5-2015

Discovering the rise and fall of software engineering ideas from scholarly publication data

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

Santonu SARKAR

Sajeev A. S. M.

Nishant KUMAR

Discovering the Rise and Fall of Software Engineering Ideas from Scholarly Publication Data

Subhajit Datta*
Singapore University of Technology & Design
subhajit.datta@acm.org

Santonu Sarkar
Dept of CSIS, BITS Pilani Goa Campus, India
santonus@acm.org

A.S.M. Sajeev
Melbourne Institute of Technology, Sydney, Australia
asmsajeev@gmail.com

Nishant Kumar
Dartmouth College, Hanover, USA
Kumar.nishant1@gmail.com

ABSTRACT

For researchers and practitioners of a relatively young discipline like software engineering, an enduring concern is to identify the acorns that will grow into oaks – ideas remaining most current in the long run. Additionally, it is interesting to know how the ideas have risen in importance, and fallen, perhaps to rise again. We analyzed a corpus of 19,000+ papers written by 21,000+ authors across 16 software engineering publication venues from 1975 to 2010, to empirically determine the half-life of software engineering research topics. We adapted existing measures of half-life as well as defined a specific measure based on publication and citation counts. The results from this empirical study are a presented in this paper.

Categories and Subject Descriptors

D.2.9 [**Software Engineering**]: Management—*Life cycle*; H.5.3 [**Information Systems**]: Group and Organization Interfaces—*Collaborative computing, Computer-supported cooperative work*

Keywords

software engineering research, publication, half-life

1. INTRODUCTION

Every discipline seeks to impart to its new disciples a core set of ideas. These are perceived to be the most enduring credo that characterize the discipline, as well as serve as a basis for future development of the discipline. With time, the core ideas come to represent the gradually accumulating body of work – something in the nature of Kuhn's "normal science" [1]– that defines a collection of beliefs around which a broad consensus has developed in the discipline's community .

It was in 1968 that the phrase "software engineering" (SE) was used in public discourse for the first time [2]. We are in the fifth decade of the journey of software engineering as a discipline with a separate identity. But where are we in the organization of software engineering education around a core set of ideas? There is evidence of varied and sustained confusion about the character of software engineering. We are still debating whether software engineering will "ever be engineering" [3]; There has also been episodic efforts towards reaching a common understanding of a kernel of software engineering ideas[1]. Many of these efforts are lead by individuals or small groups, who identify a set of canons, build a framework upon them, and present it to the SE community for dialogue and debate, with a view to facilitating general acceptance of the canons as the most enduring credo of software engineering. While we recognize the value of such an approach, we believe it needs to complemented by **an empirical examination of how software engineering ideas vary in importance over time**. In this paper we report results from a study of **19,731 research papers by 21,282 authors from 1975 to 2010, a total of 36 years, across 16 publication venues**

The paper is organized as follows: The next section motivates our study and highlights some of the utility of our results in software engineering education. In the subsequent section we describe our data set, followed by a review of existing half-life measures in bibliometric analysis and the definition of the RHL metric. Our results are next presented and discussed. We then give an overview of existing literature in the context of our results. The paper ends with a summary of the results and conclusions.

*Corresponding author

WWW 2015 Companion, May 18–22, 2015, Florence, Italy.
ACM 978-1-4503-3473-0/15/05.
http://dx.doi.org/10.1145/2740908.2741734.

[1]http://semat.org

2. MOTIVATION

With the availability of Web based bibliometric repositories in the last decade, it has become increasingly easy to access large volumes of publication data. We start with the premise that as with any scientific discipline, in software engineering too, ideas wax and wane in importance over time. Older ideas making way for new ones is at the cornerstone of research. But how *quickly* new ideas appear, gain and lose importance is an interesting pointer to the level of maturity of a discipline, and hence how stable its core set of ideas are.

In this paper we invoke the notion of half-life to reflect on how long ideas remain current in software engineering literature. We examine existing formulations of half-life in similar contexts and define the **Relative Importance based Half-Life (RHL)** measure which we believe aptly captures varying importance of research topics over time. Results from applying the RHL measure help us identify several patterns whose implications we discuss in detail.

