6-2016

How long will this live? Discovering the lifespans of software engineering ideas

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

Santonu SARKAR
BITS Pilani

A. S. M Sajeev
BITS Pilani

IEEE TRANSACTIONS ON BIG DATA, VOL. 2, NO. 2, APRIL-JUNE 2016
DOI: 10.1109/TBDATA.2016.2580541

How Long Will This Live? Discovering the Lifespans of Software Engineering Ideas

Subhajit Datta, Santonu Sarkar, and A. S. M. Sajeev

Abstract—We all want to be associated with long lasting ideas; as originators, or at least, expositors. For a tyro researcher or a seasoned veteran, knowing how long an idea will remain interesting in the community is critical in choosing and pursuing research threads. In the physical sciences, the notion of *half-life* is often evoked to quantify decaying intensity. In this paper, we study 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. In the absence of any consistent and well-accepted methodology for associating research topics to a publication, we have used natural language processing techniques to semi-automatically identify and associate a set of topics with a paper. We adapted measures of half-life already existing in the bibliometric context for our study, and also defined a new measure based on publication and citation counts. We find evidence that some of the identified research topics show a mean half-life of close to 15 years, and there are topics with sustaining interest in the community. We report the methodology of our study in this paper, as well as the implications and utility of our results.

Index Terms—Big data, software engineering, research, half-life

1 Introduction and Motivation

The contours of a scientific discipline are defined by a core set of ideas. This core is not a static mass, it evolves as new ideas replace old ones in the progression of "normal science" as conceived by Kuhn [1]. While the replacement of old ideas with new is common, the periodicity of change varies across disciplines. This variation is closely tied to the perception of a discipline from within and without. There is a perception that software engineering (SE) is largely driven by "buzzwords"; "fads" and "fashions" dominate for a while and then fade into obscurity, only to be replaced by the next fad or the next fashion [2].

1.1 The Search for Software Engineering's Identity

We are close to half a century of SE's existence as a discipline, considering that "software engineering" was used for the first time in public in 1968 [3]. But we seem to be still unsure about what constitutes software engineering. Reflections on the "unconsummated marriage" between "software" and "engineering" [4], debates whether software engineering will "ever be engineering" [5]; confusions about how it relates to other fields of computing [6], qualms on why the discipline keeps on learning and unlearning its "lessons" [7]; and misgivings around why SE fails to systematically apply what is already known [8], all point to SE's identity crisis.

- *S. Datta is with the Singapore University of Technology and Design, Singapore 487372, Singapore. E-mail: subhajit.datta@acm.org.*
- *S. Sarkar and A.S.M. Sajeev are with BITS Pilani, Goa 403726, India and with the Melbourne Institute of Technology, Melbourne, VIC 3000, Australia, respectively.*
  E-mail: santonus@goa.bits-pilani.ac.in, asmsajeev@gmail.com.

1.2 Characterizing Software Engineering Ideas

While there has been efforts at defining software engineering's character unambiguously,[1] many of these initiatives are led by individuals or small groups, who seek to identify a set of canons, build a framework upon them, and present it to the SE community for dialogue, debate, and eventual acceptance. We posit such approaches can be complemented by empirical examinations 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.

1.3 Overview of Our Approach

As Web-enabled bibliographic repositories have proliferated in the last decade and half, it has become easy to access and analyze large volumes of publication data. As researchers, educators, and practitioners in software engineering, we saw an interesting opportunity in this availability of data to determine whether and how different trends dominate the SE research ecosystem over time. Our initial premise is that, as with any scientific discipline, in software engineering too, the importance of ideas change over time. Older ideas being replaced by new ones is at the cornerstone of research. But how *quickly* new ideas appear, gain and lose importance reflects on the level of maturity of a discipline. In this paper, we use the notion of half-life for a quantitative understanding of how long SE ideas remain current. We investigate existing studies around half-life in similar contexts and define the *Relative Importance based Half-Life (RHL)* measure. Results from applying the RHL measure on our aforementioned data-set help us identify trends that influence the progression of SE.

1. http://semat.org

1.4 Organization of the Paper

In the next section we outline the research contributions of our work, followed by a discussion of how bibliometric half-life has been measured in existing literature, definition of the RHL measure, and its distinct characteristics. Subsequently, we describe our study setting and methodology. Our results and discussions are presented next, followed by their implications and utility, and threats to their validity. The paper ends with an overview of related work, and conclusions.

2 Research Contributions

Along with the burgeoning scope of big data, interest in big scholarly data has also increased in recent times. Bigger the volume of data, harder it is to find clear, crisp summary statistics that highlight aspects of interest. We identify the following contributions from our research:

1) In this work we have characterized software engineering ideas using the well-known topic model, on the basis of natural language processing of the corpus of software engineering publications.
2) We have proposed a new RHL measure to characterize life cycle of an idea, since its inception, and how the interest around the idea grows and decays from both publication and citation standpoints. As discussed later in this paper, other existing measures mostly characterized the impact and importance of a single publication, or an archival journal, but not research ideas. Unlike a paper or a journal, a research idea is formed dynamically, in a more collaborative manner, and its lifespan depends on the volume and the impact of publications. Our approach reflects these aspects of a research idea more closely than existing studies.
3) We have proposed a methodology, backed by a set of in-house tools to collect, curate and analyze the corpus of publications.

Our study has the potential to be useful to the SE community in the following ways, by way of its implications as well as future enhancements and extensions.

1) As in other disciplines, in software engineering also, whether or not a particular idea remains current is largely a matter of perception, with the idea's backers and baiters holding different views. This situation is problematic when we strive to teach today's students to be tomorrow's practitioners. 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 the degree of active interest that the community has on a particular research area. 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) One of the key choices for a researcher is to confront the hedgehog and the fox question [9]—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.
4) Our results can contribute to the increasing body of literature around analysis of scientific publication data to discern useful insights—the "science of science"—that informs individual as well as organizational decisions in the scientific enterprise [10].

3 Existing Approaches

The quest for empirical insights from scientific publication data has long interested researchers. We give below an overview of approaches which are most relevant to the study reported in this paper.

One of the pioneering studies was conducted by de Solla Price who had the ambitious goal of describing ". . . in the broadest outline, the nature of the total world network of scientific papers" [11]. Price's insights on how papers accumulate citations led to his formulation of the preferential attachment mechanism that is considered one of the foundations of network growth models [12]. Price focused on the dynamics of citations, and concluded that patterns of references reflect the orientations of scientific disciplines.

Glanzel introduced the diachronous (prospective) and synchronous (retrospective) approaches to understanding the ageing of scientific literature [13]. He underscores how the prospective and retrospective views are not merely counting citations in opposite directions and proposes a stochastic model to describe both processes. On the basis of Glanzel's insights, we compare and contrast results from a prospective approach vis-a-vis our study, in subsequent sections.

Journal Citation Reports as maintained by Thomson Reuters[2] present a set of parameters to indicate scientific influence of journals. The *citing half-life* [14] and *cited half-life* [15] are relevant metrics we consider in our subsequent discussion.

