ELSEVIER

The Journal of Systems and Software 59 (2001) 99–106

The Journal of Systems and Software

www.elsevier.com/locate/jss

The determinants of visibility of software engineering researchers

J.M. Verner *, W.M. Evanco, K.W. McCain, G.W. Hislop, V.J. Cole

College of Information Science and Technology, Drexel University, 3141 Chestnut Street, Philadelphia, PA 19104, USA

Received 24 July 1999; received in revised form 10 August 1999; accepted 27 November 2000

Abstract

The objective of this paper is to examine the visibility of software engineers who are highly cited in the literature and to present an analysis of the predictors of this visibility. We selected 59 leading software engineering researchers (the subjects) from a much larger group of well-respected software engineers using three criteria: (1) frequency of co-citation for the period 1991–1997, (2) extent to which the research is representative within the field, and (3) adequate coverage of the software engineering subject field. The visibility of the subjects was determined by asking other software engineers to classify each of the subjects by research area. The percentage of respondents who were able to identify the subject by his/her research area was taken as a measure of that subject's visibility. A number of variables were used to explain visibility including the area of expertise, the breadth of the research, and the vintage and form of publication of the subjects' most cited work. © 2001 Elsevier Science Inc. All rights reserved.

Keywords: Software engineering; Co-citation analysis; Research specialties; Software engineer visibility; Logistics analysis

1. Introduction

Software engineering is rapidly maturing as a discipline and is gradually evolving into many specialty areas. More and more research papers are being published as the number of software engineering researchers increases worldwide. While researchers and practitioners may be well aware of advances in their own areas of expertise, some also follow important contributions in other relevant software engineering specialties. But it is becoming increasingly difficult for software engineers to maintain familiarity with the burgeoning spectrum of emerging specialties.

For the past five years, Glass (1998) has compiled a list of the most published software engineering researchers in the Journal of Systems and Software. However, this list, while interesting, says nothing about the extent to which these researchers and their work are known and used in the research of others. With the increasing maturity of software engineering as a discipline, a core set of authors is becoming generally known in the software engineering community. This broad acceptance is reflected in the software engineering literature through the citation of the work of these authors.

We began our work in this area with an interest in conducting a domain analysis of software engineering – mapping the body of knowledge of software engineering into major research areas using data extracted from the published journal literature combined with data gathered from currently active, publishing software engineers. Our studies have focused on the patterns of co-citation of both software engineering journals and authors, and the similarities and differences between authors' works as they are cited and perceived by those who write and cite (this research will be reported elsewhere).

Co-citation analysis is a standard technique in information and library science that uses bibliographic citations as a trace measure to analyze the structure of a body of literature and, by implication, the structure of the field producing it (White and McCain, 1989). Possible units of analysis include the individual cited work, a body of work by a particular author (i.e., an *oeuvre*), or the journal in which one or more cited articles is published. Two authors (or documents, or journals) are co-cited if both are included in a list of references. Being co-cited is used as an indicator that the work of the two authors is somehow related. The number of co-citations of the pair of authors is a measure of the closeness or strength of that relationship and the annual mean co-citation rate (AMCR) across the set of authors is a measure of their relative prominence. One might expect,

---

* Corresponding author.

E-mail addresses: june.verner@cis.drexel.edu (J.M. Verner), william.evanco@cis.drexel.edu (W.M. Evanco).

PII: S 0 1 6 4 - 1 2 1 2 (0 1) 0 0 0 5 1 - 6

for example, Grady Booch and James Rumbaugh to be co-cited in many papers related to the object-oriented (OO) approach, thus indicating a close relationship between the work of these two authors (and the existence of the OO approach as an observable structural feature of software engineering). Very high mean co-citation rates would identify them as major figures in their research specialty and in software engineering as a whole. Co-citation analysis applies multivariate techniques, such as cluster analysis, to large sets of citation data to explore the structure of a body of literature implied by co-citation patterns. This structural analysis, coupled with measures of prominence, gives a rich and informative portrait of the field.