Our results can help the development, and delivery of software engineering education in the following ways:

1. There is a perception that SE is largely driven by buzzwords; fads and fashions dominate for a while and then fade into obscurity. Whether or not a particular idea remains current is largely a matter of perception, with its aficionados and detractors holding very different views. This situation is problematic when we strive to teach today's students to be tomorrow's software engineers. Which ideas do we present in a historical context, and which are the ones we connect to the state of art and practice? Such questions are customarily addressed on an ad-hoc basis, depending on a particular instructor's experience and perspective. Our results can help augment the instructor's response, by offering evidence on the patterns of changing importance of SE ideas.
2. In addition to classroom instruction, SE education is also concerned with the choices graduate students make in their selection of research topics. Such choices are often guided by a variety of factors, not the least of which is the desire to work on a "hot" topic – an area of active interest in the community. Readily identifying such topics is often not easy for a fresh graduate students, entering as (s)he is just into the field, without a deep knowledge of existing literature. The trends of topic importance we have discerned in this study, can inform graduate students' selection of specific topics for closer scrutiny and research problem definition.
3. Much of education is informed by the state of art in research. What is at the frontier of research today may likely be in the mainstream instruction tomorrow. In addition to helping shape the mainstream of SE instruction (as discussed earlier), our results can also facilitate some of the decisions SE researchers need to continually make, to remain influential in their fields. One of the key choices for a researcher is to confront the hedgehog and the fox question [4] – whether to find and pursue one defining idea for their research, or choose to synthesize varied ideas into a research agenda. The search for one defining idea or many ideas to synthesize essentially involves a review of how ideas have varied in importance in the past. The results reported in this paper can serve as a rubric to guide such a review.

Table 1: Publication Venues

TSE - IEEE Transactions on Software Engineering
TOSEM - ACM Txn on SE & Methodology
JSS - Journal of Systems and Software
IEEE SW - IEEE Software
ICSE - Intl. Conference on Software Engineering
OOPSLA/SPLASH - OO Progg, Syst, Lang. App
FSE - Intl. Symposium on the Foundations of SE
ECOOP - European Conference on OO Progg.
FASE - Intl. Conf on Fundamental Approaches to SE
ASE - Intl. Conference on Automated SE
APSEC - Asia-Pacific Software Engineering Conf
ISSTA - Intl. Conf on Software Testing and Analysis
KBSE - Knowledge-Based SE Conference
WICSA - Working Conference on Software Arch
CBSE - Component-Based Software Engineering
ISSRE - Intl. Symposium on Software Reliability Engg

In summary, the results reported this paper can inform the development of a SE curriculum, help students select research topics, and guide seasoned researchers in their development of a research plan.

3. MEASURING HALF-LIFE

Invoking the notion of *half-life* to understand the varying patterns of importance of research ideas is not new. In this section, we identify two existing approaches, outline how they can be customized in our context and then introduce our RHL measure.

As a proxy for research ideas, we have discovered *topics* from our data-set using an established natural language processing algorithm as described in the Methodology section below. In the remainder of the paper, "idea" and "topic" are used interchangeably. Our data-set is a corpus of *19,731 research papers* by *21,282 authors* from *1975* to *2010*, a total of *36* years, across the following *16* venues. Table 1 identifies each of the venues.

3.1 Existing measures

Based on bibliometric literature, we identified the following established measures of half-life for research publications:

- **Cited half-life (CHL):** CHL is a popular bibliometric measure that calculates the half-life of journals [5] with respect to a reference year [2]. We adapted CHL to measure topics instead of journals; the CHL of a topic with respect to a reference year is the median age of the papers in that topic that were cited in that year. In our analysis, the reference year is taken as 2010.
- **Prospective citation half-life (PHL):** PHL of an article or a set of articles (with respect to a reference year) is the time period over which half the citations to this set of articles were made [5]. We cannot use this definition to compute the half-life for a topic, as new papers get added to the topic over the years. Furthermore, papers in a topic do not have the same time of inception. Hence we have modified the definition as follows: We first compute the

[2]http://admin-apps.webofknowledge.com/JCR/help/h_ctdhl.htm

PHL for each paper in the topic from its year of publication till the reference year – taken as 2010 in our analysis. Then we compute the median value of each paper's PHL and consider that to be the PHL of the topic.