Wang et al. seek to establish a mechanism for quantifying long term scientific impact [16]. The authors propose a model to predict the citation count of a paper over a given period of time. For building the model, they have considered a well-known survival function and used this function in the context of paper publication. This is an interesting study, leading to some general insights on how papers retain importance in the long term, as well as specific metrics to measure aspects of that interest. For example, the authors define the *impact time* metric as the characteristic time it takes for a paper to collect the bulk of its citations [16]. However, as we are interested in the patterns of gain as well as decline in importance of ideas, and our study is not predictive in nature, and thus specific

2. http://admin-apps.webofknowledge.com/JCR/help/h_toc.htm

TABLE 1
Existing Metrics Vis-a-Vis RHL: An Overview

Metric/Model	Definition/Context	Remarks
Cited half-life	The number of years (counting backward from and including that year) for a journal, which accounts for 50% of the citations received from the sample of journals being considered; reflects how quickly a particular journal's article ceases being cited [15].	RHL is defined for a topic rather than a journal, and in our analysis we take into account publication as well as citation counts.
Citing half-life	The median age of all articles cited in a given sample of articles for a particular year for a journal; reflects how quickly a journal stops citing articles published by itself or other publications [14].	The insights based on RHL are not confined to specific journals.
Prospective half-life	The time period over which half the number of citations for a set of articles are made [13]; reflects how much attention a set of articles are likely to attract.	Instead of speculating on future events, insights from RHL are based on historical data.
Price Index	Proportion of publications cited by a paper that is no more than five years older than the paper that is citing; large value reflects a discipline with active current research [11].	Calculation of RHL is not based on any arbitrary boundary condition such as five years, hence the analysis based on RHL is free from biases introduced by such conditions.
Beauty coefficient and awakening time	For a given paper, this approach compares its citation history and a reference line, that is computed from its publication year, the maximum number of citations received in a year within the observation period, and the year when such maximum is achieved and identifies the year in which the abrupt change in citation accumulation happens[17].	RHL is not defined for explaining a specific phenomenon such as awakening of a paper's importance after prolonged dormancy, thus RHL has wider applicability.
Long term impact & impact time	Mechanistic model and metrics for citation dynamics of individual papers that helps synthesize citation histories of papers from different sources into a single curve [16].	RHL is based on topics rather than papers, and analysis based on RHL leads us to discern several different characteristics of varying topic importance.

measures such as impact time as proposed in [16] are not suited for direct comparison with our approach.

It is sometimes observed in different disciplines that a paper suddenly starts gaining attention, after lying in relative obscurity since its publication. Ke et al., have called such papers "sleeping beauties" (SB) and defined metrics to study the phenomenon [17]. The goal is to find out the time when the importance of a paper gets recognized since it is published. Furthermore, it is assumed that the importance is recognized with a sudden spike in its popularity. To capture these two notions, the authors first introduce the idea of a straight line starting from the year of inception till it reaches the peak citation (t_{max}), with a coordinate system where the x axis represents the year, and the y axis represents the citation count. The sleeping beauty quotient B is a value that indicates whether a paper typically remains dormant (citation count graph is mostly below the line, resulting in a positive B) or active (the graph is above the line, resulting in a negative B). The authors also bring the notion of an awakening time t_a, such that $t_a <= t_{max}$, when the distance between the citation count graph with the straight line is maximum. Ke et al.'s study is an important contribution to the field of scientometric research. The motivations for studying the SB phenomenon resonate in a limited sense with our focus on the varying importance of research topics.

In examining the details of Ke et al.'s approach, we observe that, in formulating the metrics, the authors did not consider the possibility of multiple years where the citation can reach its peak. Even for a fixed year where the citation reaches the peak, there can be multiple awakening times. The authors did not discuss how this is addressed. More importantly, the metrics consider absolute values of the citation counts. This can be misleading since one of the contributing factors in the overall growth of citations for a paper can be the increasing volume of publications in that discipline over time. Our methodology, as described in subsequent sections, tries to address some of these challenges in the related, but different content of our study.

In Table 1, we have summarized how the existing studies outlined above relate to our approach.

4 Ideas and Their Impact

Identifying the key ideas or a research topic on which a paper has been published, is a non-trivial task. ACM has defined a taxonomy of various computer science related topics.[3] However, tagging the paper with an appropriate set of keywords from such a classification framework is left to

3. https://www.acm.org/about/class/2012

the discretion of the author. Furthermore, papers published in most non-ACM publication venues are not categorized according to this framework. While collecting the publication data from the publicly available source (described later in this paper), we observed that these keywords, which would have been an important source for research topic identification, are often not available. While it may be possible to manually classify a paper using a prescribed framework, it is not practical when the data-set is large. As a proxy for research ideas, we therefore have resorted to an automatic discovery of *topics* from our data-set using an established natural language processing algorithm as described in Section 6. A topic is a collection of keywords which are thematically linked; we discuss our topic discovery process in detail in a following section. In the remainder of the paper, "idea" and "topic" are used interchangeably.

After identifying the topics, it is necessary to understand how "impactful" a particular idea has been, since its inception. To measure such impact we have used the concept of *half-life*. Invoking the notion of half-life to understand the varying patterns of research importance is not new. However, to the best of our knowledge, half-life based metrics have not been used to measure the importance of a *research topic*.

The metrics defined in the following subsections are applied on a corpus of data as described in Section 6.

4.1 Relative Importance Based Half-Life (RHL)

We now define our metric, Relative importance based half-life.

4.1.1 Defining RHL

The basic assumption behind our notion of half-life is that topics *decay* in importance over time. So, to be able to measure a topic's half-life it must be decreasing in importance over time. With this background, we define the half-life as follows:

Definition 1. *Let us denote a set of topics by* Γ *and each individual topic by* $\tau \in \Gamma$. *Let us further denote the numerical value of the importance of a topic* τ *by* $\nu(\tau)$. *The* half-life $\mathcal{H}(\tau)$ *of a topic* τ *is the duration between the time of its peak value of importance and the* latest *time (within the measurement-period) when it drops to or below half the peak value. Let* t_{max} *be the time such that* $\nu(\tau)[t_{max}]$ *is the maximum, and* t_{half} *be the time such that* $\nu(\tau)[t_{half}] = (\frac{\nu(\tau)[t_{max}]}{2})$ *and* $(\forall t > t_{half} : \nu(\tau)[t] < \nu(\tau)[t_{half}])$. *The half-life of* τ *is computed as* $\mathcal{H}(\tau) = |t_{half} - t_{max}|$.

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 to reflect importance in terms of quantity and citation count to reflect importance in terms of quality. For a balanced sense of a topic's importance, it is essential to take a relative view, rather than an absolute one.

4.1.2 Computing Relative Importance

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 have influenced 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.

To define the relative importance, let us first define a relation $PaperTopic(\tau)$ to be the set of the papers $p \in \mathcal{P}$ that have been classified under the topic τ. The set of all topics is denoted by Γ. Next we use the notation $p.y$ to denote the publication year of p, and we denote the total number of papers in a given year t to be $\mathcal{P}(t)$. We create two sets of frequency distributions for each year from 1975 till 2010. The first one is the publication count based frequency distribution $\nu_p(\tau)$ for a topic. For a year t, this is defined as follows:

$$\nu_p(\tau)[t] = |\{p | p \in \mathcal{P}(t) \cap PaperTopic(\tau)\}|, \forall \tau \in \Gamma.$$

Next, we define the citation count based frequency distribution $\nu_c(\tau)$ for each year from 1975 till 2010. For this, first we denote the set of papers that have cited a particular paper p during our observation period (1975 till 2010) to be $\mathcal{C}ref(p)$. From this we then compute the set of papers that have cited p in a given year t, i.e., $\{p' \in \mathcal{C}ref(p) | p'.y = t\}$, where $p'.y$ is the publication year of the paper p'. Now we define $\nu_c(\tau)$ as:

$$\nu_c(\tau)[t] = \sum_{p \in Z} |\{p' \in \mathcal{C}ref(p) | p'.y = t\}|, \forall \tau \in \Gamma,$$

where $Z = \{P(t) \cap PaperTopic(\tau)\}$

From the above, the relative publication importance and the relative citation importance can be easily computed, and they are denoted by $\overline{\nu_p}$ and $\overline{\nu_c}$ respectively.