Similarly, one can gather data on perceived relationships between authors (and their *oeuvres*) by asking experts in a field (*respondents*) to cluster the authors (*subjects*), using card-sorting or other techniques, and to label the clusters. Since the respondents are asked to identify the authors under study (at least as to their major contributions or fields of work), it is necessary to permit a response of “don’t know”. The proportion of “don’t know” responses is an inverse measure of visibility in the field. By using multivariate techniques, these data can be analyzed and the two structures – that arising from the literature and that arising from the aggregate perceptions of experts – can be compared.

We had expected that authors prominent in the software engineering literature would also be well known by our card-sorting respondents. However, we noticed during preliminary analyses that some of the subjects were known by all of our respondents, while other subjects, although having high mean co-citation rates, were relatively unknown by the respondents. Evidently, high levels of being cited and co-cited alone are not adequate to ensure a researcher’s widespread visibility. This observation led us to identify a number of other factors that might potentially affect researcher visibility.

2. Research objectives and outline

The objective of this study is to determine what factors are statistically significant in explaining the visibility of software engineering researchers within the software engineering community. To this end, and as described above, we selected a group of these subject researchers and conducted a co-citation analysis. We then asked other respondent software engineering researchers to categorize each of the subjects by his/her major areas of research in order to derive a visibility index. Then, along with a number of other variables described below, we conducted a multivariate analysis to investigate the determinants of visibility.

In the following section, we discuss the methodology used to collect the data and follow this in Section 3 with a discussion of the data used in this study, the analyses performed, and the limitations of this research. Section 4, presents some conclusions.

3. Data collection methodology

The methodology, which is described in detail below, consists of several steps. The first step involves the selection of the subjects and the extraction of their co-citation data. In the next step, the respondents were asked to categorize the subjects according to perceived areas of specialty (including a “don’t know” category). Visibility ratings were calculated from the “don’t know” responses. We then added a number of other variables that might explain the differences in the visibilities of our subjects.

3.1. Selection of authors for analysis

A publishing author whose work is cited by others in software engineering is the unit of observation in the software engineering co-citation analysis. The goal in developing the set of subject authors is to encompass, in a relatively small list of names, a broad cross-section of software engineering research. We are not simply interested in identifying a list of the most cited researchers. A list of this sort might emphasize some topic areas at the expense of others and noticeably bias our results toward the more established specialties in software engineering. Rather, we wanted to develop a manageable set of names that are broadly representative of the field as a whole.

A combination of sources was used to establish this representative list. We began by identifying authors whose publications received five or more citations in any combination of eight software engineering journals during the period 1991–1997. These journals were:

- ACM Transactions on Software Engineering and Methodology,
- IEEE Transactions on Software Engineering,
- IEEE Software,
- IEEE Concurrency (*formerly* IEEE Parallel and Distributed Technology),
- IEE Proceedings – Software Engineering (formerly IEE Software Engineering Journal),
- Information and Software Technology,
- Journal of Systems and Software,
- Software: Practice & Experience.

Separately, we examined general software engineering texts and reference materials, including the editorial boards of prominent journals and identified frequently occurring names (the “text” list). As a result, 59 well-known software engineering researchers were identified. This list included all authors who appeared on both the cited list and the text list (50 names). Nine additional names were selected from the remaining authors on the two lists with the dual objectives of being representative and diverse.

As part of our analysis, we asked the respondents to suggest additional authors (to augment the original 59 researchers) that they felt should be on the list. A total of 89 different names were suggested, 65 of them only once. Four authors – Nancy Leveson, Gerald Weinberg, Steve McConnell, and Richard DeMillo – received more than three votes and are incorporated in an expanded co-citation analysis (currently in progress in an investigation of the body of knowledge of software engineering). The lack of consensus on missing researchers beyond these four authors suggests that our list of 59 may be reasonably representative and inclusive.

3.2. Collection of co-citation data

A more detailed discussion of co-citation analysis is deferred to a subsequent paper and only a brief account is presented here. Interested readers may also consult McCain (1990) for a technical discussion of the general data collection and analysis methods associated with co-citation analyses.

Co-citation counts are relatively easily obtained by searching the *Science Citation Index* online and specifying a pair of authors' names. For instance, the command:

SELECT CA = Booch G AND CA = Rumbaugh J

gives a count of all the articles in the database that have made at least one reference to a paper or book by Grady Booch and a paper or book by James Rumbaugh (as either first or only authors). *The annual mean co-citation rate* of an author is the average number of citations of that author with other subjects in the data set. This measure is a domain-specific measure of citation visibility and thus is an indicator of a certain kind of prominence in the field. The domain-specificity of the measure is supported by use of co-citation and not just simple citation counts. For the present study, we collected co-citation counts for each possible pairing of the 59 authors in the subject authors list.