3.2 Relative Importance based Half-Life

To complement CHL and PHL, we propose a third measure, which captures *half-life based on year to year variations in relative importance of topics*. We call this measure *relative importance based half-life*, or RHL in short. Importance of a topic can be measured using *publication count*, that is, the annual number of papers published in a topic and/or *citation count*, that is, the number of citations received by papers in a topic in a year. One might consider publication count as measuring importance in terms of quantity and citation count as importance in terms of quality.

During the measurement period (which in our analysis was 36 years), a number of changes – increasing number of venues, easier access to publications through digital libraries etc. – are likely to influence the annual number of publications and citations. Therefore, instead of taking absolute values of publication and citation counts for each topic for each year, it is more meaningful to measure the *relative importance* of a topic in a year.

Relative publication importance of a topic is the proportion of papers that appeared in that topic out of all papers published in that year.

Relative citation importance for a topic in a year is the proportion of citations that papers in that topic have earned out of total citations for all topics in that year.

The use of relative importance measures also makes it possible to compare the results of publication-based measurement with citation-based measurement.

Based on this, we define *Relative Importance based Half-Life (RHL)* as the duration between the year in which a topic reaches its peak value of importance and the *latest* time (within the measurement-period) when it drops to or below half the peak value.

The definition of RHL allows us to distinguish two distinct clusters in the variation of importance of topics:

Decaying (D): A topic is classified to be in the decaying cluster if its relative importance eventually goes below half of its peak value and does not return above the half-peak value during the measurement period.

Sustaining (G): All other topics are classified as in the sustaining cluster. Essentially, their relative importance is either growing or their decay has not reached or remained below the half-peak value.

Evidently, half-life in RHL is defined only for topics which are in the decaying cluster. The RHL method thus helps us identify topics which are decaying in importance and hence manifest a half-life in its true spirit.

RHL vis-a-vis CHL and PHL.

In our context, RHL has the following advantages over CHL and PHL:

- Since RHL does not use a single reference year for calculations of half-life, it can be used to classify topics into clusters such as decaying and sustaining by considering *year to year variations in topic importance.*
- Unlike CHL and PHL, RHL uses a *normalized, relative measure of topic importance.*
- RHL can be used to calculate half-life based on *different measures of importance.* For instance, in our study, we use RHL to calculate half-life based on both the number of publications and number of citations. We believe this allows us to mitigate the bias which any one measure may introduce.

4. METHODOLOGY

The methodology for our study had the logical components described in the following sub-sections.

4.1 Data extraction

Information around papers published in the venues in Table 1 is available at DBLP[3]. The database dump dated April 23, 2011[4] was used for our study. The citation cross indexing was constructed using information publicly available at ACM Digital Library[5], and IEEE Xplore[6]. Paper abstracts were also extracted from these bibliographic repositories. A set of Java based components was developed to further process and analyse the data.

4.2 Topic discovery

Though the ACM classification framework [7] has a comprehensive collection of topic categories, papers published in most non-ACM publication venues are not categorized according to this framework. In view of this situation, we decided to use Latent Dirichlet Allocation (LDA), which has been widely used to identify topics from large text corpora [6].

Briefly, LDA considers a document to be a mixture of a limited number of topics $\Gamma = \{\tau_1 \cdots \tau_k\}$ and each word in the document can be attributed to one of these topics.

Here, we use the set of all papers $\mathcal{P}$ published in various SE venues mentioned in Table 1, to be our text corpus. Each document in this corpus is a stemmed set of keywords obtained from the paper title and abstract from which LDA discovers a set of topics Γ in an iterative manner.