Based on the relative importance measures, we define *Relative Importance based Half-Life* 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. As an example, Fig. 1 shows the variation in the relative importance of citations of a topic from 1975 to 2010; the topic's half-life is six years (1986-1980).

In subsequent discussion "RHL" would either refer to the above definition of half-life or the method used for arriving at it, as will be clear from the context. Trying to compute the RHL of topics 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 (as in Fig. 1).

Sustaining (G): All other topics are classified as in the sustaining cluster. Essentially, their relative importance is

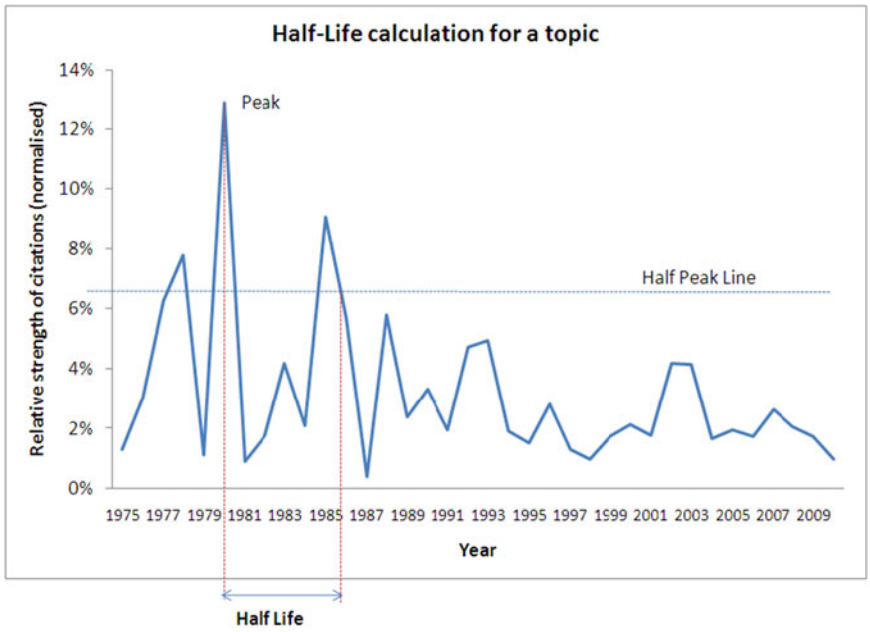

Fig. 1. Half-life calculation.

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.

5 Distinct Aspects

The metric and approach outlined above seek to complement existing studies outlined in Section 3 to characterize research impact. While we recognize the contributions of existing studies, let us highlight the aspects that set aside our approach.

5.1 Characteristics and Focus of Our Approach

Our approach is based on a topic rather than a paper (as in the study of sleeping beauties [17]) or a collection of papers published in a journal (as in the study of cited and citing half-lives [14], [15], [18], and prospective half-life [13]), or a common characteristic across a multi-disciplinary corpus of papers (as in the models of citation attraction [11] or long term impact [16]). While it may be interesting to study individual papers or journals in some contexts, we believe discerning the patterns of varying importance of topics lets us characterize a discipline such as software engineering better. Researchers are primarily concerned about ideas, and our mechanism of topic discovery allows us to isolate ideas more precisely in our context.

In our approach, we have considered the relative notion of importance vis-a-vis an absolute one, which we believe is more realistic. Moreover, the SB phenomenon is studied in [17] with reference to a paper. While studying sleeping beauties, the variation of a paper's citation count is not studied beyond the time maximum citation is reached. In contrast, our approach is topic-based and considers a topic's entire lifetime within our study period. As we consider the entire lifetime of a topic's importance, we are able to capture the common pattern of the initial hype, followed by disinterest and (possibly) subsequent resurrection of interests.

A phenomenon such as sleeping beauty [17] may be relevant for disciplines like the natural and social sciences, which have a long history of published research. In a relatively young discipline like software engineering, it is uncommon to find a paper lying dormant for long and then surging in importance. On the other hand, patterns of varying importance of research topics is common across disciplines. So, our approach as illustrated on a corpus of SE publications has a *wider applicability* beyond a particular discipline.

The focus of our study is to find out *how various research topics gain importance and how long they remain attractive to the community*. Research ideas emerge out of a collective effort of researchers publishing various papers in different venues. To the best of our knowledge, the inception, popularization and eventual decline of an idea is not completely captured by existing approaches.

5.2 Comparison with Relevant Metrics

For an objective assessment of our approach vis-a-vis the closest among existing ones, we choose following metrics for comparison of results.

- *Cited half-life (CHL):* CHL is a popular bibliometric measure that calculates the half-life of journals [18] with respect to a reference year.
- *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 [13], [18].

The above mentioned metrics have not been defined keeping a research idea in mind. The granularity of the CHL metric is at the level of a journal, not at an individual paper. PHL, on the other hand is meant for an article. The main challenges in using either of them in the context of a topic are:

TABLE 2
Publication Venues and Other Details

TSE - IEEE Transactions on Software Engineering
TOSEM - ACM Transactions on Software Engg. & Methodology
JSS - Journal of Systems and Software
IEEE SW - IEEE Software
ICSE - Intl. Conference on Software Engineering
OOPSLA/SPLASH - Object-Oriented Progg, Systems, Lang. & App.
FSE - Intl. Symposium on the Foundations of Software Engg.
ECOOP - European Conference on Object-Oriented Programming
FASE - Intl. Conf on Fundamental Approaches to Software Engg.
ASE - Intl. Conference on Automated Software Engineering
APSEC - Asia-Pacific Software Engineering Conference
ISSTA - Intl. Conference on Software Testing and Analysis
KBSE - Knowledge-Based Software Engineering Conference
WICSA - Working Conference on Software Architecture
CBSE - Component-Based Software Engineering
ISSRE - Intl. Symposium on Software Reliability Engineering
Total number of years (1975 to 2010, both inclusive) - 36
Total number of venues - 16
Total number of papers - 19,731
Total number of authors - 21,282

- A topic comes out of a collection of conference and journal papers.
- The topic, as a collection, dynamically grows over time as as new papers get added to the topic over the years.
- Papers in a topic do not have the same time of inception (or a reference year as required by PHL).
- A paper can belong to multiple topics, unlike a journal or a conference.

Given these constraints, the original definitions of these metrics had to be customized to fit our topics based approach so that the results can be compared (Section 7).

We have adapted CHL to measure topics instead of journals; we consider the CHL of a topic τ with respect to a reference year—taken as 2010 in our analysis—as the median age of the papers in $PaperTopic(\tau)$ that were cited in 2010. Since, we cannot use the original definition of PHL to compute the half-life for a topic, we have modified the definition as follows: We first compute the PHL for each paper $p \in PaperTopic(\tau)$, from its year of publication $p.y$ 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.

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.

Unlike CHL and PHL, the sleeping beauty metric [16] can not be tailored to the topic level, thus there is no scope of comparing the SB metric values with RHL. Since SB is at the paper level, papers in a topic will have different t_{max} and t_a, and aggregating these two values at the topic level will not be meaningful. Furthermore, our approach considers relative notion of importance vis-a-vis an absolute one, which we believe is more realistic. While studying sleeping beauties, the variation of a paper's citation count is not studied beyond the time when maximum citation is reached (t_{max}). In contrast, our approach considers a topic's entire lifetime within our study period. As we consider the entire lifetime of a topic's importance within the study period, we are able to capture a larger pattern of varying importance.