3.3. Collection of card-sorting data

The card-sorting technique is a method for eliciting perceptual data from individuals. For instance, we can discover the mental model that faculty members have of the academic structure of a university by asking them to sort cards bearing the names of departments into piles; the occurrence of two departments in the same pile implies some perceived relationship between the two. Aggregating across a large number of faculty members yields a general model of the university and its major perceptual dimensions (Biglan, 1978).

For the card-sorting data collection, we solicited participation from the software engineering community and received 46 responses. Respondents were provided with a set of cards, each card bearing the name of one of the subject authors in the study. The respondents sorted the cards into piles based on their perceptions of the relatedness of the authors' work, their perceptions of the structure of software engineering, or any other criteria they might like to use. The respondents were then requested to label the individual piles in terms of the domain knowledge that these piles represent. A pile could consist of a single author or may be labeled with "don't know these people", or "not software engineers". The respondents were also asked to fill out a short questionnaire. As part of the questionnaire, they were offered the opportunity to suggest additional names that they thought should be included in the study.

Aggregating the card-sorting data over a reasonably large number of respondents, allows us to analyze these data in the same way as the co-citation data. In this manner, one can compare the structure represented by the scholarly use of published materials (the authors as *oeuvres*) and the structure represented by respondents' perceptions of the authors (the authors as *personae*). See McCain (1988) for a similar study in the field of macroeconomics.

4. Empirical data

In this section, we identify and define the variables used in the analysis of visibility and indicate their ranges within our data set.

4.1. Visibility

The set of labels assigned to an individual author in the card sorting exercise can be thought of as a profile of perceived research areas. This profile provides the basis for several of the study variables, one of which is a measure of *visibility*, the dependent variable in this analysis. As mentioned previously a respondent may create a pile of unknown authors labeled "don't know" or comparable responses (e.g., "never heard of this person" or "no idea of research area"). The number of times an author is listed as unknown provides a measure of visibility of the subject across the respondent set. Visibility is defined as the percentage of respondents who are able to categorize a subject as having some specific research specialty.

Visibility ranges from a minimum of 41% (Weiser) to a maximum of 100% for Boehm, Demarco, and Yourdon. All of the respondents in our sample were aware of the research specialties of the authors having 100% visibility.

4.2. Annual mean co-citation rate

This variable was previously defined as the average number citations of an author with other subjects in the data set. The annual mean co-citation rate ranged from a high of 42 (Barry Boehm) to a low of 4 (for seven of the authors) with a mean of slightly more than 11.

4.3. Research breadth

The card sorting labels also provided a basis for categorizing the research breadth of the subjects. For this measure we used a 3-point Likert scale. A value of one indicates that the subject focuses his/her research within a fairly narrow area, while a value of three indicates that the research is broadly focused. This categorization is based on the number of research areas suggested by our respondents. Since the respondents were free to define their own category labels, some interpretation of the set of labels was required to assign research breadth. This assignment was initially done independently by three of the authors, after which a consensus was reached. 25 subjects were characterized as having a fairly narrow research area, 10 subjects as having broadly focused research, and the remaining 24 subjects in-between. We hypothesize that the research breadth of an author contributes to his/her visibility, as a wider audience will be acquainted with the author's work.

4.4. Research specialty area

The co-citation analysis discussed previously clustered the subjects into seven major research specialty areas.

- analysis and design,
- software architecture,
- modeling and tools,
- formal methods,
- software development and management,
- software testing,
- software measurement,

An author's area of specialty might be expected to have an impact on his/her visibility. For example, a highly specialized area such as formal methods may have a more delimited audience. Author's publishing only in this area may have less visibility. In addition, the differential effects of specialty area may somewhat reflect the interests of our pool of respondents.

4.5. Most cited publication

Two additional variables were identified based on the most cited publication of the subjects. The first of these was whether the most cited publication was a *book* or *journal* article. A book might be expected to contribute more to a subject's visibility than a journal article. 56% of the subjects had as their most cited publication, a book.

