Using the case survey methodology for finding high-quality grey literature in software engineering

Ashley Williams
ashley.williams@mmu.ac.uk
Department of Computing and Mathematics
Manchester Metropolitan University
Manchester, UK

Jim Buchan
jim.buchan@aut.ac.nz
Software Engineering Research Lab (SERL)
Auckland University of Technology
Auckland, NZ

ABSTRACT

Background: Mining online accounts of practitioner experience can provide new evidence for software engineering researchers. However, we need methods for assessing quality at vast scale. **Objectives**: We investigate the challenge of finding high-quality grey literature, defining high-quality in terms of a document's relevance to the consumer and its credibility. **Method**: Building on previous research, we use a version of the case survey methodology for automating the identification and assessment of high-quality grey literature. **Results**: We develop a model of credibility assessment within software engineering research and demonstrate our case survey methodology and credibility assessment model in practice. We use it to conduct a grey literature review of High Performing Teams (HPT). **Conclusions**: The paper provides a foundation for future research on automated quality and credibility assessment. Adoption of the tools and methodology presented can help researchers effectively search for and select higher-quality blog-like content.

CCS CONCEPTS

• **General and reference** → **Empirical studies**; • **Information systems** → **Blogs**.

KEYWORDS

credibility assessment, grey literature, software practice, data quality, case survey methodology

1 INTRODUCTION

Traditionally, software engineering researchers collect practitioner opinions through survey and interviews. Today's social programmers often communicate their opinions and experiences through social media however [31, 32]. This provides a new source of practitioner-based evidence that is highly accessible and freely available, referred to as grey literature.

One such source, and the focus of our research, is blog-like content [27]. We define a blog-like article as: publicly available online with an identifiable author and date; written from personal or professional experience; supports discussion through interactive elements like comments; and is editable and regularly maintained (c.f. white papers, forums, company websites).

Garousi *et al.* [9] advocate the adoption of grey literature into software engineering research as a way of bridging the gap between the state-of-art and the state-of-practice. The authors argue that grey literature provides a novel perspective and helps to avoid publication bias. This is true of blog-like content also: blog-like articles provide access to practitioner experience (including experience which may not be available through other means); blog-like articles are created and owned by an individual, meaning that they may provide visibility into actual software practice; blog-like articles are timestamped, meaning that researchers can assess trends over time; and blog-like articles are widely available but often unexplored by research, and so can be used in conjunction with other sources for multi-method triangulation.

There are of course challenges in using blog-like content as evidence. The key challenge is that there is no reliable method for identifying articles which are of 'high-quality' to research. Further to this, there is large quantity of content available on the web to filter. Considering the well-documented researcher subjectivity and context dependence on the meaning of 'high-quality' with respect to evidence, it is difficult to design a process that will be suitable for all situations. We define 'high-quality' in terms of an article's relevance to the consumer, as well as its credibility. In this paper, we present work towards addressing these challenges through the development of a credibility model for software engineering researchers, and an adapted version of the case survey methodology. We also present an empirical demonstration of the methodology in practice.

This paper takes a design science approach, presenting a methodology that we have developed and predict will be able to help software engineering researchers in identifying high-quality blog-like content for use as evidence in their research. We demonstrate the methodology in practice and reflect on the methodology's successes, short-comings, and future direction. The key contribution of this paper is that we present a semi-automated case survey methodology for systematically finding and assessing high-quality grey literature in software engineering research.

2 RELATED WORK

2.1 Evidence in software engineering research

Evidence Based Software Engineering (EBSE) seeks to integrate best evidence from research with practical experience and human values [12]. Kitchenham *et al.* [12] present EBSE as a means for improving the overall quality of software engineering research. Wohlin [40] also advocates EBSE, and presents the concept of evidence-profiles for categorising different types of evidence in order to acquire a holistic understanding of the evidence being presented in a particular case. Wohlin acknowledges a difference in the method for synthesis of evidence depending on whether the synthesis is being conducted in research or practice. In research, evidence is synthesised to objectively describe the phenomena being observed. Whereas in practice, evidence must be evaluated in different contexts as evidence reported in a specific context may not generalise to all situations. In practice, Rainer *et al.* [24] conclude that many software projects are managed without reference to empirical evidence and make suggestions such as the use of innovation diffusion theory and persuasive communication theory to increase the inclusion of empirical evidence.

Evidence is often presented in order to influence opinion to that of the writers. Researchers typically seek fact and empirical data to inform opinion. However, Devanbu *et al.* [8] and Rainer *et al.* [23] (amongst others) observe that practitioners form opinions based on their own professional experience, and the experience of their peers. Given that research often gathers evidence through practitioner survey and interview, there is an implication that this evidence is, at least partly, based on experience. Research addresses this by taking the opinions of multiple practitioners and finding patterns. However, the use, or even acknowledgement of prior beliefs, influence, experience and anecdotal evidence can be a contentious issue within software engineering research.

In assessing the quality of practitioner-generated blog-like content, we have to assess the evidence that is presented. This evidence may be experience and/or anecdote, and we may have no indication of evidence's rigour and truth, however it may still be useful to the researcher. We therefore define evidence as “an artefact provided with the intention of persuading the reader to embrace an opinion or belief". This artefact may be presented by the practitioner in any form (data, images, written experience etc.). This working definition implies that perceived evidence is assessed without evaluating the quality and truth of that evidence (actual evidence). This has implications for the quality assessment of blog-like content we propose later.