A summary of how existing approaches relate to our approach is presented in Table 1.

6 Study Setting

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 2 identifies each of the venues. We have taken 2010 as the end year to offer reasonable time for gaining importance to the later publications in our measurement period.

Fig. 2 gives an overview of the methodology of our study. The major components are described in the following sub-sections, and the results are discussed in the next section.

6.1 Data Extraction

Information around papers published in the venues in Table 2 is available at DBLP.[4] 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 analyze the data.

6.2 Topic Discovery

Latent Dirichlet Allocation (LDA) has been widely used to identify topics from large text corpora [19]. 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. Given a corpus of documents, LDA discovers a set of topics, keywords associated with each of the topics and the specific mixture of these topics for each document in the corpus. Here, we use the set of all papers $\mathcal{P}$ published in various SE venues mentioned in Table 2, 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 as shown in Fig. 3.

4. http://www.informatik.uni-trier.de/ ley/db/
5. http://dl.acm.org
6. http://ieeexplore.ieee.org

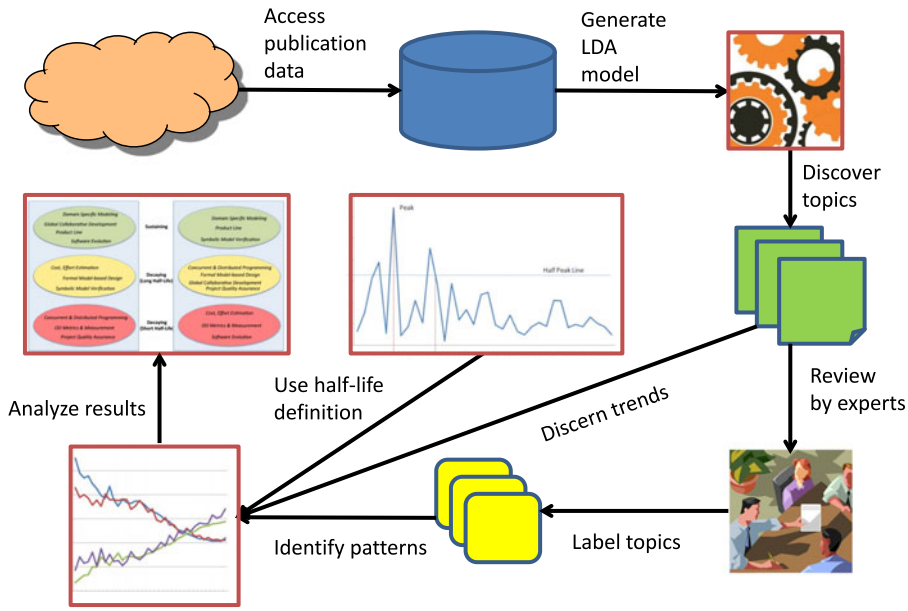

Fig. 2. Methodology of the study.

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 i) a probability distribution θ_p over topics, and for a given topic, ii) we get a probability distribution of keywords (See Fig. 3). In LDA, these two are taken to be Dirichlet distributions with parameters α and β respectively. Arriving at the optimal number of topics for a given corpus is an empirical process. We need to vary α, β, number of iterations (N) and number of topics (K) to get the log likelihood value for the model that indicates its highest level of effectiveness [20]. Iterating over these parameters several thousand times, we selected 80 topics for our study. Beyond this number, we noticed that instances of repetitions in the keywords across the topics increased substantially, thus indicating a low possibility of identifying further distinguishable topics.

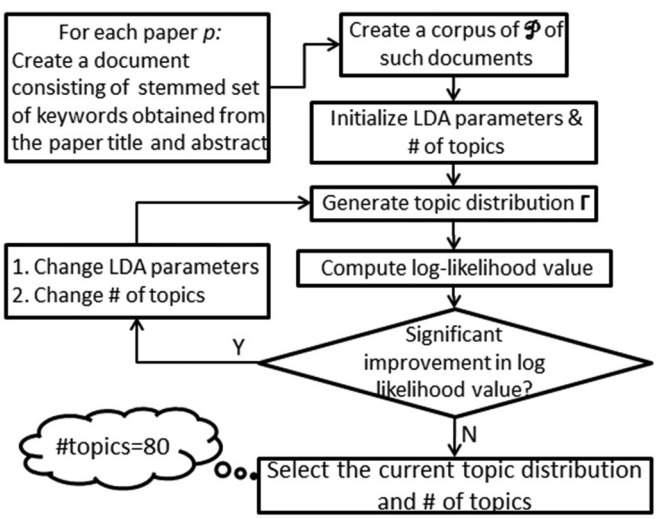

Fig. 3. Flowchart of topic analysis.

6.3 Determining Paper versus Topic Relation

Let us elaborate the process of creating $PaperTopic(\tau)$. Recall that LDA generates a probability distribution Θ_p for a paper p. Thus, $\Theta_p(\tau)$ is the probability that a topic τ is present in the paper p. Though $\sum_{k=1}^{K} \Theta_p(\tau_k) = 1$, for our corpus, if we order the topics with the decreasing order of probability values for any Θ_p, $\sum^{1} 0_{k=1} \Theta_p(\tau_k)$ lies within 0.8–0.9. Thus, it is sufficient to take the top 10 topics for a given Θ_p. Once we complete this pruning process for all papers in $\mathcal{P}$, we can create the set $PaperTopic(\tau)$ for each topic τ. In our current study, we however, do not consider the probability values while computing the frequency distributions values ν_p as well as ν_c.

LDA based topic analysis was performed using Mallet.[7] As an alternative to LDA we considered Probabilistic Latent Semantic Indexing (pLSI) [21]. However as unlike LDA, pLSI is not a generative model, its results were less useful in our context.

6.4 Topic Labeling and Half-Life Computation

Associating meaningful labels to topics discovered from our corpus can facilitate an intuitive understanding of their varying importances in the software engineering context. 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 [22]. 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 four experienced software engineering researchers to independently ascribe labels to the topics. To facilitate the process of name selection and improve consistency, we advised them to consult Microsoft Academic Search.[8] We followed the following set of guidelines while finalizing the topic labels:

1) After the four sets of labels were received, we assigned a name to a topic (which is a keyword set) if a majority

7. http://mallet.cs.umass.edu
8. http://academic.research.microsoft.com

TABLE 3
CHL, PHL, and RHL Values

Half-life formulation	Mean	Standard deviation	Confidence interval
RHL (publication based)	11.46	8.1	between 8.77 to 14.14
RHL (citation based)	10.12	8.73	between 7.14 to 13.1
CHL	7.61	2.58	between 7.05 and 8.18
PHL	5.64	0.68	between 5.48 and 5.78

of the experts chose either that name or a synonymous word/phrase, or a specialization of that name.

2) Whether a topic name is a synonym for another topic name was judged by discussion between the experts and the authors of this paper.
3) When an expert chose a name which is a specialization of another name, we chose the generic name.
4) Where there was no majority agreement among the experts, authors of this paper took the final decision on creating a topic name that best matched the keywords wherever possible.
5) 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.

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 "Object oriented metrics and measurement".

The half-lives were computed as per the formulations given in Section 4. SPSS Statistics was used for all statistical analysis and some of the diagrams were generated using Excel.

7 Results and Discussion

7.1 Insights from Calculating RHL

On the basis of our measurements of RHL (publication based) and RHL (citation based), Table 3 gives the mean, standard deviation and the 95 percent Confidence Interval (CI) (i.e., the range of values within which the mean of samples will lie with 95 percent probability). 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).