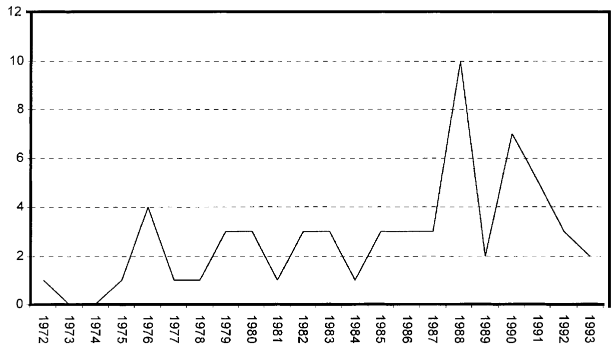

Fig. 1. Frequency of most cited publication by year.

The second variable was the *vintage* of the most cited publication expressed by the year in which the publication was issued. This study encompassed the years 1991–1997. The oldest among the most co-cited works is a paper published by David Parnas (1972). The paper is "On the Criteria to be used in Decomposing Systems into Modules" which was published by the CACM. More generally, the co-cited works tended to fall in the mid-1980s, about 5–10 years prior to the publications in which these works were cited. Fig. 1 shows graph of the frequency by year of the most cited. Books and Transactions on Software Engineering make up about 70% of the most cited publications. We hypothesize that the earlier the vintage of the most cited publication, the greater the contribution it may make to the author's visibility.

5. Analysis and results

Our investigation of visibility began with the observation that although there is some correlation between the level of visibility and co-citation rate there is substantial variation between the two variables. Fig. 2 demonstrates the extent of this variation. Subject authors are ordered from the least visible (on the left) to the most visible (on the right) with a line plot that shows the co-citation rate. Only selected authors are labeled in Fig. 2 (the full collection of authors is shown in Fig. 3).

The general trend is that the more visible authors have a higher co-citation rate, as would be expected. On the other hand, some authors deviate substantially from the trend. Those who are above the line are cited more than might be expected given their visibility. Authors below the trend line are cited less than might be expected.

The question we explore in the section that follows is exemplified by subjects such as Fagan, Pfleeger, and Mills, all of whom are very visible, but whose co-citation rates are lower than might be expected. Similarly, there