2.2 Systematic reviews

Budgen and Brereton [5] present four benefits that systematic reviews bring to software engineering research: 1) systematic reviews provide an objective summary of research evidence concerning a topic or phenomenon [12]; 2) authors benefit by having a clear set of procedures to follow in reviewing background material, and for identifying where the material supports or conflicts with their own work; 3) by producing better quality reviews and evaluations, the quality of papers improves; 4) the experience of conducting systematic reviews brings with it a number of transferable skills. Three common types of systematic review are:

Systematic Literature Reviews (SLRs). Kitchenham *et al.* [11] define SLRs as a means of identifying, evaluating and interpreting all available research relevant to a particular research question, or topic area, or phenomenon of interest.

Systematic Mapping Studies (SMSs). SMSs differ from SLRs in that they aim to provide an overall picture of the state of a research topic. Peterson *et al.* [21] describe guidelines for conducting systematic mapping studies that states “while systematic reviews aim at synthesising evidence, also considering the strength of evidence, systematic maps are primarily concerned with structuring a research area".

Rapid Reviews (RRs). RRs are differentiated in that they sacrifice a portion of rigour and thoroughness in order to deliver results quicker than SLRs [6]. Rapid reviews present only the key findings of the review in favour of simplicity and speed. They are presented as being systematic because the method they follow is systematic. However, the reduction in rigour means that results are not exhaustive and complete.

Both SMSs and RRs are normally seen as precursors for a more detailed SLR.

2.3 Incorporating grey literature into systematic reviews

Blog articles are a type of social media which are used by software practitioners for sharing information, opinions, and experience. Little research exists on the use of blog articles as evidence in software engineering research. However, there is research that investigates other forms of social media (e.g. [2, 4, 29]). We propose that research should also consider blog articles as a source of evidence. Rainer and Williams [27] discuss the benefits and challenges of using blog articles as evidence. To summarise, blogs provide access to practitioner experience, novel insight into practice, and a new source of analysis for multi-method triangulation. There are many challenges however: there is no established process for assessing quality; any analysis needs to work at a potentially vast scale; and blogs vary in structure, language and formality. In relation to systematic reviews, two further types of review incorporating grey literature have emerged:

Grey Literature Reviews (GLRs). GLRs involve systematically reviewing grey literature with the goal of gaining an understanding of the opinions of practitioners.

Multi-vocal Literature Reviews (MLRs). MLRs aim to collate the views and opinions of both research and the grey literature. Garousi *et al.* [9] present MLRs as a method for closing the gap between the state-of-art where research operates and the state-of-practice. Garousi *et al.* identify other benefits such as, promoting the voice of the practitioner, and adding context and real-world relevance to research.

In contrast to the types of systematic reviews mentioned earlier, GLRs and MLRs do not assume primary studies as their document of analysis. Researchers need to be careful when analysing both grey literature documents and primary studies due to their varying and uncertain quality and rigour.

2.4 Blog-like content as grey literature

The Web hosts a large number of software engineering blogs[1], not all of which hold value to researchers. For example, we have previously analysed the blog of Joel Spolsky[2] [37]. Spolsky's blog is popular amongst the practitioner community and was identified as an example of where blogs may hold value in research [27]. Spolsky's blog contains a mixture of article types. Out of the 1,023 pages crawled, a manual annotation by the first author found that only 382 are opinion piece articles. The remaining 63% are made up of short updates, static pages and introductions to other articles. From this analysis, two challenges for research emerge: how do we classify which web pages are blog articles and which are not? and, how do we classify which are the 'right type' of blog article and which hold little value to research?

It is common for practitioners, who possess considerable industry knowledge and experience, to share this in blog articles. Seibert, Engineering Manager at Dropbox, lists four reasons why every software engineer should blog[3]: developing their personal brand, providing new opportunities for career growth, giving back to the community, and writing a blog helps the authors company (e.g. by presenting an authentic voice). The advantages of owning a blog do not necessarily align with the benefits to software engineering researchers looking to use blogs as evidence. However, of the reasons listed, the third – giving back – is likely to generate the content most relevant to software engineering research.

In research, we are aware of only Parnin *et al.* [18] that has explicitly looked at the motivators and challenges for software engineering bloggers (though research exists on blogs in other contexts). Parnin *et al.* identify four benefits to bloggers blogging: personal branding, evangelism and recruitment, personal knowledge repository and to solicit feedback.

Some researchers do not trust blog articles as credible evidence due to their varying quality [39]. However, there are situations where practitioners seek to set standards for content which is published online. For example, through setting guidelines (e.g. Robert Cartaino's guidelines for writing high quality Stack Overflow questions[4]), or through measuring audience appreciation.

3 CASE SURVEY METHODOLOGY FOR GREY LITERATURE (CS4GL)

3.1 The case survey methodology

For researchers to be able to assess the quality of blog-like content at scale, there is a need for a methodology, and for that methodology to automate as many of the required steps as possible. The four key steps that need to be carried out are: 1) search for candidate web pages; 2) select articles from the search results based on appropriate quality criteria; 3) download and extract the article text from each result; and 4) analyse the text based on the context of the study being undertaken.

