (and with certain exceptions), it was necessary to choose a single reference discipline, or a single topic, or a single research approach/method. Most SE papers touch on multiple matters within a single category, and it sometimes became an exercise in compromise to select the dominant category. Two independent researchers (hereafter referred to as 'coders') categorized each of the papers, compared results, and adjudicated any differences.

As mentioned above, for certain categories it was permissible for the coders to make multiple choices. For example, for the topic Problem-domain-specific concepts, coders were encouraged to choose another dominant topic as well as the domain-dependent topic. And for research approach and method, the coding sheet used provided for multiple instances of each of those categories.

Table 2  
Research approach

|              |  |
|--------------|--|
| Descriptive: |  |
| DS           | Descriptive system                     |
| DO           | Descriptive other                      |
| DR           | Review of literature                   |
| Evaluative:  |  |
| ED           | Evaluative–deductive                   |
| EI           | Evaluative–interpretive                |
| EC           | Evaluative–critical                    |
| EO           | Evaluative–other                       |
| Formulative: |  |
| FF           | Formulative–framework                  |
| FG           | Formulative–guidelines/standards       |
| FM           | Formulative–model                      |
| FP           | Formulative—process, method, algorithm |
| FT           | Formulative–classification/taxonomy    |
| FC           | Formulative–concept                    |

### 3.1.1. Classifying topic

Topic is arguably the key issue in understanding SE research. SE as a field addresses itself to the solutions in any possible application domain, and—as the history of the computing field progresses—the diversity of those domains becomes far-reaching.

Historically, however, SE research has focused more on generic solutions than it has on specific solutions in specific problem domains [11]. Because of that, the topic classification scheme is necessarily focused more on application-independent approaches than it is on problem-specific approaches.

In the interests of comprehensiveness, we relied on previous topic classification schemes. To ensure that our

Table 3  
Research methods

|     |   |
|-----|---|
| AR  | Action research                           |
| CA  | Conceptual analysis                       |
| CAM | Conceptual analysis/mathematical          |
| CI  | Concept implementation (proof of concept) |
| CS  | Case study                                |
| DA  | Data analysis                             |
| DI  | Discourse analysis                        |
| ET  | Ethnography                               |
| FE  | Field experiment                          |
| FS  | Field study                               |
| GT  | Grounded theory                           |
| HE  | Hermeneutics                              |
| ID  | Instrument development                    |
| LH  | Laboratory experiment (human subjects)    |
| LR  | Literature review/analysis                |
| LS  | Laboratory experiment (software)          |
| MA  | Meta-analysis                             |
| MP  | Mathematical proof                        |
| PA  | Protocol analysis                         |
| PH  | Phenomenology                             |
| SI  | Simulation                                |
| SU  | Descriptive/exploratory survey            |

Table 4  
Reference disciplines

|    |                               |
|----|-------------------------------|
| CP | Cognitive psychology          |
| SB | Social and behavioral science |
| CS | Computer science              |
| SC | Science                       |
| EN | Engineering                   |
| EC | Economics                     |
| LS | Library science               |
| MG | Management                    |
| MS | Management science            |
| PA | Public administration         |
| PS | Political science             |
| OT | Other                         |

list of topics was sufficiently broad to include all areas of computing research, we used several sources of topics from the general discipline of computing, viz., the ACM Computing Reviews Classification scheme (www.acm.org), the categories in Ref. [3] (known as the ISRL categories), and the topic areas identified by Glass [14]. In particular, we used the classification scheme proposed by Glass [14] as the starting point for arriving at the high-level categories shown in Table 1 because its stated objective of ‘presenting a comprehensive set of topics taught in the fields of Computer Science, Software Engineering, and Information Systems’ best fit our completeness criterion.

The overall classification scheme, as shown in Table 1, divides the topics of the computing research field into several major categories:

|                                  |                                      |
|----------------------------------|--------------------------------------|
| Problem-solving concepts         | Systems/software management concepts |
| Computer concepts                | Organizational concepts              |
| Systems/software concepts        | Societal concepts                    |
| Data/information concepts        | Disciplinary issues                  |
| Problem-domain-specific concepts |                                      |

Each of those major categories, as you might expect, is divided into subordinate categories. There are eight such subordinates under Systems/software, five under Problem-domain-specific, and six under Systems/software management. The findings of Section 4.1 will be represented at the

Table 5  
Levels/units of analysis

|     |   |
|-----|---|
| SOC | Society   |
| PRO | Profession                                      |
| EXT | External business context                       |
| OC  | Organizational context                          |
| PR  | Project   |
| GP  | Group/team                                      |
| IN  | Individual                                      |
| CS  | System  |
| CE  | Computing element—program, component, algorithm |
| AC  | Abstract concept                                |

subordinate category level, as well as being aggregated at the major category level.

### 3.1.2. *Classifying research approach*

In addition to classifying topic, we also categorized the research techniques used by the papers. We divided those techniques into Research Approach, the overall approach undertaken in performing the research, and Research Method, the more detailed techniques used. In this section, we discuss Research Approach.

Surprisingly, there is little unanimity in the field regarding classifying research techniques. Several approaches have been proposed, including Adrion [2], and Glass [12], which tend to define the Research Approaches into such categories as Informational, Propositional, Analytical, and Evaluative. In the end, however, we used Morrison and George's [22] categorization of research approaches as a starting point for determining the research approaches to be examined in this study. Based on an analysis of articles in both Software Engineering and Information Systems between 1986–1991, they characterized the four major research approaches as Formulative, Evaluative, Descriptive, and Developmental. With the exception of Developmental, which we included in the Descriptive category, we subdivided these categories to reflect a rich set of research approaches. Note, also, that our formulative and descriptive categories further characterize the non-empirical category used in papers by Farhoomand and Drury [6], Hamilton and Ives [16,17], and Alavi and Carlson [1].

Table 2 shows the categories used to classify research approach in this study. The descriptive approach has three subcategories. Subcategory DS is based on Morrison and George's Descriptive category and is used to capture papers whose primary focus is describing a system. Descriptive–Other (DO) was added to capture those papers that used a descriptive approach for describing something other than a system, for example, an opinion piece. We added DR as a subcategory into which we categorized papers whose primary content was a review of the literature.