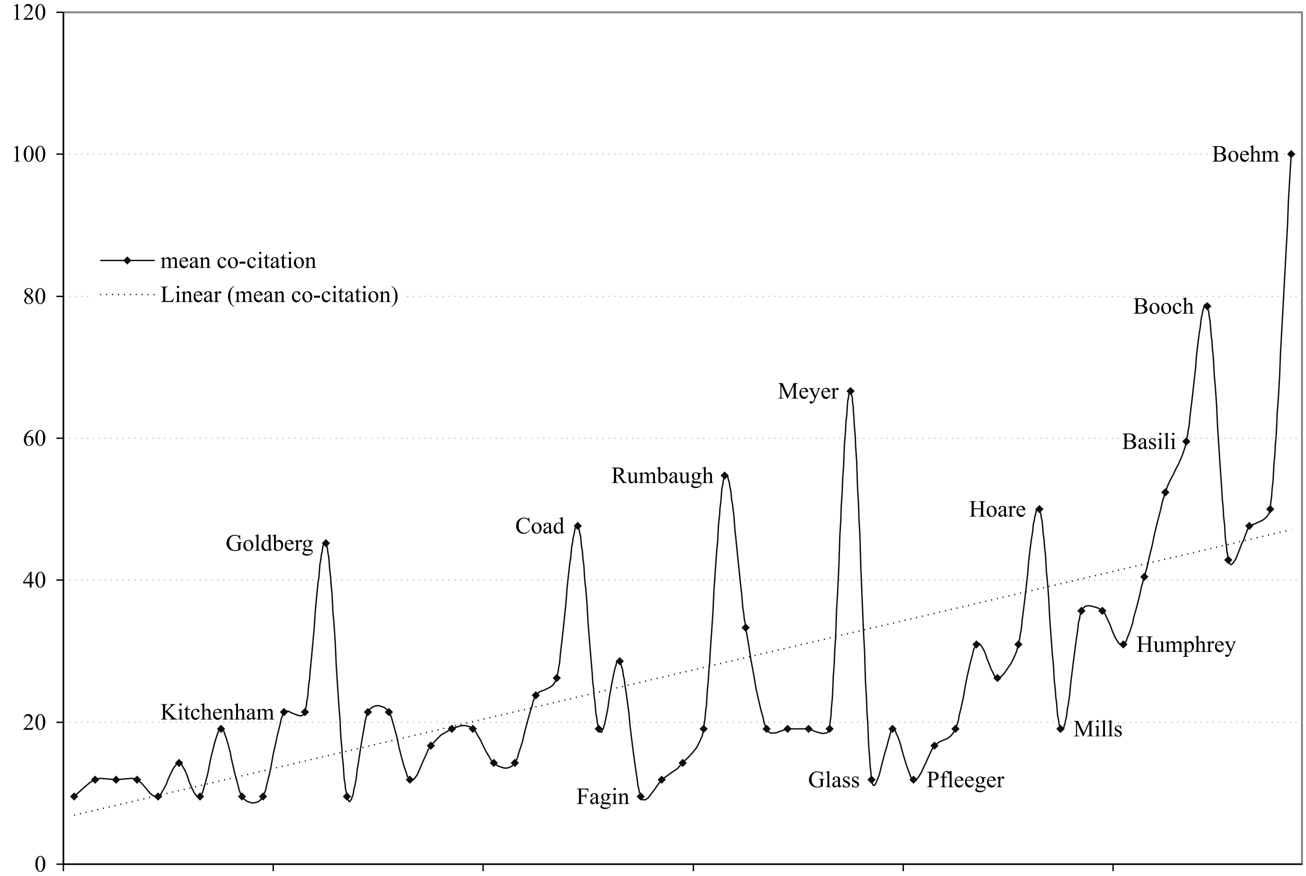

Fig. 2. Co-citation rate ordered by visibility.