Commonly researchers (e.g. [7]) use a web crawler for collating web pages that are then analysed. We present a contrasting approach that uses the case survey methodology supported by additional search heuristics and software that we are developing towards automation. This new methodology identifies multiple relevant documents, assesses the quality of the documents, and then aggregates their findings (c.f. the four steps outlined above). Larrson [14] says that case surveys "bridge the gap between nomothetic surveys and idiographic case studies to combine their respective benefits of generalizable, cross-sectional analysis and in-depth, processual analysis." Essentially, a case survey is a secondary study which surveys a number of 'cases' (we focus on blog-like documents). Relevant cases are selected by the researcher and then each case is coded to extract relevant information. The codings can then be aggregated, analysed and reported. Larrson [14] presents the case survey methodology for use in management research, together with a collection of limitations and benefits. He compares criteria from eight existing case surveys to demonstrate how the methodology works in practice. The criteria (Table 1) provide an indication of the steps that are typically found in case surveys.

Table 1: Criteria used by Larsson in comparing existing case surveys [14]

#	Criteria
1	Research questions
2	Case selection criteria
3	Number of cases
4	Coding scheme (number of variables, typical scale, number of points, confidence scoring, research design)
5	Number of rates per case
6	Author participation
7	Inter-rater reliability
8	Discrepancy resolution
9	Coding validity tests
10	Impact of case characteristics (case collection, research design, publication status, time period)
11	Analysis of data
12	Reporting study (coding scheme, sample)

In software engineering research, we are aware of only a few examples of case surveys being used. Peterson *et al.* [20] use the case survey method to investigate the choice of software components in a software system. Klotins [13] applies the case survey methodology to analyse software engineering practices in startups, although Klotins' application of the case survey is unclear. It appears that a questionnaire was developed and distributed, with each response being treated as a 'case' for analysis. In information systems research Jurisch *et al.* [10] present the case survey method as a way to generalise results from case studies, providing a typical case survey structure along with benefits and limitations. More recently, Peterson has published guidelines for conducting case surveys in software engineering research [19] and case surveys appear in developed empirical standards [28].

[1] e.g. https://github.com/kilimchoi/engineering-blogs
[2] http://www.joelonsoftware.com
[3] https://chase-seibert.github.io/blog/2014/08/01/why-blogging.html
[4] https://stackoverflow.blog/2010/09/29/good-subjective-bad-subjective/

3.2 Incorporating grey literature into case surveys

Although there are currently no guidelines for conducting primary studies which use blog-like documents as a data source, blog-like documents can be used in both primary and secondary studies. Rather than investigating blog-like documents in terms of reviewing literature (such as in GLRs and MLRs [9, 30]), blog-like documents can be used as data for analysis. Each document can be viewed as a case which comprises multiple potential units for analysis (for example, the textual content relating to a topic relevant to the research being undertaken).

Parnin, Treude, Storey and Aniche [1, 17, 18] have conducted primary studies that use blog-like documents as units to analyse. They have not treated blog-like documents as literature, rather they have explicitly treated blog-like documents as cases for analysis. Pagano and Maalej [16] analyse the blog documents written by practitioners involved in four open source projects. The study explicitly treats the blog documents as data. Finally, Rainer [22] analyses a single blog article for the reasoning and evidence that it presents. This can be interpreted as the analysis of a single case.

Deciding on whether to conduct a secondary study (literature review) or primary study (e.g. case survey) depends on the context of the study being undertaken. Rainer and Williams [27] provide a comparison of the two approaches.

3.3 Positioning the proposed methodology relative to other proposals

The large quantity of blog-like documents available on the web means that we require a methodology which combines the depth-focus of case studies with an inclusive breadth-search and selection process. The case survey methodology combines the case study method and survey method to achieve this combination. In a previous paper [25], we have presented a version of Jurisch *et al.*'s [10] case survey method (repeated here in Table 2). Given that the specific implementation of the method depends on the particular study being conducted, Table 2 instead provides examples of the steps and objectives typical of the methodology.

The aim of the methodology is to identify blog-like documents which are relevant to the research being undertaken, and of sufficient quality so that they might be useful to the researcher. The methodology should identify such documents for the researcher to then encode, aggregate and analyse in order to investigate a pre-determined research question. To identify relevant results, we use traditional search engines since this reflects the way in which researchers currently search for and identify documents. Existing literature review processes apply a series of search selection/rejection criteria for identifying articles for review (for example, search terms to indicate topic and time periods to indicate timeliness). When looking at incorporating grey literature, many more results exist then searching for research alone and search engines return a lot of irrelevant and off-topic results. By adding quality assessment as an additional search dimension, our aim is for researchers to more easily identify the articles which are of value to them, to then conduct their case survey analysis.

Our methodology provides two areas for quality assessment to take place: 1) during the searching (using keyword-based search criteria) and 2) during the analysis of the documents text after downloading the search results. The guidelines for Systematic Reviews [5, 21] and for MLRs [9] provide some guidance in identifying quality criteria. However, given that quality assessment is subjective, and dependent on the study being undertaken, there is a need for a more formal model to be developed which shows how software engineering researchers assess quality of documents. This model and our subsequent quality criteria are presented in Section 4.