The formulative research approach was subcategorized into a rich set of possible entities being formulated, including Processes/procedures/methods/algorithms (FP), and Frameworks and Guidelines/standards (FF and FG, respectively). In all, there are six subcategories of formulation.

Our evaluative categories are based on the three major 'evaluative' epistemologies identified by Orlikowski and Baroudi [23]: Positivist (Evaluative–Deductive in our scheme), Interpretive (Evaluative–Interpretive), and Critical (Evaluative–Critical). We added an 'Other' category here to characterize those papers that have an evaluative component but that did not use any of the three approaches identified above. For example, we classified papers that used opinion surveys to gather data (as opposed to construct-based questionnaires) under Evaluative–Other.

Because authors rarely indicated the research approach they employed explicitly in the abstract, keyword, or even in

the introduction, we usually categorized the primary research approach used by examining relevant sections of the article.

### 3.1.3. *Classifying research method*

As discussed in Section 3.1.2, research techniques were divided into research approach and research method for the purpose of this study. Here we present our classification for research method, the more detailed level of research technique.

Our problem for research method was almost the opposite of the problem noted above, where there were few candidate research approach classifications from which to choose. On the contrary, here—for the topic of research method—there was a plethora.

Recall that the categories and taxonomies used in this paper are intended to cover the whole of the computing field, not just SE. In the IS field alone, there were many prior papers categorizing IS research that had identified a number of commonly-used methods (see, for example, Refs. [1,6]). All of these articles identify, for example, Laboratory Experiments (using human subjects), Field Studies, Case Studies, and Field Experiments. Several other research methods have also been identified, including Action Research and Ethnography [18], Conceptual Analysis (or Conceptual Study), Literature review [20], and Instrument development [1]. More specific to the field of SE, there were also studies of research methods. A fairly simplistic approach, derived from Ref. [2], was suggested in [12]. It involved categorizing methods into scientific, engineering, empirical, and analytical. Zelkowitz and Wallace [35] also proposed similar SE categories, specific to methods for experimental or empirical studies.

Table 5 shows therefore the plethora of research method categories used in this research. They are not easily grouped into major categories and subcategories (as, for example, were the research approaches). Research method was thus coded at the lowest (and only!) level. Not all of the research methods included will be appropriate to any particular branch of computing. Many of them were invented for one or more of the other computing subfields and are unlikely to apply to SE. To assist in the SE categorization, we added: Conceptual Analysis/Mathematical and Mathematical Proof to facilitate the classification of papers that utilize mathematical techniques; Simulation, to allow categorization of papers that utilized simulation in their research methods; and Concept Implementation (for papers whose prime research method was to demonstrate proof of a concept). We also added the category Laboratory Experiment (Software) to characterize those papers that, for example, compare the performance of a newly-proposed system with other (existing) systems. Finally, we used the category Exploratory Survey to classify papers that conducted an 'exploratory field study in which there is no test of relationships between variables' Cheon et al. [4].

### 3.1.4. Classifying reference discipline

By reference discipline, we mean any discipline outside the SE field that the SE researchers have relied upon for theory and/or concepts. Generally, a reference discipline is one that provides an important basis for the research work being conducted. For example, there is a branch of software research known as Empirical Studies of Programmers, which relies heavily on cognitive psychology for the concepts that it explores in the field of software development. Another example would be such software research fields as statistical process control, and even process maturity models, which at least at the outset relied heavily on manufacturing for their concepts.

Reference discipline as a concept is frequently mentioned in the Information Systems field of computing. It was there that we went for help in defining the classification scheme of reference disciplines presented in Table 4. Various IS researchers have characterized the reference disciplines used in IS research (see, for example, Refs. [5,28,33]). Culnan and Swanson [5] examined the reliance of IS publications on publications in the fields of Computer Science, Organization Science, and Management Science. Swanson and Ramiller [28] identified Computer Science, Management Science and Cognitive Science, Organizational Science, and Economics as the four key reference disciplines. Barki et al. [3] also include Behavioral Science, Organizational Theory, Management Theory, Language Theories, Artificial Intelligence, Ergonomics, Political Science, and Psychology, while Westin et al. [33] further identified Mathematics/Statistics and Engineering.

Table 4 presents our reference discipline categories, which represent a comprehensive aggregation of the categories addressed in prior research, i.e., some of our categories subsumed one or more of the categories outlined above. The Management category, for example, subsumes Organizational Theory and Management Theory. Similarly, Artificial Intelligence and Software Engineering is subsumed within Computer Science. Finally, the category Social and Behavioral Science subsumes the Communication (e.g. Media Richness Theory) and Social Psychology (e.g. Theory of Reasoned Action) literature.

### 3.1.5. Classifying level of analysis

Another concept borrowed primarily from the field of Information Systems is that of level of analysis. Here, we refer to the notion that research work is often couched in some more social setting. For example, the Watts Humphrey work on Team Software Process is about teams (which we here called GP, for Group/Team), whereas his Personal Software Process work is about the individual (IN). Some research work is done at the level of the Profession (PRO), of which this paper is an example, while others may be conducted within an enterprise at the Organizational (OC) level.

For the benefit of SE and other, more technical (as opposed to behavioral), computing research work, we

added the (non-social) levels of analysis: Computing System (CS) and Computing Element (CE, representing a program, component, algorithm, or object) and Abstract Concept (AC).

The levels of analysis used in this paper are presented in Table 5.

### 3.2. The classification process

Two of the three authors of this paper independently classified (coded) each of the SE articles. Following the individual codings, the first author of this paper resolved all differences, choosing a final coding that was usually—but not always—one of the two original codings.

Agreement between the individual authors varied. For example, ‘levels of analysis’ coding was at high levels of agreement, whereas ‘topic’ coding was more problematic. Disagreement occurred most often when a paper could legitimately have been coded in more than one way. For example, was a particular paper on the subject of design techniques more about ‘methods/techniques’ or ‘software life cycle—design’? Original agreements occurred at roughly the 60% level, which was considered acceptable given the sometimes-subjective nature of the final resolution.