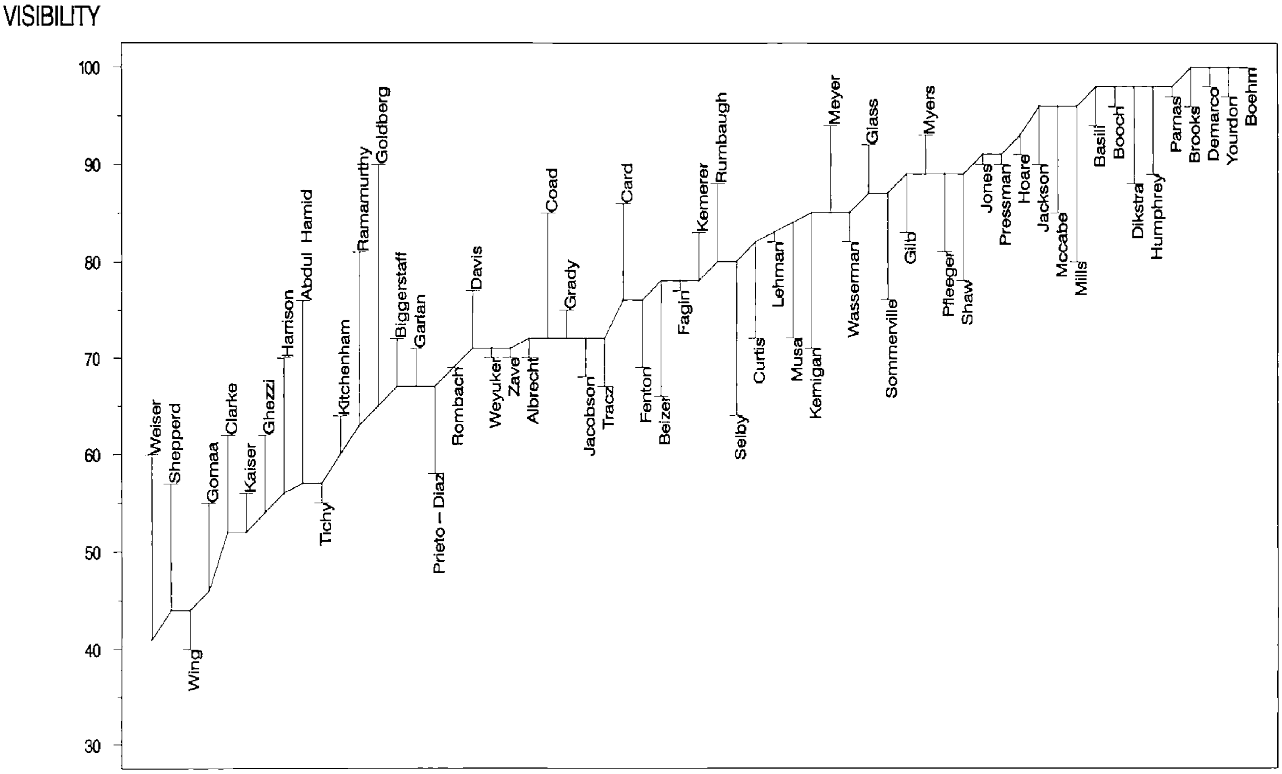

Fig. 3. Actual and predicted visibilities.

is another group of authors, all with ties to OO software, who are co-cited more than might be expected given their visibility (Goldberg, Coad, Rumbaugh, Meyer, and Booch). Thus, to further explain the variations, we included in the visibility analysis of the following section the other variables discussed above.

5.1. Predicting visibility

To investigate the possible determinants of visibility, we conducted a multivariate analysis in which the visibility of the subject author is the dependent variable. Visibility of a subject (denoted by VIS) is defined as the fraction of respondents who categorized the subject as being in a research specialty other than unknown.

The independent variables used to explain the visibility of each subject included:

- mean co-citation rate (MCR),
- research breadth,
- research specialty area,
- whether or not the subject's most co-cited publication was a book,
- year of publication of the subject's most co-cited publication.

Since the visibility variable ranges between zero and unity, the multivariate analysis was conducted using a logistics function, the specific functional form of which is:

$$\mathrm{VIS} = \frac{\mathrm{e}^z}{1 + \mathrm{e}^z}, \tag{1}$$

where Z is a function of the explanatory variables discussed in the previous section. Note that as $Z \to \infty$, $\mathrm{VIS} \to 1$.

Solving (1) for Z, we get the equation:

$$\ln\left(\frac{\mathrm{VIS}}{1 - \mathrm{VIS}}\right) = Z, \tag{2}$$

where "ln" is the natural logarithm. VIS/(1 − VIS) is called the odds ratio.

We assume that Z is a linear function of the explanatory variables given by:

$$\begin{aligned} Z = a_0 &+ a_1(\mathrm{MCR}) + a_2(B_1) + a_3(B_2) + a_4(G_1) \\ &+ a_5(G_2) + a_6(G_3) + a_7(G_4) + a_8(G_5) + a_9(G_6) \\ &+ a_{10}(\mathrm{BOOK}) + a_{11}(\mathrm{YRPUB}), \end{aligned} \tag{3}$$

where $a_1, a_2, \ldots, a_{11}$ are parameters to be estimated on the basis of the empirical data. B_1 and B_2 are dummy variables (with values of 0 or 1) indicating the breadth of the research of the subject.

$B_1 = 1$ if breadth of research is narrow,
$B_2 = 1$ if breadth of research is average.

The possible values of (B_1, B_2) are (1,0), (0,1), and (0,0). Researchers of the highest breadth are characterized by $(B_1, B_2) = (0, 0)$.

Similarly, G_1–G_6 are dummy variables characterizing the seven research specialty groups defined by the co-citation analysis as follows:

$G_1 = 1$ if a member of the analysis and design group,
$G_2 = 1$ if a member of the software architecture group,
$G_3 = 1$ if a member of the modeling and tools group,
$G_4 = 1$ if a member of the formal methods group,
$G_5 = 1$ if a member of the software development and management group,
$G_6 = 1$ if a member of the testing group.

A subject with all of the $G_i = 0$, $i = 1, \ldots, 6$ is a member of the seventh group, software measurement.

The dummy variable indicating whether or not the most co-cited publication for an author is a book is given by:

BOOK = 1 if most co-cited publication is a book
= 0 otherwise.

Finally, the year of publication for the most co-cited work of a subject is

YRPUB = year of publication of the most co-cited publication.