From a text corpus LDA creates two sets of probability distributions. One of these sets models topic mixture over documents (denoted as $\Theta = \{\theta_p | p \in \mathcal{P}\}$) and the other set models keyword mixture over topics. For a paper p, we get a probability distribution θ_p over topics, and for a given topic, we get a probability distribution of keywords Arriving at the optimal number of topics (80 in our study) is an iterative process where we need to observe when the log-likelihood value becomes optimal, indicating its highest level of effectiveness [6]

LDA based topic analysis was performed using Mallet[8]. As an alternative to LDA we considered Probabilistic Latent Semantic Indexing (pLSI)[7]. However as pLSI is not a generative model unlike LDA, its results were less useful in our context. Recently, two interesting variations of the LDA model i) dynamic LDA for studying longitudinal variation of topic importance and ii) correlated topic model have emerged[8]. In our future work, we plan to investigate if

[3]http://www.informatik.uni-trier.de/ ley/db/
[4]http://dblp.l3s.de/dblp++.php
[5]http://dl.acm.org
[6]http://ieeexplore.ieee.org
[7]http://www.acm.org/about/class/1998
[8]http://mallet.cs.umass.edu

these variations are better or the basic LDA model is sufficient for our analysis.

4.3 Half-life computation

The half-lives were computed as per the formulations given in the Measuring Half-Life section. SPSS Statistics 18 was used for all statistical analysis and some of the diagrams were generated using Excel.

4.4 Topic Labelling

Automatically ascribing labels to groups of keywords constituting a topic discovered by LDA is an area of research by itself and outside the scope of our current work [9]. In this paper we manually inspected the set of keywords corresponding to the 80 topics and marked each topic by an appropriate label. To increase the reliability of the process, we requested five experienced software engineering researchers to independently ascribe labels to the topics. After the four sets of labels were received, we assigned a topic name to a keyword set if a majority of the experts chose either that name or synonymous word/phrase, or a specialization of that name. When an expert chose a name which is a specialization of another name, we chose the generic name.

As an example, for a topic with generated keywords: {`develop domain driven gener languag model specif transform uml`} experts gave the labels: "domain specific modeling", "DSL for Generating UML Diagram", "DSL development", and "Software Design". For this topic we assigned the name: "Domain specific modeling". Similarly, for a topic with keywords {`complex design larg measur metric object orient program studi`} expert delineated labels were: "OO metrics and measurement", "object oriented metrics", " Metrics for Java", and "Object Oriented Architecture". Here we chose the label to be "OO Metrics & measurement". Where there was no majority agreement among the experts, authors of this paper took the final decision of creating a topic name that best matched the keywords wherever possible. If it was not possible to arrive at any satisfactory name, we left the topic unlabelled; there were 10 such unlabelled ones in our set of 80 topics.

5. RESULTS

We measured CHL and PHL of the 80 topics with 2010 as the reference year. The CHL values ranged from 3 to 15 whereas PHL values ranged from 5 to 7. The mean and standard deviation (SD) are given below. We also give the corresponding values of RHL (publication based) and RHL (citation based). For RHL calculation, we discarded the topics that reached their half-peak values in the last five years, because their future pattern is less clear compared to those that reached their half-peak value earlier and continued to remain below that value. In the decaying category, there were 55 topics out of 80 on publications-based measurement (of which 35 reached their half-peak before the last five years), and 45 on citation-based measurement (of which 33 reached their half-peak before the last five years).

CHL mean=7.61, SD=2.58
PHL mean=5.64, SD=0.68
RHL (publication based) mean=11.46, SD=8.1
RHL (citation based) mean=10.12, SD=8.73

As we notice, there is significant difference in the half-lives calculated using CHL, PHL, and RHL. This difference is expected, as the methods are different, including the fact that RHL is calculated taking into account year to year variations over a lengthy period of time vis-a-vis a single reference year of the other methods.

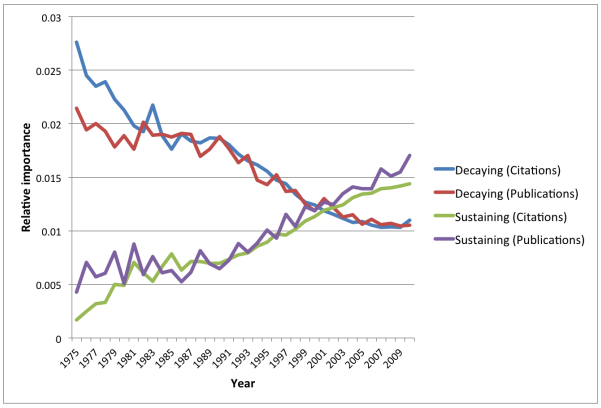

Figure 1: Trends in varying importance of topics

Though a large number of the topics exhibit half-life characteristics for both publication based and citation based measurements using RHL, the rest of the topics do not exhibit consistent decay over time. Figure 1 shows the trends in relative importance of topics on the average in each cluster for both publications and citations based measurements. Topics in the sustaining cluster on the average show steady growth in relative importance in terms of citations, whereas in terms of publications they show greater fluctuations; however, both curves follow a close trajectory. A similar pattern, albeit downwards, is observable for the relative importance of decaying topics on the average.