## 4. Findings

Tables 6–10 present raw data for the number of articles examined in each of the six journals in the five-year period from 1995 to 1999. The findings that follow represent the results of answering our research questions regarding SE research—its topics, research approaches and methods, reference disciplines, and levels of analysis. They represent both some results that are predictable, and some that are fairly surprising.

### 4.1. Findings for topic

Because of our perception that this may be the most meaningful SE category to SE researchers, we first look at the findings for SE research topic.

Table 6 shows that SE publications cover all of the nine major topic categories (Problem-solving concepts, Computer concepts, Systems/software concepts, Data/information concepts, Problem-domain specific concepts, Systems/software management concepts, Organizational concepts, Societal concepts, and Disciplinary issues), although they are not, of course, equally distributed. In fact, the dominant category is that of Systems/software concepts. Two hundred and two papers (54.8% of the total number) were assigned to that major category.

The second major category was Systems/software management concepts. Here, there were 42 papers (11.6%). Note that this is a significant drop from the number in the dominant category. Clearly, the interests of the SE



research field are heavily focused on the technology of systems and software engineering, with a considerably diminished interest in the other categories, such as management, for example.

Regarding subcategories, as would be expected, most of the top topics are found within the Systems/software concepts major category. The leading subcategory, with 67 papers (18.2%), is Methods/techniques. This category was defined to include such a diverse collection of topics as reuse, parallel processing, process models, data models, and formal methods.

Other important subcategories include:

|     |   |            |
|-----|---|------------|
| 3.5 | Tools (including compilers and debuggers)   | 45 (12.1%) |
| 2.2 | Inter-computer communications (from the computer concepts major category) (including networks and distributed systems)    | 35 (9.5%)  |
| 3.2 | Software life cycle engineering (including each of the life cycle phases (e.g. design, maintenance))                      | 32 (8.7%)  |
| 3.6 | Product quality (including performance and fault tolerance)   | 31 (8.4%)  |
| 6.3 | Measurement and metrics (from the systems/software management concepts major category)                                    | 23 (6.2%)  |
| 1.3 | Methodologies (from the problem-solving major category)(including object, function/process, information/data, and others) | 18 (4.9%)  |
| 4.2 | Data base/warehouse/mart organization (from the data/information concepts major category)                                 | 17 (4.6%)  |
| 3.3 | Programming languages   | 14 (3.8%)  |
| 6.1 | Project/product management (from the systems/software management concepts major category) (including risk management)     | 12 (3.3%)  |

Perhaps the biggest surprise in the topic coding is the lack of entries in the Problem-domain-specific-concept major category. Only 10 papers (2.7%) fell into that category. And this small number is in spite of the fact that, during coding, papers could be classified into both another topic area and a domain-specific topic area. Software engineering authors in recent years (see, for example, Refs. [19,24]) have heavily promoted the need for domain-specific research and software construction approaches. This data seems to reflect the fact that scant research attention is being paid to those cries.

Other major categories to which a significant number of papers were assigned were Data/information concepts (28, 7.6%), problem-solving concepts (22, 5.9%), and Disciplinary issues (this major category includes research about research, and research into pedagogy), with 13 (3.5%).

It is interesting to note that, in spite of the dearth of entries in the Problem domain category, Table 6 demonstrates that the spread of SE research topics was fairly broad. Not only were all the major categories covered, but there were significant numbers of papers in several of the subcategories outside the top two or three major categories where most of the SE research was focused.

#### 4.2. Findings for research approach

Table 7 shows the research approaches used by SE researchers. Note that, in the overall categories Describe, Formulate, and Evaluate, the dominant research approach was Formulate, with 55.3% of the papers so categorized. Description trailed at 27.9%, and Evaluation came in last at 13.8%.

When we examine the subcategories of research approach, we come to an even more dominant finding. There is one subcategory of the Formulate category into which the majority of the Formulate papers fall. The predominant approach used by SE researchers is FP, a multifaceted subcategory that includes formulating processes, procedures, methods, or algorithms. Fully 36% of the total research approaches fell into this category. Note that there was no lack of other ‘formulate’ subcategories: Table 7 shows us that there were five other such choices. In fact, the number two subcategory was DO, a Descriptive subcategory used for describing something other than a system; 18.2% of the SE paper research approaches fell here. Number three was FM, formulation of a model, which captured nearly 10% of the papers. Next was DS, description of a system, which came in at roughly 8%. Interestingly, the next subcategory was EO, evaluate using ‘other’ approaches, with 7.3%.

Thus, although ‘formulate’ was the dominant category and ‘formulate process/procedure/ method/ algorithm’ was the dominant subcategory, following that the assignment of SE papers to research approaches was rather diverse. The bottom line seems to be that SE researchers like to formulate particular kinds of things; they are considerably less interested in describing or evaluating.

#### 4.3. Findings for research method

Table 8 shows the research methods used by SE researchers. Note that three categories dominate—Conceptual analysis (with 43.3% of the papers); a related method, Conceptual analysis /mathematical (where the analysis is heavily mathematical) (with 10.6%, ranking third); and Concept implementation (with 17.1%, placing it second only to Conceptual analysis).

No other research method reached double digits; none of the other methods was, therefore, very significant to SE research. There were very few papers that solely focused on using mathematical proofs, for example. Note, however, that some of those papers may have been included in the Conceptual analysis / mathematical category. There were very few simulations despite the fact that this category was added specifically for SE research! There were very few case or field studies. In short, the field of SE research is surprisingly focused in its choice of research methods.