5.2. Analysis results

We used a maximum likelihood approach to estimate the coefficients on the basis of the VIS variable in (1), the results of which are shown in Table 1. Regression 1 uses the full collection of variables discussed above. The

Table 1
Coefficient estimates for logistics prediction model

Coefficient	Regression 1	Regression 2
Intercept	4.98	4.42
	(1.14)	(1.06)
MCR	0.096	0.098
	(0.012)	(0.012)
B_1	−1.34	−1.31
	(0.228)	(0.223)
B_2	−0.727	−0.709
	(0.230)	(0.227)
G_1	−0.543	−0.421
	(0.212)	(0.187)
G_2	−0.380	−0.253
	(0.185)	(0.157)
G_3	−0.752	−0.632
	(0.247)	(0.229)
G_4	−0.590	−0.463
	(0.189)	(0.162)
G_5	−0.212	
	(0.180)	
G_6	−0.235	
	(0.227)	
BOOK	0.690	0.638
	(0.130)	(0.124)
YRPUB	−0.044	−0.039
	(0.013)	(0.012)

coefficient of determination, R^2, for this model is 0.72, meaning that 72% of the variation of visibility is explained by the variables in the analysis. The standard errors of the coefficients are listed in parentheses. All of the coefficient estimates are significant to within the 5% level with the exception of G_5 and G_6, which are not significant at all ($> 10\%$). Consequently, in Regression 2, we eliminated the G_5 and G_6 variables from the analysis. Thus, the base group for this regression consists of a merging of the software development and management group, the testing group, and the software measurement group. The coefficient of determination for Regression 2 is 0.70 and all of the coefficient estimates are significant to within the 1% level with the exception of G_2, which is significant at the 10% level.

Note also for both regressions that the estimated coefficients for the mean co-citation rate and the research breadth dummy variables are of the expected signs. The positive sign of the coefficient for the mean co-citation rate indicates that as the mean co-citation rate increases, so does the visibility measure, VIS. Similarly, the signs for the coefficients of B_1 and B_2 are negative and the coefficient associated with B_1 is less than that for B_2, implying that subjects with less research breadth tend to have less visibility. With regard to the research specialty area membership dummy variables, subjects in the $G_3 = 1$ group, modeling and tools, have the least visibility, followed by the formal methods and analysis and design groups. Since all of the group membership coefficients are negative, the group with the most visibility is the software measurement group (i.e., all $G_i = 0, i = 1, \ldots, 6$) in Regression 1 and the merged group (software development and management, testing, and measurement) in Regression 2.

The positive sign associated with the BOOK variable indicates that publishing a book adds to a subject's visibility. The coefficient for the year of publication variable, YRPUB, is negative, leading us to the not surprising conclusion that earlier publications lead to greater visibility.

While the results of this analysis seem reasonable, we were somewhat concerned that the analysis might have been biased due to over-representation of groups G_5 (software development and management) and G_7 (software measurement) among the respondents. To test this issue, we split the subjects into two groups and reapplied the experimental analysis separately to each of these subject groups. When we did this we found minor differences in the coefficients of our estimates but no difference in signs and significance levels.

Fig. 3 shows actual visibility (as determined by the card-sorting exercise) and predicted visibility based on the logistics prediction analysis for the full set of subject authors. The authors are sorted by increasing actual visibility along the x-axis of the graph. The solid line represents the actual visibilities, while the drop lines labeled by the subjects' names show the predicted visibilities. It is not surprising that there still remains some deviations of the predicted visibilities from the actuals, because the logistic equation accounts for only 70% of the variation in the data. Factors such as the numbers of keynote speeches given, book sales, and editorial board and conference program committee participation may also contribute to visibility.

5.3. Limitations of the analysis

In concluding the discussion of our analysis, it is important to note some of the limitations of this approach. First is the fact that the analysis is retrospective by definition. We are not examining current software engineering research, but instead are mining the levels on which current work builds.

Second, in using co-citation we are selecting a particular perspective on software engineering research. Being co-cited is one possible measure of importance and acceptance of a published work, but it is certainly not a complete measure of value of research. It does not measure researcher productivity, and it may under-represent very novel work, which might be harder to tie to existing and prior work. By using co-citation analysis we are focusing rather more on citations within the software engineering disciple rather than on the more general literature. In this respect, it is worth noting that Brook's most cited work *The Mythical Man Month* (Brooks, 1975) has been cited in over 75 journals most of which are not software engineering journals.