6. DISCUSSION

As mentioned earlier, RHL calculates half-life only for topics of decaying importance. Around 31% (25 out of 80) and 43% (35 out of 80) topics when measured by publication and citation respectively did not belong to the decaying category, thus further demonstrating that a significant number of topics continue their useful life for periods of time much longer than the conjectured five years.

6.1 Volume of Publication

The changes in the rate of publications in the two clusters also provide support for the general longevity of software engineering as a research field. In 1976, the total publications in all venues were 181; by 2010 this has grown to 1505. However, in 1976 on the average there were 2.7 papers published per topic from the decaying cluster, and 1.3 papers from the sustaining cluster; whereas in 2010, on the average, there were 15.7 papers published per topic from the decaying cluster, but 26 papers per topic from the sustaining cluster. Thus, from 1976 to 2010, the average number of papers per topic in the decaying cluster grew 5.8 times, whereas those in the sustaining cluster grew 20 times. The publication interest in sustaining topics has thus increased by more than a factor of 3 (20 to 5.8) when compared with decaying topics. Thus, not only decaying software engineering topics on the average have half-lives lengthier than the five year conjecture, but also there is a substantial increase in the proportion of sustaining topics over the years.

6.2 Topic Trends

Among the decaying topics, we can consider those in the upper quartile (i.e. top 25%) of half-lives as enduring (denoted by D-E), having *long* half-lives and those in the lower quartile (i.e. bottom 25%) as having *short* half-lives (denoted by D-S). Such a differentiation between high-value and low-value groups based on upper and lower quartiles is quite common in research fields of various domains like medicine, psychology and business (e.g. [10, 11]).

Table 2: Topic Keywords, Labels and Clusters

Keywords for a Topic	Label	Clust	
		Pub	*Cit*
1. complex design larg measur metric object orient program studi	OO Metrics & measurement	D-S	D-S
2. abstract design formal interfac languag model requir specif tool	Formal model based design	D-E	D-E
3. approach architectur develop featur line model product requir tool	Product line	G	G
4. develop domain driven gener languag model specif transform uml	Domain specific modeling	G	G
5. collabor design develop environ global knowledg manag project tool	Global collaborative dev	G	D-E
6. adapt architectur awar compon configur context evolu manag model	Software evolution	G	D-S
7. data execu invari model program specif symbol test verif	Symbolic model verification	D-E	G
8. concurr control design develop distribut environ parallel process program	Concurrent & distributed programming	D-S	D-E
9. concurr control design develop distribut environ parallel process program	Project quality assurance	D-S	D-E
10. cost develop effort estim evalu measur model qualiti reliabl	Cost, effort estimation	D-E	D-S

Figure 2 shows the behavior of a sample of ten topics outlined in detail in Table 2. Topics have been labeled using the approach described in the Methodology section. On the left side of Figure 2 are topics divided into categories based on their relative publication importance, and on the right side are the same topics, but measured in terms of relative citation importance. Each side has three categories: *sustaining* (G), *decaying with long half-life* (D-E) and *decaying with short half-life* (D-S). A topic such as Object-Oriented Metrics & Measurement has been decaying in importance with respect to both relative publications and relative citations and had a short half-life; this is irrespective of the publications in that area increasing in absolute numbers. On the other hand, Domain Specific Modelling is an example of a topic that has been sustaining in both publications and citations, indicating that papers in this area are being written as well as cited actively at a rate proportionately higher than that of decaying topics. Interestingly, a topic such as Software Evolution has a non-decaying publications count showing that it is an attractive area for publications,