As we have seen, because research papers are complex, and may employ several research approaches within the

Table 6  
Findings for computing topic

|     |   |       |      |   |       |
|-----|---|-------|------|---|-------|
| 1.0 | Problem-solving concepts  | 5.9%  | 5.2  | Information systems (incl. decision support, group support systems, expert systems) | 1.6%  |
| 1.1 | Algorithms  | < 1%  | 5.3  | Systems programming   | 0%    |
| 1.2 | Mathematics/computational science   | 0%    | 5.4  | Real-time (incl. robotics)  | < 1%  |
| 1.3 | Methodologies (object, function/process, information/data, event, business rules, ...)  | 4.9%  | 5.5  | Edutainment (incl. graphics)  | < 1%  |
| 1.4 | Artificial intelligence   | < 1%  | 6.0  | Systems/software management concepts  | 11.6% |
| 2.0 | Computer concepts   | 10.9% | 6.1  | Project/product management (incl. risk management)                                  | 3.3%  |
| 2.1 | Computer/hardware principles/architecture   | 0%    | 6.2  | Process management  | 2.2%  |
| 2.2 | Inter-computer communication (networks, distributed systems)  | 9.5%  | 6.3  | Measurement/metrics (development and use)   | 6.2%  |
| 2.3 | Operating systems (as an augmentation of hardware)  | 1.4%  | 6.4  | Personnel issues  | < 1%  |
| 2.4 | Machine/assembler-level data/instructions   | 0%    | 7.0  | Organizational concepts   | 1.9%  |
| 3.0 | Systems/software concepts   | 54.8% | 7.1  | Organizational structure  | < 1%  |
| 3.1 | System architecture/engineering   | 1.9%  | 7.2  | Strategy  | 0%    |
| 3.2 | Software life-cycle/engineering (incl. requirements, design, coding, testing, maintenance)  | 8.7%  | 7.3  | Alignment (incl. business process reengineering)                                    | < 1%  |
| 3.3 | Programming languages   | 3.8%  | 7.4  | Organizational learning/knowledge management  | 0%    |
| 3.4 | Methods/techniques (incl. reuse, patterns, parallel processing, process models, data models, etc)                                 | 18.2% | 7.5  | Technology transfer (incl. innovation, acceptance, adoption, diffusion)             | < 1%  |
| 3.5 | Tools (incl. compilers, debuggers)  | 12.2% | 7.6  | Change management   | 0%    |
| 3.6 | Product quality (incl. performance, fault tolerance)  | 8.4%  | 7.7  | Information technology implementation   | 0%    |
| 3.7 | Human-computer interaction  | 1.1%  | 7.8  | Information technology usage/operation  | 0%    |
| 3.8 | System security   | < 1%  | 7.90 | Management of 'computing' function  | 0%    |
| 4.0 | Data/information concepts   | 7.6%  | 7.11 | IT Impact   | < 1%  |
| 4.1 | Data/file structures  | < 1%  | 7.11 | Computing/information as a business   | 0%    |
| 4.2 | Data base/warehouse/mart organization   | 4.6%  | 7.12 | Legal/ethical/cultural/political (organizational) implications                      | < 1%  |
| 4.3 | Information retrieval   | 1.4%  | 8.0  | Societal concepts   | < 1%  |
| 4.4 | Data analysis   | < 1%  | 8.1  | Cultural implications   | 0%    |
| 4.5 | Data security   | < 1%  | 8.2  | Legal implications  | 0%    |
| 5.0 | Problem-domain-specific concepts (use as a secondary subject, if applicable, or as a primary subject if there is no other choice) | 2.7%  | 8.3  | Ethical implications  | 0%    |
| 5.1 | Scientific/engineering (incl. bio-informatics)  | < 1%  | 8.4  | Political implications  | < 1%  |
|     |   |       | 9.0  | Disciplinary issues   | 3.5%  |
|     |   |       | 9.1  | 'Computing' research  | 1.1%  |
|     |   |       | 9.2  | 'Computing' curriculum/teaching   | 2.4%  |

same piece of work, the coders could choose to assign more than one research approach and more than one research method. This approach would have tended to diversify the research approach and research method results. As it turned out, this multiple classification possibility was seldom utilized, either because those doing the classification saw only one approach/method, or because one approach/method seemed to them to be sufficiently dominant.

Thus it is possible to say that, of all the categories discussed in this paper to date, research method is the least diverse of the SE research categories. SE researchers tend to analyze and implement new concepts, and they do very little of anything else.

#### 4.4. Findings for reference discipline

Table 9 shows the reference disciplines used by the SE researchers. For the most part SE research eschews reliance on other fields for its fundamental theories and/or concepts. Ninety-eight percent of the papers examined had no reference discipline. There were trivial instances of papers

that relied on Cognitive Psychology (at the .54% level!), Social and Behavioral Science, Management, and Management Science, each of the latter at the .27% level.

It is interesting to note that there was no reliance on such fields as Mathematics, Engineering, or any of the Sciences. Without passing any judgment on this finding, it is clear that SE research tends to be quite self-contained, not relying on any other disciplines for its thinking.

#### 4.5. Findings for level of analysis

Table 10 shows the levels of analysis used by SE researchers. Here, there was a scattering of use of these categories to characterize SE research. The most dominant level of analysis, however, was the Abstract Concept category, which was used to describe a concept such as a data model. Approximately 50% of the SE research papers were conducted at this level.

The second most common level of analysis category was CE, or Computing element, with 27.9%. CS, Computing system, accounted for another 10.6%.

There was very small usage of the more 'social' levels of

Table 7  
Findings for research approach

|              |   |       |
|--------------|---|-------|
| Descriptive: |   | 27.9% |
| DS           | Descriptive system                        | 8.1%  |
| DO           | Descriptive other                         | 18.2% |
| DR           | Review of literature                      | 1.6%  |
| Evaluative:  |   | 13.8% |
| ED           | Evaluative–deductive                      | 4.3%  |
| EI           | Evaluative–interpretive                   | < 1%  |
| EC           | Evaluative–critical                       | 1.4%  |
| EO           | Evaluative–other                          | 7.3%  |
| Formulative: |   | 55.3% |
| FF           | Formulative-framework                     | 4.1%  |
| FG           | Formulative-guidelines/<br>standards      | 4.3%  |
| FM           | Formulative-model                         | 9.8%  |
| FP           | Formulative—process,<br>method, algorithm | 36.0% |
| FT           | Formulative-<br>classification/taxonomy   | 1.1%  |
| FC           | Formulative-concept                       | 3.0%  |

analysis categories—4.1% were project-level studies, 2.4% were professional-level studies, 2.2% of studies were conducted in an organizational context, and there were even lesser numbers for group, individual, and societal levels.

It is clear from this data that SE research is fundamentally about technical, computing-focused issues, and that it is seldom about behavioral issues.