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. Fig. 4 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.

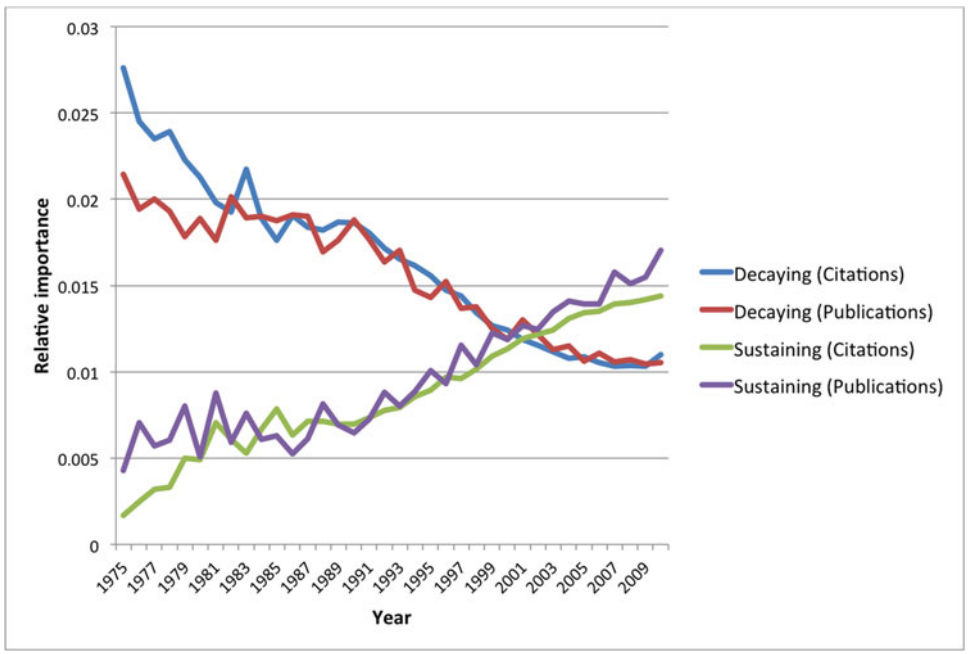

Fig. 4. Trends in varying importance of topics.

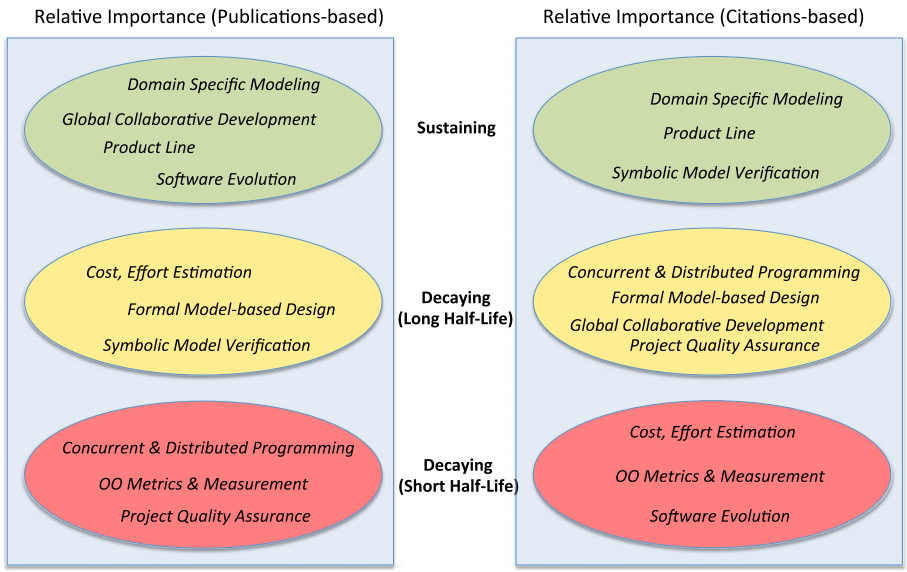

Fig. 5. Topic classification example.

7.2 Comparing RHL Vis-a-Vis CHL and PHL

Table 3 also gives values of CHL and PHL of the 80 topics with 2010 as the reference year. 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 for the other methods.

While the approaches for identifying sleeping beauties [17] and RHL are not directly comparable (as we have explained earlier in Sections 3 and 5), we nonetheless computed the beauty coefficient B [17] for the papers in our corpus. We observed that 45.54 percent papers have +ve B, indicating that they are more or less dormant. Only 32.32 percent papers have -ve B, indicating that they have been actively cited since their inception, whereas 22.14 percent papers are B neutral, indicating that they were neither dormant nor active till they reached their respective t_{max} years.

7.3 Trends in Varying Topic Importance

In a 2008 article in *IEEE Software*, Kruchten conjectured that the "half-life of software engineering ideas is roughly five years" [23]. Based on the measures in our study, the mean half-life of SE topics is greater than the five years conjectured by Kruchten (PHL is the closest to five years). As mentioned earlier, we note that RHL calculates half-life only for topics of decaying importance. Around 31 percent (25 out of 80) and 43 percent (35 out of 80) topics did not belong to the decaying category, when measured by publication and citation respectively. This further demonstrates that a significant number of topics continue their useful life for periods of time much longer than the conjectured five years.

In order to analyze the topics in software engineering research that have attracted higher interest in publishing, let us revisit the publication data once again. In 1976, the total publications across all venues were 181; by 2010 this has grown to 1505. However, in 1976 on 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 three (20 versus 5.8) when compared with decaying topics. Thus, not only do 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. Furthermore, Fig. 4 shows that the relative importance of the sustaining topics on the average have been increasing over the years. These trends offer an interesting perspective on the longevity of software engineering research topics.

7.4 Longevity of Topics

Given a topic manifests half-life, we want to be able to get a sense of how long the topic importance remained current. Among the decaying topics, we can consider those in the upper quartile (i.e., top 25 percent) of half-lives as enduring (denoted by D-E), having *long* half-lives and those in the lower quartile (i.e., bottom 25 percent) 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. As an example, Riegelsberger et al. classified an e-commerce vendor as trusted or untrusted based on whether the vendor was in the upper or lower quartile of a set of measurements [24]. As another example, in a study on obesity, Freedman et al. defined over-fat as those with mean skin fold thickness in the upper quartile of their subject group [25].

Fig. 5 shows the behavior of a sample of ten topics.[9] Topics have been labeled using the approach described in Section 6.4. On the left side of Fig. 5 are topics divided into

9. Please refer to bit.ly/1Tyvw5K for a full description of topics and their keywords, a selection of which are outlined in detail in Table 4 and Fig. 5.

TABLE 4
Topic Keywords (Stemmed), 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 engineering	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 development	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. cycl develop life manag model perform project qualiti requir	Project quality assurance	D-S	D-E
10. cost develop effort estim evalu measur model qualiti reliabl	Cost, effort estimation	D-E	D-S

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). We see evidence that 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 modeling* 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 non-decaying publications showing that it is an attractive area for publications, 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 look at the three topics, *Concurrent & distributed programming*, *Project quality assurance* and *Cost, effort estimation* in Table 4. (In the Appendix section, Tables 5 and 6 give a listing of the topics which could be classified in either of the three categories.) 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.