4 A MODEL FOR CREDIBILITY ASSESSMENT

The methodology presented in the previous section is predicated on having some definition of what researchers deem to be 'high-quality.' We define 'high-quality' here in terms of the relevance of the documents to the research being undertaken and their perceived credibility. We use the term 'perceived' as we are looking to assess the credibility of the document without prior knowledge of how factually accurate the content presented is (actual credibility).

Automatically checking a documents' relevance is difficult because it requires knowledge of the context of the study being undertaken and the type of content that the researcher is looking for. Therefore, we utilise traditional search engines because this is a mechanism which is currently familiar to researchers searching for grey literature. We then apply our search heuristics [26] as a first-pass to identify documents with high-level credibility.

Automatic credibility assessment is also a difficult task as credibility is subjective. In a previous literature review of credibility assessment [36], ten of the thirteen publications analysed mentioned subjectivity as a challenge of credibility assessment. Previous studies have addressed the issue of subjectivity by looking at credibility assessment for a particular user group. For example, Tan and Chang [33] look at students aged between 18 and 25 years old and with experience reading travel blogs, and Menchen-Trevino and Hargittai [15] look at college students from two Mid-western US universities. We are aware of no previous study which investigates how software engineering researchers assess credibility, and therefore we develop our own model of credibility assessment for this user group.

Our model of credibility assessment is based on designing a set of conceptual credibility criteria which are relevant to software engineering researchers. The development of our credibility model was conducted in two phases. First, a literature review [36] was conducted to gather a set of credibility criteria that had been used in assessing online media in previous research. The literature review was built on by a survey of software engineering researchers, which was used to refine and validate these credibility criteria [39]. Respondents were asked to score the importance of nine criteria on a 7-point Likert scale (Table 3). Each appropriate quantitative question in the survey is accompanied with an option to provide open-ended comments. Respondents often suggested other criteria in their comments and these were also included in the final credibility model. So, our credibility assessment model is based on the credibility criteria from the survey results. Studies which use our suggested adaptation of the case survey methodology may find it useful to select a variety of criteria from our model, depending on the context of the study being undertaken.

Table 2: An example mapping of our version of the methodology to that in Jurisch *et al.* [10]

Step	Objective	Explanations and examples
1	Research questions	An example research question (based on [34]) is, *Do software testing practitioners cite software research in their online articles?*
2	Case study sourcing, composition	The respective case (or unit of analysis) for the example research question could be: an online article that satisfies the relevant search *and* post-search quality criteria.
	1. Criteria for case selection	Criteria for case selection would comprise: criteria for determining relevance of blog articles (e.g. topic words), and criteria for determining quality of article e.g. argument indicators [35]
	2. Construction of search terms	Create a set of keyword-based search terms for each case selection criteria, and structure those keywords search terms according to our structured method [26]
	3. Execution of searches	Execute the automated searches using the search tool
	4. Downloading of search results	Download the web pages of search results using the crawler
3	Survey development, comprising	
	1. Identification of variables	Which variables are important given the context of the study being undertaken?
	2. Operationalisation of the variables	Identify measures already offered through the analyser, add additional measures; or used an alternative or complementary analyser
4.	Data collection	Execute the analyser/s.
5.	Data analysis	Interpret the results of the analysis.
6.	Report results	Write up and publish results.

Table 3: Statistics and rankings for credibility criteria (originally published in [38]

	Statistics					Rankings		
	Me	Mo	Md	SD	%(6)	Md	Me	%(6)
Reason	5.1	6	5	1.0	38.6	1	1	1
RED	4.9	6	5	1.0	31.8	1	2	2
CoW	4.6	5	5	1.2	29.5	1	3	3
RM	4.6	4	5	1.3	27.3	1	3	4
PExp	4.5	5	5	1.2	20.5	1	4	5
URL-R	4.3	5	5	1.5	13.6	1	5	6
URL-P	4.0	5	4	1.4	9.1	2	6	7
Beliefs	3.1	3	3	1.9	6.8	3	7	8
IofO	3.0	2	3	1.8	6.8	3	8	8

Me: Mean; Mo: Mode; Md: Median; SD: Standard deviation; %(6): Percentage of respondents rating the criterion as *6 Extremely important.* Reason: Reasoning; RED: Reports empirical data; CoW: Clarity of writing; RM: Reports data collection method; PExp: Professional Experience; URL-R: Links to research source(s); URL-P: Links to practitioner source(s); Beliefs: Prior beliefs; IofO: Influence of others

Our model provides a holistic view of the interactions that take place throughout the entire blog-like document life-cycle and how the researcher may interact with such documents when assessing them for use as evidence in a primary study (such as our adaptation of the case survey methodology). The model is based on Berlo's Source, Message, Channel, Receiver (SMCR) model of communication [3], showing that credibility assessment takes place at each stage of the model. The difference between Berlo's model and ours is that we distinguish between the individual reader (e.g. the researcher assessing the document) and all other readers (i.e. the audience) who provide the document with its reputation. Thematic analysis conducted on the survey data shows that researchers use the reputation of the practitioner as a criterion for assessing credibility.

The models interactions take place as follows; practitioners experience some phenomenon and form opinions which are moulded by their prior beliefs. These opinions are developed and written up as a report which the practitioner then publishes on the web via a specific channel (e.g. a personal blog). The researcher consumes this report and assesses its perceived credibility and relevance to the particular domain/problem in which they are studying. Reports of sufficient relevance and credibility may be considered as evidence in the researcher's study using the case survey method.