Table 8  
Findings for research methods

|     |  |       |
|-----|--|-------|
| AR  | Action research                              | 0%    |
| CA  | Conceptual analysis                          | 43.5% |
| CAM | Conceptual analysis/<br>mathematical         | 10.6% |
| CI  | Concept implementation (proof<br>of concept) | 17.1% |
| CS  | Case study                                   | 2.2%  |
| DA  | Data analysis                                | 2.2%  |
| DI  | Discourse analysis                           | 0%    |
| ET  | Ethnography                                  | 0%    |
| FE  | Field experiment                             | < 1%  |
| FS  | Field study                                  | < 1%  |
| GT  | Grounded theory                              | < 1%  |
| HE  | Hermeneutics                                 | < 1%  |
| ID  | Instrument development                       | 0%    |
| LH  | Laboratory experiment (human<br>subjects)    | 3.0%  |
| LR  | Literature review/analysis                   | 1.1%  |
| LS  | Laboratory experiment<br>(software)          | < 1%  |
| MA  | Meta-analysis                                | 0%    |
| MP  | Mathematical proof                           | < 1%  |
| PA  | Protocol analysis                            | 0%    |
| PH  | Phenomenology                                | 0%    |
| SI  | Simulation                                   | 1.1%  |
| SU  | Descriptive/exploratory survey               | 1.6%  |

Table 9  
Findings for reference disciplines

|    |                               |       |
|----|-------------------------------|-------|
| CP | Cognitive psychology          | < 1%  |
| SB | Social and behavioral science | < 1%  |
| CS | Computer science              | 0%    |
| SC | Science                       | 0%    |
| EN | Engineering                   | 0%    |
| EC | Economics                     | 0%    |
| LS | Library science               | 0%    |
| MG | Management                    | < 1%  |
| MS | Management science            | < 1%  |
| PA | Public administration         | 0%    |
| PS | Political science             | 0%    |
| OT | Other                         | 98.1% |

#### 4.6. Findings by journal

This research was designed to characterize the SE research discipline, not the journals of the SE field. However, some interesting conclusions can be drawn from the findings regarding journal differences. Table 11 presents raw data for the number of articles examined in each of the six journals, while Tables 12–16 present the data by journal for each of the categories examined.

1. The dominant content of TOSEM is SE methods/techniques. Whereas the other journals devoted roughly 14–22% of their papers to this topic, TOSEM devoted 66.7%.
2. TSE is perhaps the most diverse journal with respect to topics, spreading its content around the topics of inter-computer communication (15.9%), software life-cycle engineering (17.5%), methods/techniques (14.3%), product quality (15.9%), and measurement/metrics (11.1%).
3. SPE is primarily about systems rather than concepts; the leading research approach category for SPE was description of a system (at 32.9%), compared with 2–8% for the other journals.
4. Correspondingly, SPE publishes more research involving concept implementation than the other journals. Sixty percent of its papers used this research method, while for the other journals, the figure was 8–18%. Conceptual analysis was the leading research method in the other journals.

Table 10  
Findings for levels of analysis

|     |  |       |
|-----|--|-------|
| SOC | Society  | < 1%  |
| PRO | Profession   | 2.4%  |
| EXT | External business context                          | 0%    |
| OC  | Organizational context                             | 2.2%  |
| PR  | Project  | 4.1%  |
| GP  | Group/team   | 1.4%  |
| IN  | Individual   | 1.4%  |
| CS  | System   | 10.6% |
| CE  | Computing element–program,<br>component, algorithm | 27.9% |
| AC  | Abstract concept                                   | 49.9% |

Table 11  
Numbers of articles examined by journal and year

| Journal | JSS | SPE | TOSEM | TSE | IST | SW | Totals |
|---------|-----|-----|-------|-----|-----|----|--------|
| 1995    | 21  | 14  | 2     | 15  | 13  | 8  | 73     |
| 1996    | 19  | 14  | 3     | 12  | 13  | 10 | 71     |
| 1997    | 20  | 14  | 2     | 10  | 17  | 10 | 73     |
| 1998    | 24  | 15  | 3     | 15  | 14  | 8  | 79     |
| 1999    | 21  | 13  | 2     | 11  | 17  | 9  | 73     |
| Totals  | 105 | 70  | 12    | 63  | 74  | 45 | 369    |

5. Nearly all of the papers in TOSEM were at the computing element level of analysis (91.7%). For the other journals, this category ranged from 2 to 40%. SPE focused on computing element in 40% of its papers; computing element, and computing system at 30%, dominated its levels of analysis. For the other journals, the dominant level of analysis was ‘abstract concept.’

## 5. Discussion and implications

Recall that our intention in performing this research was to describe the current state of SE research, but not to pass judgment on that state (that is, the results were to be descriptive rather than normative). To achieve our objective, we examined the following characteristics of SE research: topic, research approach and method, reference discipline, and level of analysis. We believe this study represents the first attempt to take an objective, unbiased look at the totality of research in the field.

In the course of this research, we examined and analyzed 369 research papers in six leading SE journals over the period 1995–1999, inclusive. Out of that analysis came these findings:

Regarding topic, SE research is fairly diverse, but there is a heavy concentration of papers in the major category ‘Systems/software concepts,’ especially in its subcategory ‘methods/techniques.’ The second leading subcategory was ‘tools’ (also under ‘Systems/software concepts’), but, interestingly, there were also high concentrations in some other major categories, including ‘Computer concepts,’ ‘Systems/software management,’ and ‘Data/information concepts.’

Table 12  
Representation by topic

|              | Overall (%) | JSS (%) | SPE (%) | TOSEM (%) | TSE (%) | IST (%) | SW (%) |
|--------------|-------------|---------|---------|-----------|---------|---------|--------|
| Prob-solving | 5.96        | 8.57    | 4.29    | –         | 1.59    | 8.11    | 6.67   |
| Computer     | 10.84       | 13.33   | 14.29   | –         | 15.87   | 4.05    | 6.67   |
| Systs/SW     | 54.74       | 44.76   | 71.43   | 91.67     | 65.08   | 47.30   | 40.00  |
| Data/Info    | 7.59        | 9.52    | 5.71    | –         | 1.59    | 14.86   | 4.44   |
| Prob-domain  | 2.71        | 2.86    | 2.86    | –         | 1.59    | 5.41    | –      |
| Sys/SW Mgt   | 12.47       | 11.43   | –       | 8.33      | 14.29   | 12.16   | 33.33  |
| Org’al       | 1.90        | 3.81    | 1.43    | –         | –       | 1.35    | 2.22   |
| Societal     | 0.27        | 0.95    | –       | –         | –       | –       | –      |
| Disc Issues  | 3.52        | 4.76    | –       | –         | –       | 6.76    | 6.67   |

Regarding research approach, SE research is quite narrow. The majority of papers fell into the research approach ‘Formulate,’ and by far the leading subcategory was ‘Formulate procedure/process/method/algorithm,’ despite the fact that there were other formulate categories, such as formulate model or formulate guidelines. Papers that described things, such as systems or the results of literature searches, were represented to a much lesser extent, and papers that performed evaluation as their dominant research approach fell well behind that.