Finally, it is important to remember that co-citation is based on first author information only. The structure of citation databases is such that no second or additional author information is available. This means that researchers who frequently publish with others as secondary authors are likely to be under-represented in this analysis or any kind of citation analysis.

None of these issues invalidate the general picture of software engineering research presented by our analysis. However, the limitations do mean that this data should not be taken as attempt to evaluate the contribution of any particular individual author.

6. Conclusions

The development of a generally recognized body of literature is an essential step in the evolution of software engineering as a discipline. The visibility of key authors is a good indicator that this evolution is continuing. In addition, the number of years that papers are highly co-cited indicates a foundation of respected software engineering research.

We began this investigation with the observation that the visibility and co-citation rate of software engineering researchers varies substantially for many of the subjects in our sample. Our analysis indicated that the most visible software engineering researchers are those whose most co-cited work is a book, who published relatively early, in a relatively broad research area, and who are frequently cited with their peers.

At the same time, we hypothesized that other factors may account for some of the unexplained variation in visibility. These factors, which are not readily quantifiable for a study like this, may include professional society activities, editorial work, consulting activities, and geographic location.

We are pleased to report that, of the other variables that we investigated but did not report in this study, gender was one that was not significant in predicting visibility. Although Transactions on Software Engineering supplied 62% of the most cited journal publications, the specific journal in which the most cited publications appeared had no significant effect on the visibility of the author. On a lighter note, for researchers interested in increasing their visibility, our analysis indicates that their first step should be to seriously consider publishing a book as soon as possible.

References

Biglan, A., 1978. The characteristics of subject matter in different academic areas. Journal of Applied Psychology 57, 195–295.

Brooks Jr., F.P., 1975. The Mythical Man Month – Essays on Software Engineering. Addison-Wesley, Reading, MA.

Glass, R.L., 1998. An assessment of systems and software engineering scholars and institutions (1993–1997). The Journal of Systems and Software 43 (1), 59–64.

McCain, K.W., 1990. Mapping authors in intellectual space: a technical overview. Journal of the American Society for Information Science 41 (6), 433–443.

McCain, K.W., 1988. Co-cited author mapping as a valid representation of intellectual structure. Journal of the American Society for Information Science 37 (3), 111–122.

Parnas, D., 1972. On the criteria to be used in decomposing systems into modules. CACM 15 (12), 1053–1058.

White, H.D., McCain, K.W., 1989. Bibliometrics. Annual Review of Information Science and Technology 24, 119–186.

June M. Verner is a professor in the College of Information Science and Technology at Drexel University. She received a BS, Master of Business Studies and Ph.D. from Massey University, New Zealand. Her primary research interests are in software metrics, software process measurement and improvement. Dr Verner is and Associate Editor for Journal of Systems and Software and on the editorial boards of Information and Software Technology and journal of Empirical Software Engineering.

William M. Evanco received the BS degree in physics from Carnegie-Mellon University, Pittsburgh, PA and the Ph.D. degree in theoretical physics from Cornell University. He is currently on the faculty of Drexel University in Philadelphia, PA where his primary interests are in software measurement and analyses and in performance modeling.

Katherine W. McCain is a professor in the College of Information Science and Technology at Drexel University. She holds two degrees in the life sciences (BS in Zoology and MS in Biology) as well as the Ph.D. in Information Studies. Her current research interests include scholarly communication, information production and use in the research process, bibliometric domain analysis of research disciplines, and evaluation of information retrieval systems. In 1998, she and her colleague Howard D. White received the "Best Paper of the Year" award from the American Society for Information Science for their domain analysis of information science–"Visualizing a Discipline: An Author Co-Citation Analysis of Information Science, 1972–1995", Journal of the American Society for Information Science 49(4), 327–355.

Gregory W. Hislop is on the faculty of the College of Information Science and Technology at Drexel University. His interests include software process and reuse, software engineering education, and online education. Dr. Hislop also has nearly 20 years industrial experience in software engineering and enterprise systems management.

Vera J. Cole is a doctoral candidate in the College of Information Science and Technology at Drexel University. Her research focuses on modeling production-type business processes for purposes of cycle time and throughput management. Ms. Cole holds a Masters of Engineering from Arizona State and a BS in mechanical engineering from New York Institute of Technology.