Our final list of credibility criteria for each stage of Berlo's model of communication can be found in the Table 4. Existing quality assessment guidelines, such as Garousi *et al.*'s [9] guidelines for conducting MLRs and Kitchenham *et al.*'s [11] guidelines for conducting SLRs in software engineering, are often made up of a series of questions in order to assess multiple criteria which are also present in our model (e.g. reputation, expertise, clear aims and methodology, references, stated threats & limitations, balanced, conflicts & bias, presentation of data). Our model both aligns with, and extends these quality checklists.

To complement the methodology and credibility model, we are developing and have released versions of three Python libraries. In working towards automating as much of the methodology as possible, our aim is to minimize the time and effort required of the researcher. Each tool is preceded with the acronym `COAST` (Credible Online Article Search Tool). The following describes the three tools.

`COAST_CRAWL`. Crawls web pages from the page's domain and store results in a database[5].

[5] https://github.com/serenpa/coast_crawl

Table 4: The credibility criteria relevant to the source, message, channel, and receiver

Criteria	Description
Source (i.e. the author of the document)	
Prior beliefs	The prior beliefs of the author affect their opinions. The prior beliefs are made up of previous experience, the influence of others (e.g. through storytelling), self reflection and the authors personal values.
Motivation	The motivation the author has for writing the article.
Affiliation(s)	Is the author biased by an affiliation/sponsor?
Expertise	What is the expertise of the author? Are they qualified in the area of the article being written?
Message (i.e. the textual content within the document)	
Opinion	The opinion(s) being portrayed.
Reasoning & Explanation	The reasoning presented on which opinions are based.
Experience	The experiences reported as evidence for reasoning.
Data	Any data reported as evidence for reasoning.
Method	The reporting of the method in which any data has been collected.
Citations to research	Any citations to research sources used as evidence.
Citations to practitioners	Any citations to other practitioner sources used as evidence.
References	Any formal references used as evidence.
Threats	The declaration of any threats that may affect/alter what the document says.
Declaration of conflicts	The declaration of any conflicts of interest/bias's that the author of the document may have e.g. sponsorship.
Timestamp	Does the document contain a timestamp of when it was published and last updated?
Comments/ discussion	Does the document allow for comments and discussion? Is there evidence of the author engaging in discussion?
Updates	Has the document been updated since being initially published? Is there evidence of the author being willing to fix mistakes and update the ideas portrayed as time progresses?
Meta	Is the writing clear? Is the document of sufficient detail? Is the writing honest/balanced/fair?
Channel (i.e. the medium in which the document is published)	
Age of the site	How old is the site? Is it updated regularly?
Site contains an author bio	Does the site contain a description of the author and their expertise?
Sponsored/ affiliations	Are sponsorship's and affiliations declared? Do they bias the content?
Overall themes and topic of the site	What are the overall topics that are regularly discussed?
Perceived reputation of the site	Does the site appear to have a good reputation (e.g. large following/reputation for being a good source of knowledge)?
Receiver (i.e. the audience of the document)	
Prior beliefs	The prior beliefs of the reader affect their perceptions and the opinions that they form. As with the author, the prior beliefs are made up of previous experience, the influence of others, self reflection and the authors personal values.
Relevance	Not a credibility criteria as such, but a document must be relevant to the topic being studied in order to be useful to the researcher.

COAST_SEARCH. Uses the Google Custom Search API[6] to search for results in the same way that researchers currently find grey literature. To account for stochastic variation in results, COAST_SEARCH can be scheduled to run multiple times over multiple days and then de-duplicate results. COAST_SEARCH also allows researchers to automatically apply our search heuristics [26][7].

COAST_CORE. Takes an article as input, and develops a series of credibility metrics. These metrics are based on a subset of the credibility criteria detailed in the previous section. We plan to continue development on COAST_CORE to cover more of our credibility model[8].

5 APPLYING THE METHODOLOGY TO A GLR ON HIGH PERFORMANCE TEAMS (HPT)

In this section, we demonstrate the methodology by applying it to a grey literature review on High Performance Teams (HPTs). The first author took the role of the operator of the software tools. The second author took the role of the grey literature reviewer. In other words, the second author was treated as a client who has the aim of conducting a grey literature review (GLR) to find out

[6] https://developers.google.com/custom-search/v1/overview

[7] https://github.com/serenpa/coast_search

[8] https://github.com/serenpa/coast_core

what practitioners were writing about high performance teams in industry. These two roles were independent of each other and the GLR was conducted prior to agreeing to co-publish the GLR in the current paper. The research question being investigated was: **RQ1** What are the team capabilities, behaviours, attitudes, characteristics and values that distinguish a high-performance team from a low-performance team? From the research question, we identify four search dimensions: team(s), team type, factors and domain. Through discussion with the client, we developed a set of keywords for each dimension that can be used for searching (Table 5).