7.5 Towards an Identity for Software Engineering

In the introductory discussion, we have motivated this study with the need to address software engineering's supposed identity crisis, among other factors. Let us now examine whether our results can help define an identity for SE. We may recall that the common perception is that SE is characterized by more than expected churn, with a procession of ideas coming in and going out of "fashion" [26]. If reality matched perception, we would expect our results to reveal all topics to be in the decaying with short half-life (D-S) category. As we have seen in the immediately preceding discussion, this is not the case. We see a mixture of topics with sustaining interest, decaying interest with long half-life and decaying interest with short half-lives. This is indicative of a matured discipline where old ideas give way to new ones - some faster than others - while some other ideas continue to attract a steady interest from the community. Thus we find no empirical evidence that SE research topics quickly decline in importance. The *extent* to which SE topics stay current versus topics in other discipline(s) can be determined by replicating our study on publication corpora of other disciplines; this is planned as future work. Our approach is discipline agnostic, so such a replication will not pose any technical challenges.

In summary, this study indicates software engineering to be a discipline that *does not* show notable fickleness in the changing importance of research topics, contrary to some common perception. Thus, these results can better inform the characterization of SE as a discipline.

7.6 Utility of Our Study

In addition to the general research contributions identified in Section 2, let us identify how our methodology and results can be useful in practice.

1) The approach for calculating and interpreting RHL illustrated in this paper can be applied to other disciplines beyond SE. After controlling for peripheral factors, comparison of RHL values can help distinguish between disciplines in terms of how long ideas remain current. This will facilitate a more objective discussion whether a particular discipline concerns itself with enduring ideas, vis-a-vis evanescent ones.
2) Using RHL, a new researcher entering a discipline can gain perspectives on whether particular research topics are decaying or sustaining in importance. This can help her choose an appropriate area of research.
3) We have used LDA to extract a set of topics and mapped each paper to the set of topics using a probability distribution. Using the outcome of LDA, it is also possible to analyze papers being written by a new researcher and map these papers into existing topics. Combined with the RHL based analysis, it

can be suggested whether these newly written papers will likely to be decaying or sustaining in importance.

4) Promotion and tenure decisions are based on research impact. Frequently, such decisions are taken by considering number of publications and citations, and/or metrics closely related to these measures, such as the H-Index [27]. These approaches have been questioned, and the need for a more meaningful reflection on research impact are being actively sought in recent times [28]. There is an emerging consensus that researchers need to be evaluated on the basis of longer term impact of their work [29]. Applying the RHL measure on an individual's body of work can help discern how interest in the topics of her work has varied over time. We believe this can effectively complement existing ways of measuring research impact.

8 Threats to Validity

As in any research, there are threats to the validity of our results. We discuss these limitations below with respect to *construct validity*, *internal validity*, *external validity* and *reliability*.

Construct validity implies that variables are measured correctly. In our case, other than PHL and CHL as outlined in Section 4, we have not been able to identify other definitions of half-life in existing literature that are applicable to the importance of research topics. Therefore, we defined RHL in a manner that is consistent with the spirit of half-life most relevant to our context. Measuring impact and importance of a publication by counting the number of citations is widely prevalent, though not without qualms [28]. By design, RHL is a non-cumulative measure. This choice enables us to avoid some of the well known biases of cumulative measures of research impact, related to the *Mathew effect* [12]. However, it makes RHL more sensitive towards fluctuations in interest around research topics. We recognize these threats to construct validity, however their extent does not invalidate the general direction of our results.

A study shows *internal validity* if it is free from systematic errors and biases. Since our data-set is derived from information available in the public domain for a predefined set of publication venues, issues that can affect internal validity such as mortality (that is, subjects withdrawing from a study during data collection) and maturation (that is, subjects changing their characteristics during the study outside the parameters of the research) do not arise in our case. We believe we have accessed the maximal amount of data in our scope that is available in the public domain. We have chosen 16 major publication venues that focus on software engineering research. However, selection bias can occur from the manner in which venues of publications are selected for the study. Although our data-set covers a major portion of the discipline's research publication corpus, we cannot claim to have captured all published software engineering papers in 1975-2010. Whether or not a particular venue included in our study exclusively focuses on software engineering is a matter of judgment, as is the question of what is truly a software engineering paper. Thus it is likely our corpus consists of some papers which relate to SE only in a broad sense, and we have missed out some software engineering related papers published in other venues. As mentioned earlier, the citation counts were based on citation cross indexing between papers that we constructed across several of our data sources. For the papers for which citation information is not available in the public domain, could not be included in our analysis. A common problem in studies of scientific publication comes from the ambiguity of author names. Such inconsistencies are minimized in DBLP through significant human intervention [30]. This threat does not relate to our study as our unit of analysis is topics, rather than authors.

External validity indicates the generalisability of the results of a study. The population for our study is all software engineering publications. Even though our sample size and the sampling method are unlikely to be a threat to external validity, the segregation of topics into clusters are based on observations during the measurement-period. Thus, the categorization 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. We do not claim our results to be generalizable across disciplines. We elaborate this point below.

Whether the conclusions from our results can be generalized to extend to another discipline depend to a significant extent on the nature of that discipline. If the topics in that discipline are strongly defined by short living ideas which die fast and reincarnate in whole or as parts of subsequent ideas, our RHL based analysis may not reveal this metamorphosis. The software engineering ideas we consider in this paper are extracted as topics from research publications only. Thus, ideas which have never appeared in a research publication - but may have had significant impact on the state of art or practice - are not considered in this study. If a discipline is strongly influenced by ideas that are heavily guarded through patents and trade-secrets before appearing in the public domain as market-ready products, calculating the RHL on academic papers in such a discipline may not be particularly insightful.

Reliability of a study is related to reproducibility of the results. A threat in this context arises from the fact that topics were identified automatically using the LDA approach and then manually labeled by a panel of experts as described in Section 6.4. Manual labeling is a subjective process for which repeatability may be a threat. 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 [31]. 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 [32]. In our future work, we plan to investigate if these variations are better or the basic LDA model is sufficient for our analysis.

9 Future Work

In addition to addressing the above limitations in our future work, we plan to expand our study to other

disciplines beyond software engineering. The code framework behind the work presented here can be made available to the SE community at large by exposing our analysis methods and the associated tooling as an on-line "service" that always maintains an up-to-date publication corpus, evolves the topic model over time, and maintains association among various entities like papers, venues, authors, citations and the topics. This service can potentially create a community of its own that can improve the quality of the topic model and topic labels through active discussion and feedback. The service can provide an year-on-year analysis of various topics that are sustaining vis-a-vis decaying in importance, in terms of growing number of publications and citations. The analysis can help individuals and organizations in identifying fields with varied and lively research topics. Furthermore, the topics extracted by our framework and the set of papers associated with a topic can also help researchers identify related body of work for a given area of interest. We have not considered the probability values from the LDA model in calculating the importance values used to compute the half-life measures; we plan to include these probability values while computing the importance values when we replicate the study on other disciplines.

10 Related Work

We have compared our approach around RHL with existing studies in earlier sections. We now summarize the comparison and give a brief overview of related work in the area of analyzing research publication data.

10.1 Summary of Comparison

Unlike cited half-life [15] and citing half-life [14], insights from RHL relate to research topics rather than specific publication venues, and take into account publication as well as citation information. While prospective half-life speculates [13] on the level of interest that may be generated on a set of articles in future, RHL's insights are based on historical data. RHL differs from the Price Index [11] as RHL does not impose any arbitrary boundary condition of five years to determine research impact, and RHL considers a research topic rather than a single paper as its unit of analysis. RHL has a wider applicability than Beauty coefficient [17], as it does not seek to capture a particular phenomenon. RHL can complement mechanistic [16] and generative [33] models of research impact, by discerning patterns of varying importance of research topics.