Regarding research method, SE research was only a little more diverse. The dominant categories had to do with conceptual analysis and concept implementation. Perhaps surprisingly, there were very few instances of case/field studies, simulations, or mathematical proofs, although some of those might have fallen into the category ‘Conceptual analysis/mathematical.’

Clearly, from the first three SE research characteristics above, SE research is focused heavily on systems/software technical matters—how to build systems, how to formulate better ways to build systems, and how to analyze or implement promising new concepts.

Regarding reference disciplines, SE research seldom relies on other disciplines as the basis for its work. Although there have been discussions, over the years, of the relationship between SE research and such fields as cognitive psychology, quality, engineering, and manufacturing, at this point in time there is little evidence that SE seeks to assimilate learnings from other fields.

Regarding levels of analysis, once again we see that SE research is predominantly technical. There was very little use of the ‘social’ level of analysis categories. Instead, SE research fell into the more technical ‘computing element’ or ‘computing system’ categories.

Overall, it would seem reasonable to conclude that SE research, circa 2001, is focused inward, emphasizing technical matters above behavioral matters, but willing and able to address itself to a fairly wide variety of topics.

It is interesting, as an illustration of how this research was conducted, to classify this paper itself according to the five classification schemes described in the

Table 13  
Representation by research approach

|    | Overall (%) | JSS (%) | SPE (%) | TOSEM (%) | TSE (%) | IST (%) | SW (%) |
|----|-------------|---------|---------|-----------|---------|---------|--------|
| DO | 18.16       | 16.19   | 25.71   | 8.33      | 9.52    | 8.11    | 42.22  |
| DR | 1.63        | 0.95    | –       | –         | –       | –       | 11.11  |
| DS | 8.13        | 1.90    | 32.86   | –         | 7.94    | –       | –      |
| ED | 4.34        | 5.71    | –       | 8.33      | 6.35    | 6.76    | –      |
| EI | 0.81        | 0.95    | 1.43    | –         | –       | 1.35    | –      |
| EO | 7.32        | 3.81    | 2.86    | –         | 11.11   | 10.81   | 13.33  |
| EC | 1.36        | –       | –       | –         | 3.17    | 1.35    | 4.44   |
| FC | 2.98        | 9.52    | 1.43    | –         | –       | –       | –      |
| FF | 4.07        | 5.71    | 2.86    | –         | 7.94    | 2.70    | –      |
| FG | 4.34        | 2.86    | –       | –         | 1.59    | 4.05    | 20.00  |
| FM | 9.76        | 14.29   | 7.14    | 8.33      | 6.35    | 14.86   | –      |
| FP | 36.04       | 35.24   | 25.71   | 66.67     | 46.03   | 50.00   | 8.89   |
| FT | 1.08        | 2.86    | –       | 8.33      | –       | –       | –      |

preceding sections. Regarding topic, the classification is easy. This is research about SE research, and therefore this paper is a 9.1, focused on Computing research. Regarding research approach, the categorization is more difficult. It could be categorized as DR (review of the literature), or EO (evaluative–other). Because the paper is predominantly about evaluating software engineering research, we chose EO. Its research method is LR, ‘literature review/analysis.’ As to reference discipline, this paper is most readily categorized into the ‘None’ or ‘Not applicable’ category, as were so many of the SE research papers. And level of analysis? This paper is focused at the professional level (PRO). This analysis, incidentally, resulted in the keywords presented at the beginning of this paper. We encourage researchers in the future to use our classification scheme to write their abstracts and select their keywords. Such

a practice would aid other researchers immeasurably in assessing the relevance of published research to their own endeavors.

### 5.1. Limitations

Limitations of this research fall into these categories:

1. The accuracy of the classification schemes used.
2. The complexities of categorizing papers into those schemes.
3. Any limitations resulting from the choice of journals.

Regarding the classification schemes, the biggest problem was, for the most part, the diversity of candidates to choose from, and their inadequacies. The authors of this paper had prior experience with most of those candidate

Table 14  
Representation by research methodology

|     | Overall (%) | JSS (%) | SPE (%) | TOSEM (%) | TSE (%) | IST (%) | SW (%) |
|-----|-------------|---------|---------|-----------|---------|---------|--------|
| AR  | –           | –       | –       | –         | –       | –       | –      |
| CA  | 45.53       | 63.81   | 27.14   | 41.67     | 52.38   | 59.46   | 46.67  |
| CAM | 10.57       | 10.48   | 4.29    | 33.33     | 14.29   | 16.22   | 4.44   |
| CI  | 17.07       | 8.57    | 60.00   | –         | 7.94    | 9.46    | 17.78  |
| CS  | 2.17        | 1.90    | –       | 8.33      | 6.35    | 1.35    | 11.11  |
| DA  | 2.17        | 2.86    | 1.43    | –         | 3.17    | 2.70    | 2.22   |
| ET  | –           | –       | –       | –         | –       | –       | –      |
| FE  | 0.27        | –       | –       | –         | –       | 1.35    | –      |
| FS  | 0.81        | 1.90    | 1.43    | –         | –       | –       | 8.89   |
| GT  | 0.27        | 0.95    | –       | –         | –       | –       | –      |
| HE  | 0.27        | –       | –       | –         | –       | 1.35    | –      |
| ID  | –           | –       | –       | –         | –       | –       | –      |
| LH  | 2.98        | 2.86    | 1.43    | 8.33      | 4.76    | 4.05    | –      |
| LR  | 1.08        | 1.90    | –       | 8.33      | 1.59    | –       | 4.44   |
| LS  | 0.81        | 0.95    | 1.43    | –         | 1.59    | –       | –      |
| MP  | –           | –       | –       | –         | 6.35    | –       | –      |
| PA  | –           | –       | –       | –         | –       | –       | 2.22   |
| SI  | 1.08        | 1.90    | 2.86    | –         | –       | –       | 2.22   |
| SU  | 1.63        | 1.90    | –       | –         | 1.59    | 4.05    | –      |