Table 5: The keywords decided upon for each dimension

Team	Team Type	Factors	Domain
Team	High performance	Capabilities	Software engineering
	Low performance	Behaviours	Practitioners
	Work	Attitudes	Development
	Performance	Characteristics	Coding
	Productivity	Values	

We search using the heuristics suggested by Rainer and Williams [26]. The client specified that given the context of the study, it is not necessary to search for all combinations of dimension. The client did not want to include studies that do not relate to teams (i.e. NOT Team) but does want to study high performing teams in other domains. This means that only two combinations of queries are run. These are:

(1) Team AND Team Type AND Factors AND Domain
(2) Team AND Team Type AND Factors AND NOT Domain

The queries were run over a period of seven days starting on the 10th of December 2018. We collected 100 results per query per day. One of the days failed to run at all and another only partially ran, meaning that overall, we successfully retrieved 6 days of results for the first query and 5 days of results for the second. Overall, 289 unique URLs were identified through searching. The results contained a mixture of grey literature and research. As we were only interested in the grey literature, we removed all research links using COAST_CORE. This reduced our number of articles for analysis to 268 (however, some research sources were missed). We successfully extracted the article text from 253 articles. The remaining 15 failed to extract due to reasons such as 404 errors or the URL containing a PDF which could not be extracted. Finally, we removed all articles whose word count fell below the number of the first quartile. This was done to remove non-article results such as short update posts. Looking at a more rigourous method for identifying articles from non-articles is one future direction where supervised machine learning techniques may help. Removing these low word count articles left us with 190 articles for analysis.

We considered four credibility criteria for assessing the quality of the 190 results. Each criterion was measured using one or more metrics. Where appropriate, we normalised the metrics against the word count so that we could order the articles proportionally by each criterion. COAST_CORE was used for all analysis. The metrics used to measure each criteria are listed below, with descriptive statistics for each metric presented in the online appendix[9]:

Reasoning. We searched for the frequency of the 86 reasoning markers developed in [35]. For each article, we counted the absolute number of markers present and divided by the word count so that articles can be compared by their proportion of reasoning.

Experience. We used six metrics to assess experience: we counted the frequency of 9 experience markers (I, me, we, us, my, experience, experiences, experienced, our); temporal events using a modified version of the TIMEX library[10]; the presence of verbs in their base form, past tense verbs and gerund/present participle verbs; bigrams where the first word is 'I' and the second word is a past tense or past participle verb. We refer to these as 'iverbs'; named entities using the NLTK named entity chunker[11]; and the presence of pronouns.

Citations. We counted the number of citations (hyperlinks) that are made to sources outside of the domain of the articles URL; the number of these external citations that we were able to classify using our classification scheme developed in [38]; and the number of these classified citations which have been specifically classified as being made to research sources.

Clarity of writing. We assessed the writing using three metrics: readability (using the Flesch reading ease score), grammar (the total number of grammar issues found), and sentiment (we used the TextBlob[12] library to measure the articles polarity and subjectivity).

Under most circumstances, the next stage of the process would be that the results are exported and presented to the researcher under taking the study. The reason for this is that aggregating and ranking the metrics is dependent on the context of the study being undertaken. Therefore, there can be no general method for combining results. However, we continued the analysis here for demonstrative purposes.

We ordered and assigned a rank to each article based on its normalised score. This was done for every individual metric except for the Flesch reading ease score, the subjectivity score, and the polarity score as these are scores assigned to the text as a whole and therefore already normalised. To get an overall rank for each credibility criteria, we took the sum of each metric and divided by the number of metrics measured for each criteria. We then summed the ranks for each criterion to give an overall rank score. Ranking this way is problematic as it applies equal weight to each of the metrics (exploring other methods of ranking is left for future research). The top five articles and their ranks are presented in Table 6. Each of the articles, with the exception of the fifth which is a false negative from filtering our research results, is an instance of practitioners writing about high performance teams in industry and sharing their experiences.

As an additional layer of validation, the client was asked to look at the results and select which they would consider for including in a grey literature review. In other words, the client was asked to replicate the inclusion/exclusion review phase on a subset of the data. This process consisted of three rounds of annotation, each

[9] https://github.com/ash-williams/CS4GL_evaluation
[10] https://github.com/nltk/nltk_contrib/blob/master/nltk_contrib/timex.py
[11] https://www.nltk.org/
[12] https://textblob.readthedocs.io/en/dev/

Table 6: Top five ranked articles.

URL	Reasoning rank	Experience rank	Citations rank	Clarity of writing rank	Sum of ranks
https://www.huffingtonpost.com/nezha-alaoui-/the-fundamental-values-fo_b_13149368.html	29	23.5	6	98.8	157.3
https://www.siteground.com/blog/team-festival-embodies-values/	53	48.5	42	44.8	188.3
https://life.taxjar.com/core-values-remote-team/	47	36.83	24	87	194.83
https://www.tlnt.com/the-15-characteristics-of-a-highly-productive-team/	33	53.33	11	100.2	197.53
https://papers.ssrn.com/abstract=932948	24	86.83	30.33	64.8	205.97

round contained 30 articles to be reviewed. Articles in the first round were selected randomly, and then the following rounds were selected by trying to predict a balanced set of which articles would be included/excluded. This was done in an attempt to learn more about the internal assessment that occurs when deciding what to include and what to reject.

Overall, the client selected 24 out of the 90, and rejected 60. The remaining 6 were labelled as 'maybe' or 'n/a.' The descriptive statistics for the 24 selected articles can be found in the online appendix[13]. There is little difference between the metrics of the selected articles when compared with the metrics of the entire dataset. This implies that the client adopted other criteria/metrics when classifying (the client also provided short justifications for each classification). However, there is an indication from the descriptive statistics that the client favours documents with more verbs (an indication of the author talking about their own experience), and articles that contain fewer grammar and spelling mistakes. A more detailed evaluation is left for future research.