10.2 Analyzing Research Publication Data

Boerner et al. analyze the impact of co-authorship teams by studying a set of 614 articles by 1,036 authors between 1974 and 2004 [34]. They observe a trend towards deepening global collaboration in the production of scientific knowledge. Bettencourt et al. study publication data from six different fields and infer that, while each field develops differently over time, population contagion models adapted from epidemiology can generally explain their development [35].

The dynamics and evolution of scientific disciplines is studied by Herrera et al. [36]. They build an idea network of American Physical Society's 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. [37]. 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. Bird et al. construct a collaboration network for computer science, define 14 sub-areas (including software engineering) and use topological measures to examine behaviors of individuals and collaboration patterns across areas in terms of how centralized, integrated and cohesive they are [38]. The authors of this paper have only considered seven venues for software engineering, all of them conferences, which in our opinion does not offer a representative sample of SE publication data.

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 [39]. 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" [40]. The same set of authors have also compared methods and topics between what they call the "three major subdivisions of the computing realm" - computer science, software engineering, and information systems - and conclude that each field has preferred research approach and methods, which is not necessarily "respected" by the other fields [41].

While complementing these existing studies, our work introduces a standard for calibrating the patterns of varying importance of research topics.

11 Summary and Conclusions

The vitality of a research discipline is defined by the progression of its ideas. The movement of ideas across inception, acceptance, importance, obscurity, and occasional revival influence how the discipline is viewed by academics and practitioners. In this paper, we sought to understand how software engineering ideas vary in importance over time. We defined the Relative Importance based Half-Life (RHL) measure and applied it on a set of research topics extracted from a large corpus of software engineering research publications. Calculation of RHL on the basis of publication as well as citation counts helped us confront some of the perceptional bias inherent in the debate about what is important in software engineering vis-a-vis what is not.

TABLE 5
Topic Categorization by Publication Based RHL

Sustaining (G)	Decaying with long half-life (D-E)	Decaying with short half-life (D-S)
Agile development, Code inspection, Automated test generation, Domain specific modeling, Architectural modeling, Challenges of open source development, Cryptography, Multi agent programming, Product line engineering, Mobile agent programming, Automated bug detection, Project cost estimation, Aspect orientation, Dynamic analysis in object oriented programming, Modular design, Authentication, Clone detection, Global Collaborative Development, Domain specific architectural modeling, Software evolution, Defect and fault estimation, Web services modeling, Model driven design.	*Architecture reuse and integration, Extreme programming, Fault detection, Object oriented programming, Project execution and estimation, Reliability and fault modeling, Reuse of processes, Formal model based design, State machine modeling, Round trip engineering, Symbolic model verification, Protocol modeling andspecification, Web services architecture and performance, Cost, effort estimation.*	*Pattern based software development, Reuse and CASE tools, Object oriented design environments, Configuration management, Load balancing, Debugging, Object oriented modeling, Process maturity model, Project quality assurance, Object oriented metrics and measurement, Component oriented software engineering, Resource allocation, Concurrency detection and program slicing, Concurrent and distributed programming, Data types.*

TABLE 6
Topic Categorization by Citation Based RHL

Sustaining (G)	Decaying with long half-life (D-E)	Decaying with short half-life (D-S)
Regression test coverage, Aspect orientation, Model driven design, Component oriented software engineering, Cryptography, Round trip engineering, Automated test generation, Defect and fault estimation, Multi agent programming, Web services modeling, Protocol modeling and specification, Project cost estimation, Agile development, Automated bug detection, Modular design, Dynamic analysis in object oriented programming, Authentication, Domain specific modeling, Object oriented programming, Product line engineering, Reuse of processes, Clone detection, Mobile agent programming, Symbolic model verification, Maintenance and support, Pattern based software development, State machine modeling, Challenges of open source development, Process management practice.	*Code inspection, Concurrent and distributed programming, Formal model based design, Industrial case studies, Configuration management, Requirement specification and object oriented design, Concurrency modeling, Project Quality Assurance, Stochastic modeling, Resource allocation, Global collaborative development.*	*Algorithmic optimization, Concurrency detection and program slicing, Web services architecture and performance, Debugging, Architecture reuse and integration, Object oriented metrics and measurement, Software evolution, Protocol verification and synthesis, Fault detection, Cost, effort estimation, Extreme programming, Reliability and fault modeling, Object oriented design environments.*

Our results reveal that within our lengthy period of measurement, a significant proportion of topics are non-decaying in importance. Among the decaying topics, the mean RHL is significantly more than the conjectured five year half-life of SE ideas. Additionally, we find evidence of short and long half-lives among decaying topics. Analysis of our results point to the nuances of varying importance of SE research topics, and the dangers of characterizing any discipline by a perception of the durability of its ideas. Although the study has been carried out on a corpus of SE research publications, we believe our results can initiate an informed discussion around the life-cycle of ideas in various computing disciplines. Our methodology can also facilitate objective decision making by individuals and organizations in the pursuit of research excellence.

12 Appendix

Tables 5 and 6 lists the topics that could be labeled and classified in either of the three categories G, D-E, and D-S on the basis of publication and citation counts respectively. These tables offer a ready reference for software engineering topic categorization on the basis of RHL. SE researchers, practitioners, and students can refer to these tables to get a sense of the pattern of interest in a particular topic. On the other hand, if a particular topic seems to generating a lot of recent "buzz", its presence, (or absence) in the list can indicate how much (if any) interest it has generated in the community. Additionally, Tables 5 and 6 can help understand how materials covered in SE textbooks relate to the patterns of varying importance of research topics.

References

[1] T. S. Kuhn, *The Structure of Scientific Revolutions*, 3rd Ed. Chicago, IL USA: Univ. Chicago Press, Dec. 1996.

[2] I. Jacobson, Everyone Wants to be Agile, 2008. [Online]. Available: http://blog.ivarjacobson.com/everyone-wants-to-be-agile/. Accessed on: Jul. 28, 2010.

[3] F. L. Bauer, L. Bolliet, and H. J. Helms, "NATO software engineering conference," 1968, http://homepages.cs.ncl.ac.uk/brian.randell/NATO/NATOReports/

[4] D. L. Parnas, "Software engineering: An unconsummated marriage," *Commun. ACM*, vol. 40, no. 9, Sep. 1997, Art no. 128.

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

[6] M. Young and S. Faulk, "Sharing what we know about software engineering," in *Proc. FSE/SDP Workshop*, 2010, pp. 439–442.

[7] M. V. Zelkowitz, "What have we learned about software engineering?" *Commun. ACM*, vol. 55, no. 2, pp. 38–39, Feb. 2012.

[8] D. L. Parnas, "Risks of undisciplined development," *Commun. ACM*, vol. 53, no. 10, pp. 25–27, Oct. 2010.