Table 15  
Representation by reference discipline

|     | Overall (%) | JSS (%) | SPE (%) | TOSEM (%) | TSE (%) | IST (%) | SW (%) |
|-----|-------------|---------|---------|-----------|---------|---------|--------|
| CP  | 0.54        | 1.90    | –       | –         | –       | –       | –      |
| SB  | 0.27        | 0.95    | –       | –         | –       | –       | –      |
| CS  | –           | –       | –       | –         | –       | –       | –      |
| EC  | –           | –       | –       | –         | –       | –       | –      |
| IS  | –           | –       | –       | –         | –       | –       | –      |
| MG  | 0.27        | –       | –       | –         | –       | –       | 2.22   |
| MS  | 0.27        | –       | 1.43    | –         | –       | –       | –      |
| OT  | 0.27        | –       | –       | –         | –       | –       | 2.22   |
| SC  | 0.27        | –       | –       | –         | –       | 1.35    | –      |
| N/A | 98.10       | 97.14   | 98.57   | 100.00    | 100.00  | 98.65   | 95.56  |

schemes, and frequently found them wanting. In defining our own classification scheme, we relied heavily on such prior schemes, but we made sure that our revisions solved the problems we had previously encountered. It is our hope, as mentioned earlier in this paper, that these are not only accurate classification schemes for the purposes of this research, but that they will be found useful by other researchers in further studies. The one problem we found in using our schemes was the lack of mutual exclusion. It was often possible to categorize a paper into more than one element of some of the schemes. We believe that this speaks more to the complexity of SE research, however, than the inadequacy of the schemes.

Regarding the complexity of categorizing papers, sometimes we found that two coders derived two perfectly reasonable but different categories for certain of the characteristics. What this means is that subjective differences in categorizing might have had minor effects in the ranking of those categories.

Regarding the choice of journals, we believe that the six journals chosen do indeed constitute an excellent representation of the SE research field, especially because those same journals have been used in previous, related research. However, the size and publication frequency of these journals means that the findings are biased toward those journals that publish more papers more often (see Table 11). For example, TOSEM is a quarterly journal, publishing perhaps a dozen papers per year, while JSS is published 15 times per

year at present, and thus publishes around 100 papers per year. We used the same sampling rate (one paper out of five) for each of the journals.

## 5.2. Implications

We want to be careful in discussing the implications of our research to avoid passing judgment on the findings. Our findings do, however, have a number of implications for SE research and SE in general.

### 5.2.1. Implications for SE research

For those categories for which SE research was quite diverse, such as topic, there are few implications to consider. Obviously, SE research is well spread among its candidate topics, a finding that strikes us in all ways as positive.

However, for the remaining categories, the narrowness of the choices made by SE researchers provides considerable food for thought. Is there benefit, for example, to broadening research approaches and methods? Would case and field studies, for example, provide richer and more valuable findings for SE research? Would increasing amounts of evaluation provide benefit, particularly in improving the rate of technology transfer in the field? Or is the current approach of focusing heavily on the technical aspects of the field, using conceptual analysis to understand and enhance them, providing the most benefit?

Table 16  
Representation by level of analysis

|     | Overall (%) | JSS (%) | SPE (%) | TOSEM (%) | TSE (%) | IST (%) | SW (%) |
|-----|-------------|---------|---------|-----------|---------|---------|--------|
| SOC | 0.27        | –       | 1.43    | –         | –       | –       | –      |
| PRO | 2.44        | 4.76    | –       | –         | –       | 1.35    | 6.67   |
| EXT | –           | –       | –       | –         | –       | –       | –      |
| OC  | 2.17        | 3.81    | –       | –         | 1.59    | –       | 6.67   |
| PR  | 4.07        | 4.76    | –       | –         | 3.17    | 1.35    | 15.56  |
| GP  | 1.36        | –       | 1.43    | 8.33      | –       | 4.05    | –      |
| IN  | 1.36        | 2.86    | 1.43    | –         | –       | –       | 2.22   |
| AC  | 49.86       | 40.00   | 25.71   | –         | 66.67   | 78.38   | 53.33  |
| CS  | 10.57       | 6.67    | 30.00   | –         | 4.76    | 2.70    | 13.33  |
| CE  | 27.91       | 37.14   | 40.00   | 91.67     | 23.81   | 12.16   | 2.22   |

### 5.2.2. Implications for the SE field in general

There is a severe decoupling between research in the computing field and the state of the practice of the field. That is particularly problematic in the SE field. Studies have shown that some technical advances are ignored almost entirely, and others take, on average, 15 years to move from initial discovery to general practical usage [25]. The findings were reinforced more recently by Ref. [8], who explored a so-called ‘assimilation gap’ between the first acquisition of a new technology and its 25% penetration into software development organizations. They found the gap, as defined here, to be 9 years for relational databases, 12 years for Fourth Generation Languages, and an astonishingly longer period for CASE tools, in which penetration was so slow that it had not reached 25% when their study concluded!

There have been few if any studies of why technology transfer takes so long. It would be easy to conclude that there may be two major potential problems: the irrelevance of the research, and the intransigence of the practitioners. But, to the best of our knowledge, there has been little more than editorial opinion on the subject of which of these two is the more likely culprit.

Research into both of these problem areas seems warranted. In the absence of any such research, it is difficult for us to form any implications based on our study. However, it is important to say at least this: to the extent that irrelevance of the research is a problem, it is the task of SE research to overcome that problem. Just perhaps, if SE researchers were able to use the findings of this study to seek ways to improve their work, the field of SE might benefit immeasurably.