6 THREATS, FUTURE WORK & CONCLUSIONS

6.1 Threats to validity

Each threat provides an opportunity for future work. The credibility model has been developed based on the findings of a previous literature review and survey. The review was not conducted systematically and only 13 papers were selected for analysis. A broader, systematic review may yield new important criteria for the model to consider. The survey was intended to verify the review and find new criteria. Although the response rate was good, the overall number of responses in comparison to the community of software engineering researchers is relatively low. Therefore, work is needed to ensure the model generalises. In addition, it is difficult to formally evaluate the methodology in practice as we lack the corpora necessary to do so. Instead, we have relied on a series of case surveys and have been iterating development of the methodology as the research has matured. One future direction is to evaluate the methodology via an expert panel.

6.2 Future research

There are two key areas for future research: With regards to credibility assessment, an SLR would provide a mechanism to find the criteria and methods used by related research as a starting point. On completion of this review, results can be verified again through surveying a larger number of software engineering researchers and a more complete credibility model can be produced. With respect to development of the methodology, more criteria can be added to the model and alternative methods of assessing criteria and the overall credibility of a document can be investigated. We are currently investigating the use of meta-knowledge in a technique for inferring overall credibility.

6.3 Conclusions

In this paper, we present our work on developing a model for how software engineering researchers assess the quality of blog-like content. We then present a methodology for semi-automating such assessment using this model. We also demonstrate the use of the methodology in practice.

Given the breadth of different types of software engineering researcher, as well as the subjective nature of credibility assessment, this model is still not likely to apply to all situations. Instead, we encourage researchers to adapt the model, selecting the assessment criteria most relevant to their views and the context of the study being undertaken. Adopting credibility assessment into grey literature reviews requires a substantial amount of extra time and effort from the researcher undertaking the study. Therefore, it is important that we attempt to alleviate some of this through automation. Our methodology not only provides structure to the process of incorporating credibility assessment, but we have also developed a series of tools to aid with automation and adoption of the methodology.

Overall, the work presented here serves as a foundation for future research to build upon. The research has an abundance of potential future directions and we anticipate that our credibility model and the tools which we have developed will aid future work on grey literature quality assessment.

REFERENCES

[1] Maurício Aniche, Christoph Treude, Igor Steinmacher, Igor Wiese, Gustavo Pinto, Margaret-Anne Storey, and Marco Aurélio Gerosa. 2018. How modern news aggregators help development communities shape and share knowledge. In *2018 IEEE/ACM 40th International Conference on Software Engineering (ICSE)*. ACM, NY, USA, 499–510.

[2] Anton Barua, Stephen W Thomas, and Ahmed E Hassan. 2014. What are developers talking about? an analysis of topics and trends in stack overflow. *Empirical Software Engineering* 19, 3 (2014), 619–654.

[3] David K Berlo. 1960. *The Process of Communication: An Introduction to Thery and Practice.* Rinehart Press, NY, USA.

[4] Gargi Bougie, Jamie Starke, Margaret-Anne Storey, and Daniel M German. 2011. Towards understanding twitter use in software engineering: preliminary findings,

[13] https://github.com/ash-williams/CS4GL_evaluation

ongoing challenges and future questions. In *Proceedings of the 2nd international workshop on Web 2.0 for software engineering*. ACM, NY, USA, 31–36.

[5] David Budgen and Pearl Brereton. 2006. Performing systematic literature reviews in software engineering. In *Proceedings of the 28th international conference on Software engineering*. ACM, NY, USA, 1051–1052.

[6] Bruno Cartaxo, Gustavo Pinto, and Sergio Soares. 2018. The Role of Rapid Reviews in Supporting Decision–Making in Software Engineering Practice.. In *Proceedings of the 22nd International Conference on Evaluation and Assessment in Software Engineering (EASE)*. ACM, NY, USA, 24–34.

[7] Michael Chau, Jennifer Xu, Jinwei Cao, Porsche Lam, and Boby Shiu. 2009. A blog mining framework. *It Professional* 11, 1 (2009), 36–41.

[8] Prem Devanbu, Thomas Zimmermann, and Christian Bird. 2016. Belief & evidence in empirical software engineering. In *Proceedings of the 38th international conference on software engineering*. ACM, NY, USA, 108–119.

[9] Vahid Garousi, Michael Felderer, and Mika V Mäntylä. 2019. Guidelines for including grey literature and conducting multivocal literature reviews in software engineering. *Information and Software Technology* 106 (2019), 101–121.

[10] Marlen C Jurisch, Petra Wolf, and Helmut Krcmar. 2013. Using the case survey method for synthesizing case study evidence in information systems research. In *Proceedings of the Nineteenth Americas Conference on Information Systems*. Citeseer, IL, USA, 1–8.

[11] B. Kitchenham and S Charters. 2007. Guidelines for performing Systematic Literature Reviews in Software Engineering.

[12] Barbara A Kitchenham, Tore Dyba, and Magne Jorgensen. 2004. Evidence-based software engineering. In *Proceedings of the 26th international conference on software engineering*. ACM, NY, USA, 273–281.