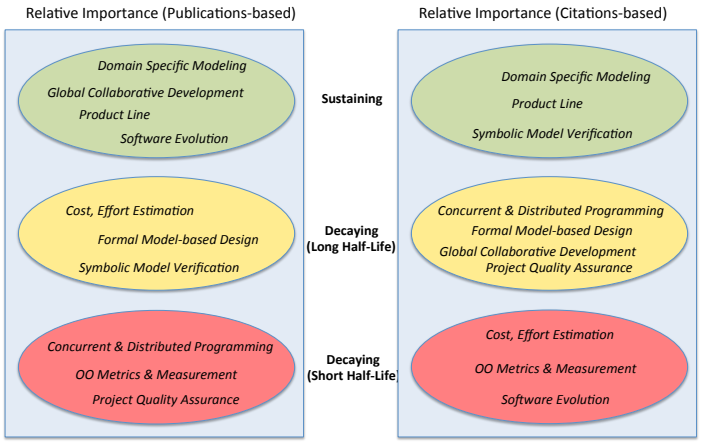

Figure 2: Topic classification example

but have reached its half-life from the citation point of view. Similarly, Global Collaborative Development is in the sustaining category with respect to publications, but its importance in terms of citations is decaying, even though, with a long half-life. In contrast, Symbolic Model Verification is an example of a topic decaying in terms of publications, but not with respect to citations.

Finally, let us select three topics namely *Concurrent & distributed programming*, *Project quality assurance* and *Cost, effort estimation* in Table 2. Their half-lives are either D-E or D-S when measured by publication or citation. Topics like *Concurrent & distributed programming* and *Project quality assurance* are in D-E category from citation point of view, indicating that researchers had been citing papers from these topics for a long time before reaching its half-life. On the contrary, a topic like *Cost, effort estimation* has endured long enough to attract publications, but researchers cited papers from these topics only for a very short duration.

7. THREATS TO VALIDITY

The conclusions from this study rest on the definition and measurement of half-life of a research topic. We have adapted definitions of half-life from existing literature, as well as introduced a definition specially suited for our context. Other definitions of half-life may alter the results. We have included a set of 16 SE publication venues based on our experience and awareness about software engineering research publications. While we do not claim this to be an exhaustive set of venues, we believe this is a reasonably representative sample for our study. Though we have cleansed the data to the extent possible, there may be some minor gaps. Thus our citation information is complete to the extent available in the public domain. Our approach to topic labeling may suffer from a degree of subjective bias. The reliability of the method can be improved by including more experts and formally deploying the Delphi method which is highly iterative, and requires higher involvement of the participants. Finally, the clustering of topics in decaying and sustaining clusters are valid within our period of measurement. A topic which has been decaying in importance in our measurement period may start growing at a later date or vice-versa.

8. RELATED WORK

Boerner et al. analyse the impact of co-authorship teams by studying a set of 614 articles by 1,036 authors between 1974 and 2004 [12]. They observe a trend towards deepening global collaboration in the production of scientific knowledge. The dynamics and evolution of scientific disciplines is studied by Herrera et al. [13]. They build an idea network of American Physical Society Physics and Astronomy Classification Scheme (PACS) numbers as nodes representing scientific concepts and use a community finding algorithm to understand the evolution of these fields between 1985-2006.

Evolution of research collaboration networks based on co-authorship information for computer science in the period 1980 to 2005 have been studied by Huang et al. [14]. They consider characteristics specific to six sub-categories within the discipline and conclude that the database community is the best connected, while the artificial intelligence community is most assortative, and computer science as a field is more similar to mathematics than to biology. Interestingly, the authors have *not* studied software engineering as a sub-category within computer science. Hassan and Holt study the collaboration networks based on co-authorship data from a very limited data-set – the proceedings of the Working Conference on Reverse Engineering (WCRE) – for the period 1993-2002 and conclude that these have properties of small-world networks [15]. Glass, Vessey, and Ramesh examine 369 papers in six software engineering publication venues and conclude that software engineering research is "... diverse regarding topic, narrow regarding research approach and method, inwardly-focused regarding reference discipline, and technically focused ... regarding level of analysis" [16].