## 6. Conclusion

In this paper, we examined the state of SE research from the point of view of the following research questions:

1. What topics do SE researchers address?
2. What research approaches do SE researchers use?
3. What research methods do SE researchers use?
4. What reference disciplines do SE researchers use as the theoretical basis for their publications?
5. At what levels of analysis do SE researchers conduct research?

To answer those questions, we categorized 369 papers in six leading research journals in the SE field, according to the above characteristics.

From that examination, we conclude that SE research is diverse in topic, narrow in research approach and method, inwardly-focused from the viewpoint of reference discipline, and technically focused (as opposed to behaviorally focused) in level of analysis.

## References

- [1] M. Alavi, P. Carlson, A review of MIS research and disciplinary development: implications for Deans/Administrators, *Journal of Management Information Systems* 8 (1992) 45–62.
- [2] W.R. Adrion, *Research Methodology in Software Engineering*, 1993 (part of Ref. [30]).
- [3] H. Barki, S. Rivard, J. Talbot, An information systems keyword classification scheme, *MIS Quarterly* 12 (2) (1988) 299–322.
- [4] M.J. Cheon, V. Grover, R. Sabherwal, The evolution of empirical research in IS: A study in IS maturity, *Information and Management*, 24 (1993) 107–119.
- [5] M. Culnan, E.B. Swanson, Research in management information systems 1980–1984: points of work and reference, *MIS Quarterly* 10 (3) (1986) 289–301.
- [6] A.F. Farhoomand, D.H. Drury, A historiographical examination of information systems, *Communication of the AIS* 1 (2000) 19.
- [7] N. Fenton, S.L. Pfeleeger, R.L. Glass, Science and substance: a challenge to software engineers, *IEEE Software* July (1994).
- [8] R.G. Fichman, C.F. Kemerer, The illusory diffusion of innovation: an examination of assimilation gaps, *Information Systems Research* Sept (1999).
- [9] R.L. Glass, T.Y. Chen, An assessment of systems and software engineering scholars and institutions, *Journal of Systems and Software* 59 (2001) 1.
- [10] R.L. Glass, *Software reflections—a pioneer’s view of the history of the field*, in: R.L. Glass (Ed.), *In the Beginning: Recollections of Software Pioneers*, IEEE Computer Society Press, New York, 1998.
- [11] R.L. Glass, I. Vessey, Contemporary application-domain taxonomies, *IEEE Software* July (1995) 63–78.
- [12] R.L. Glass, A structure-based critique of contemporary computing research, *Journal of Systems and Software* Jan (1995).
- [13] R.L. Glass, The software-research crisis, *IEEE Software* Nov (1994).
- [14] R.L. Glass, A comparative analysis of the topic areas of computer science, software engineering, and information systems, *Journal of Systems and Software* Nov (1992) 277–289.
- [15] D.G. Gregg, U.R. Kulkarni, A.S. Vinze, Understanding the philosophical underpinnings of software engineering research in information systems, *Information Systems Frontiers* 3 (2) (2001) 169–183.
- [16] S. Hamilton, B. Ives, MIS research strategies, *Information and Management* 5 (1982) 339–347.
- [17] S. Hamilton, B. Ives, Knowledge utilization among MIS researchers, *MIS Quarterly* 6 (4) (1982) 61–77.
- [18] R. Harrison, M. Wells, A meta-analysis of multidisciplinary research, In *Papers from The Conference on Empirical Assessment in Software Engineering (EASE)* (2000) 1–15.
- [19] M. Jackson, *The application domain*, Software Requirements and Specifications, Addison-Wesley, Reading, MA, 1995.
- [20] V.S. Lai, R.K. Malapatra, Exploring the research in information technology implementation, *Information and Management* 32 (1997) 187–201.
- [21] F. Land, Leo, the first business computer: a personal experience, in: R.L. Glass (Ed.), *In the Beginning: Recollections of Software Pioneers*, IEEE Computer Society Press, New York, 1998.
- [22] J. Morrison, J.F. George, Exploring the software engineering component in MIS research, *Communications of the ACM* 38 (7) (1995) 80–91 July.
- [23] W. Orlikowski, J.J. Baroudi, Studying information technology in organizations: research approaches and assumptions, *Information Systems Research* 2 (1) (1991) 1–28.
- [24] C. Potts, Software engineering research revisited, *IEEE Software* May (1993).
- [25] S. Redwine, W. Riddle, Software technology maturation, *Proceedings of the 8th International Conference on Software Engineering*, London (1985) 189–200.
- [26] M. Shaw, The coming of age of software architecture research,

- Proceedings of the International Conference on Software Engineering May (2001).
- [27] M. Shaw, Prospects for an engineering discipline of software, *IEEE Software* Nov (1990).
- [28] E.B. Swanson, N. Ramiller, Information systems research thematic: submissions to a new journal, 1987–1992, *Information Systems Research* 4 (4) (1993) 299–330.
- [29] SWEBOK, Guide to the Software Engineering Body of Knowledge, sponsored and published by the ACM and IEEE Computer Society, ongoing (the official Version 1.0 is expected to be published in the 2002 time frame, but interim drafts are available before that).
- [30] W.F. Tichy, N. Hakerman, L. Prechelt, Future directions in software engineering, summary of the Dagstuhl Workshop, *ACM SIGSOFT Software Engineering Notes* Nov (1992) 1993.
- [31] W.F. Tichy, P. Lukowicz, L. Prechelt, E. Heinz, Experimental evaluation in computer science: a quantitative study, *Journal of Systems and Software* Jan (1995).
- [32] I. Vessey, R.L. Glass, V. Ramesh, Research in computing disciplines: a comprehensive classification scheme, *Indiana University: Working Paper*, 2000.
- [33] S.M. Westin, M. Roy, C.K. Kim, Cross-fertilizations of knowledge: the case of MIS and its reference disciplines, *Information Resources Management Journal* 7 (2) (1994) 24–34.
- [34] M.V. Zelkowitz, D.R. Wallace, D.W. Binkley, Impediments to software engineering technology transfer, *Journal of Systems and Software* (2002) in press.
- [35] M.V. Zelkowitz, D. Wallace, Experimental validation in software engineering, *Information and Software Technology* 39 (1997) 735–743.