[9] I. Berlin, *The Hedgehog and the Fox: An Essay on Tolstoy's View of History*. Chicago, IL, USA: Ivan R. Dee, Publisher, 1993.
[10] G. Goth, "The science of better science," *Commun. ACM*, vol. 55, no. 2, pp. 13–15, Feb. 2012.
[11] D. J. de Solla Price, "Networks of scientific papers," *Science*, vol. 149, no. 3683, pp. 510–515, 1965. [Online]. Available: http://www.sciencemag.org/content/149/3683/510.short
[12] M. E. J. Newman, "The structure and function of complex networks," *Soc. Ind. Appl. Math. Rev.*, vol. 45, pp. 167–256, 2003. [Online]. Available: http://arxiv.org/abs/cond-mat/0303516
[13] W. Glanzel, "Towards a model for diachronous and synchronous citation analyses," *Scientometrics*, vol. 60, no. 3, pp. 511–522, 2004. [Online]. Available: http://dx.doi.org/10.1023/B%3ASCIE.0000034391.06240.2a
[14] T. Reuters, Journal citation reports - citing half-life, 2012. [Online]. Available: http://admin-apps.webofknowledge.com/JCR/help/h_ctghl.htm. Accessed on: Dec. 8, 2015.
[15] T. Reuters, Journal citation reports—-cited half-life, 2012. [Online]. Available: http://admin-apps.webofknowledge.com/JCR/help/h_ctdhl.htm. Accessed on: Dec. 8, 2015.
[16] D. Wang, C. Song, and A.-L. Barabsi, "Quantifying long-term scientific impact," *Science*, vol. 342, no. 6154, pp. 127–132, 2013. [Online]. Available: http://www.sciencemag.org/content/342/6154/127.abstract
[17] Q. Ke, E. Ferrara, F. Radicchi, and A. Flammini, "Defining and identifying sleeping beauties in science," in *Proc. Nat. Acad. Sci.*, 2015, vol. 112, no. 24, pp. 7426–7431. [Online]. Available: http://www.pnas.org/content/112/24/7426.abstract
[18] D. I. K. Sjoberg, "Confronting the myth of rapid obsolescence in computing research," *Commun. ACM*, vol. 53, no. 9, pp. 62–67, 2010.
[19] T. L. Griffiths, "Finding scientific topics," in *Proc. Nat. Acad. Sci.*, 2004, vol. 101, no. suppl_1, pp. 5228–5235.
[20] H. M. Wallach, I. Murray, R. Salakhutdinov, and D. Mimno, "Evaluation methods for topic models," in *Proc. 26th Annu. Int. Conf. Mach. Learn.*, 2009, pp. 1105–1112.
[21] T. Hofmann, "Probabilistic latent semantic indexing," in *Proc. ACM Spec. Int. Group Inf. Retr. Conf Res. Dev. Inf. Retr.*, 1999, pp. 50–57.
[22] Q. Mei, X. Shen, and C. Zhai, "Automatic labeling of multinomial topic models," in *Proc. ACM Spec. Int. Group Knowl. Discovery Data Conf. Knowl. Discovery Data Min.*, 2007, pp. 490–499.
[23] P. Kruchten, "The biological half-life of software engineering ideas," *IEEE Softw.*, vol. 25, no. 5, pp. 10–11, Sep./Oct. 2008.
[24] J. Riegelsberger, M. A. Sasse, and J. D. McCarthy, "Shiny happy people building trust?" Photos on e-commerce websites and consumer trust," in *Proc. Spec. Int. Group Comput.-Hum. Interact. Conf. Hum. Factors Comput. Syst.*, 2003, pp. 121–128.
[25] 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.
[26] I. Jacobson, P.-W. Ng, P. E. McMahon, I. Spence, and S. Lidman, "The essence of software engineering: The SEMAT Kernel," *Commun. ACM*, vol. 55, no. 12, pp. 42–49, Dec. 2012.
[27] J. E. Hirsch, "An index to quantify an individual's scientific research output," in *Proc. Nat. Acad. Sci. USA*, vol. 102, no. 46, Nov. 2005, pp. 16569–16572.
[28] D. L. Parnas, "Stop the numbers game," *Commun. ACM*, vol. 50, no. 11, pp. 19–21, Nov. 2007. [Online]. Available: http://doi.acm.org/10.1145/1297797.1297815
[29] J. Grudin, "Technology, conferences, and community," *Commun. ACM*, vol. 54, no. 2, p. 41, Feb. 2011. [Online]. Available: http://cacm.acm.org/magazines/2011/2/104400-technology-conferences-and-community/fulltext
[30] M. Ley and P. Reuther, "The problem of data quality," *EGC*, vol. RNTI-E-6, pp. 5–10, 2006.
[31] S. Datta, *Software Engineering: Concepts and Applications*. Oxford, UK: Oxford Univ. Press, 2010.
[32] D. M. Blei, "Probabilistic topic models," *Commun. ACM*, vol. 55, no. 4, pp. 77–84, Apr. 2012.
[33] T. Chakraborty, S. Kumar, P. Goyal, N. Ganguly, and A. Mukherjee, "On the categorization of scientific citation profiles in computer science," *Commun. ACM*, vol. 58, no. 9, pp. 82–90, Aug. 2015. [Online]. Available: http://doi.acm.org.library.sutd.edu.sg:2048/10.1145/2701412
[34] 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, pp. 57–67, 2005.
[35] L. M. A. Bettencourt, D. I. Kaiser, J. Kaur, C. Castillo-Chvez, and D. E. Wojick, "Population modeling of the emergence and development of scientific fields," *Scientometrics*, vol. 75, no. 3, pp. 495–518, 2008.
[36] M. Herrera, D. C. Roberts, and N. Gulbahce, "Mapping the evolution of scientific fields," *PLoS ONE*, vol. 5, no. 5, May 2010, Art no. e10355.
[37] J. Huang, Z. Zhuang, J. Li, and C. L. Giles, "Collaboration over time: Characterizing and modeling network evolution," in *Proc. Int. Conf. Web Search Web Data Min.*, 2008, pp. 107–116.
[38] C. Bird, E. Barr, A. Nash, P. Devanbu, V. Filkov, and Z. Su, "Structure and dynamics of research collaboration in computer science," in *Proc. Sub-Divisional Magistrate*, 2009, pp. 826–837.
[39] A. Hassan and R. Holt, "The small world of software reverse engineering," in *Proc. 11th Working Conf. Reverse Eng.*, 2004, pp. 278–283.
[40] R. L. Glass, I. Vessey, and V. Ramesh, "Research in software engineering: An analysis of the literature,"*Inf. Softw. Technol.*, vol. 44, no. 8, pp. 491–506, 2002.
[41] R. L. Glass, V. Ramesh, and I. Vessey, "An analysis of research in computing disciplines," *Commun. ACM*, vol. 47, pp. 89–94, Jun. 2004.

Subhajit Datta is currently a lecturer at the Singapore University of Technology and Design. He has more than 16 years of experience in software design, development, research, and teaching at various organizations in the US, India, and Singapore. He is the author of the books *Software Engineering: Concepts and Applications* (Oxford University Press, 2010) and *Metrics-Driven Enterprise Software Development* (J. Ross Publishing, 2007), which are widely used by students and practitioners. His research interests include software architecture, empirical software engineering, social computing, and big data. Subhajit received the PhD degree in computer science from the Florida State University. More details about his background and interest are available at www.dattas.net.

Santonu Sarkar received the PhD degree in computer science from the Indian Institute of Technology, Kharagpur. He is a professor of computer science and information systems BITS Pilani, K.K.Birla Goa Campus. He has more than 20 years of experience in IT industry in applied research, product and application development, architecture consulting, project, and client account management. His current research interests include building software engineering techniques to ensure dependability, performance, and ease-of-use of Cloud and HPC applications. Prior to this, he had extensively worked in different fields of software engineering, namely in the area of software metrics and measurement, software design and architecture analysis, program comprehension, and reengineering techniques. His other research interests include analysis of social networking data.

A. S. M. Sajeev received the PhD degree in computer science from Monash University. He is the director of Sydney Campus, at the Melbourne Institute of Technology, Australia. Previously he was the chair in information technology/computer science at the University of New England, Australia. He is a fellow of the Institution of Engineers, Australia. He has published widely in software engineering and other disciplines in computer science. His research interests include empirical software engineering and business information systems.