9. SUMMARY AND CONCLUSIONS

In this paper we have argued that a deeper understanding of how ideas in software engineering research vary in importance over time can better inform the state of art and practice in software engineering education. To confront some of the perceptional bias inherent in the usual debate about what is important in software engineering vis-a-vis what is not, we empirically analysed a very large corpus of software engineering research publications to measure the half-life of ideas in terms of their varying importance. To balance our perspective on the importance of ideas, we took into account publication count as well as citation count. The process for extracting topics corresponding to ideas was based on an established natural language processing algorithm. We used a panel of software engineering experts to name each topic from its collection of keywords. The calculation of half-life enabled us to categorize software engineering research ideas in terms of their pattern of varying importance.

Kruchten conjectured a five year half-life for software engineering ideas [17]. We found empirically within our lengthy period of measurement, a significant proportion of topics are non-decaying in importance. Additionally, among the decaying topics, the mean half-life is approximately twice that of Kruchten's conjecture. More significantly, the clustering of ideas based on the half-life calculation offers an insight into how software engineering ideas interrelate to one another as they vary in importance. As education – most broadly construed – concerns itself with the cartography of ideas, our results facilitate a balanced and objective perspective on what to learn and teach in software engineering.

10. REFERENCES

[1] T. S. Kuhn, *The Structure of Scientific Revolutions*, 3rd ed. University Of Chicago Press, Dec. 1996.

[2] F. L. Bauer, L. Bolliet, and H. J. Helms, "Nato software engineering conference," 1968.

[3] M. Davis, "Will software engineering ever be engineering?" *Commun. ACM*, vol. 54, no. 11, pp. 32–34, 2011.

[4] I. Berlin, *The hedgehog and the fox: an essay on Tolstoy's view of history.* Chicago: Ivan R. Dee, Publisher, 1993.

[5] D. I. K. Sjoberg, "Confronting the myth of rapid obsolescence in computing research," *Commun. ACM*, vol. 53, no. 9, pp. 62–67, 2010.

[6] T. L. Griffiths, "Finding scientific topics," *Proceedings of the National Academy of Sciences*, vol. 101, no. suppl_1, pp. 5228–5235, 2004.

[7] T. Hofmann, "Probabilistic latent semantic indexing," in *ACM SIGIR Conf on Research and Development in Information Retrieval*, 1999.

[8] D. M. Blei, "Probabilistic topic models," *Commun. ACM*, vol. 55, no. 4, pp. 77–84, Apr. 2012.

[9] Q. Mei, X. Shen, and C. Zhai, "Automatic labeling of multinomial topic models," in *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2007, pp. 490–499.

[10] J. Riegelsberger, M. A. Sasse, and J. D. McCarthy, "Shiny happy people building trust?: photos on e-commerce websites and consumer trust," in *SIGCHI conference on Human factors in computing systems*. ACM, 2003, pp. 121–128.

[11] D. S. Freedman, L. K. Khan, M. K. Serdula, W. H. Dietz, S. R. Srinivasan, and G. S. Berenson, "The relation of childhood bmi to adult adiposity: the bogalusa heart study," *Pediatrics*, vol. 115, no. 1, pp. 22–27, 2005.

[12] K. Börner, L. Dall'Asta, W. Ke, and A. Vespignani, "Studying the emerging global brain: Analyzing and visualizing the impact of co-authorship teams," *Complexity*, vol. 10, p. 57–67, 2005.

[13] M. Herrera, D. C. Roberts, and N. Gulbahce, "Mapping the evolution of scientific fields," *PLoS ONE*, vol. 5, no. 5, p. e10355, May 2010.

[14] J. Huang, Z. Zhuang, J. Li, and C. L. Giles, "Collaboration over time: characterizing and modeling network evolution," in *Proceedings of the international conference on Web search and web data mining*, 2008, pp. 107–116.

[15] A. Hassan and R. Holt, "The small world of software reverse engineering," in *Proceedings of 11th Working Conference on Reverse Engineering*, 2004, pp. 278–283.

[16] R. L. Glass, I. Vessey, and V. Ramesh, "Research in software engineering: an analysis of the literature," *Information and Software Technology*, vol. 44, no. 8, pp. 491–506, 2002.

[17] P. Kruchten, "The biological half-life of software engineering ideas," *IEEE Software*, vol. 25, no. 5, pp. 10–11, 2008.