[13] Eriks Klotins. 2017. Using the case survey method to explore engineering practices in software start–ups. In *2017 IEEE/ACM 1st International Workshop on Software Engineeting for Startups (SoftStart)*. ACM, NY, USA, 24–26.

[14] Rikard Larsson. 1993. Case survey methodology: Quantitative analysis of patterns across case studies. *Academy of management Journal* 36, 6 (1993), 1515–1546.

[15] Ericka Menchen-Trevino and Eszter Hargittai. 2011. Young adults' credibility assessment of wikipedia. *Information, Communication & Society* 14, 1 (2011), 24–51.

[16] Dennis Pagano and Walid Maalej. 2011. How do developers blog?: an exploratory study. In *Proceedings of the 8th working conference on Mining software repositories*. ACM, NY, USA, 123–132.

[17] Chris Parnin and Christoph Treude. 2011. Measuring API documentation on the web. In *Proceedings of the 2nd international workshop on Web 2.0 for software engineering*. ACM, NY, USA, 25–30.

[18] Chris Parnin, Christoph Treude, and Margaret-Anne Storey. 2013. Blogging developer knowledge: Motivations, challenges, and future directions. In *2013 21st International Conference on Program Comprehension (ICPC)*. IEEE, NY, USA, 211–214.

[19] Kai Petersen. 2020. Guidelines for Case Survey Research in Software Engineering. In *Contemporary Empirical Methods in Software Engineering*. Springer, Berlin, Heidelberg, 63–92.

[20] Kai Petersen, Deepika Badampudi, Syed Muhammad Ali Shah, Krzysztof Wnuk, Tony Gorschek, Efi Papatheocharous, Jakob Axelsson, Severine Sentilles, Ivica Crnkovic, and Antonio Cicchetti. 2018. Choosing Component Origins for Software Intensive Systems: In-House, COTS, OSS or Outsourcing?—A Case Survey. *IEEE Transactions on Software Engineering* 44, 3 (2018), 237–261.

[21] Kai Petersen, Sairam Vakkalanka, and Ludwik Kuzniarz. 2015. Guidelines for conducting systematic mapping studies in software engineering: An update. *Information and Software Technology* 64 (2015), 1–18.

[22] Austen Rainer. 2017. Using argumentation theory to analyse software practitioners' defeasible evidence, inference and belief. *Information and Software Technology* 87 (2017), 62–80.

[23] Austen Rainer, Tracy Hall, and Nathan Baddoo. 2003. Persuading developers to" buy into" software process improvement: a local opinion and empirical evidence. In *Empirical Software Engineering, 2003. ISESE 2003. Proceedings. 2003 International Symposium on*. IEEE, NY, USA, 326–335.

[24] Austen Rainer, Dorota Jagielska, and Tracy Hall. 2005. Software Engineering Practice versus Evidence-Based Software Engineering Research. In *Proceedings of the 2005 Workshop on Realising Evidence-Based Software Engineering* (St. Louis, Missouri) *(REBSE '05)*. Association for Computing Machinery, NY, USA, 1–5.

[25] Austen Rainer and Ashley Williams. 2018. Using blog articles in software engineering research: benefits, challenges and case–survey method. In *2018 25th Australasian Software Engineering Conference (ASWEC)*. IEEE, Adelaide, Australia, 201–209.

[26] Austen Rainer and Ashley Williams. 2019. Heuristics for improving the rigour and relevance of grey literature searches for software engineering research. *Information and Software Technology* 106 (2019), 231–233.

[27] Austen Rainer and Ashley Williams. 2019. Using blog–like documents to investigate software practice: Benefits, challenges, and research directions. *Journal of Software: Evolution and Process* 31, 11 (2019), e2197.

[28] Paul Ralph. 2021. Acm sigsoft empirical standards released. *ACM SIGSOFT Software Engineering Notes* 46, 1 (2021), 19–19.

[29] Christoffer Rosen and Emad Shihab. 2016. What are mobile developers asking about? a large scale study using stack overflow. *Empirical Software Engineering* 21, 3 (2016), 1192–1223.

[30] Jacopo Soldani, Damian Andrew Tamburri, and Willem-Jan Van Den Heuvel. 2018. The pains and gains of microservices: A Systematic grey literature review. *Journal of Systems and Software* 146 (2018), 215–232.

[31] Margaret-Anne Storey, Leif Singer, Brendan Cleary, Fernando Figueira Filho, and Alexey Zagalsky. 2014. The (r) evolution of social media in software engineering. In *Proceedings of the on Future of Software Engineering*. ACM, NY, USA, 100–116.

[32] Margaret-Anne Storey, Christoph Treude, Arie van Deursen, and Li-Te Cheng. 2010. The impact of social media on software engineering practices and tools. In *Proceedings of the FSE/SDP workshop on Future of software engineering research*. ACM, NY, USA, 359–364.

[33] Wee-Kheng Tan and Yun-Ghang Chang. 2016. Place Familiarity and Attachment: Moderators of The Relationship Between Readers' Credibility Assessment of A Travel Blog and Review Acceptance. *Journal of Travel & Tourism Marketing* 33, 4 (2016), 453–470.

[34] Ashley Williams. 2018. Do software engineering practitioners cite research on software testing in their online articles?: A preliminary survey.. In *Proceedings of the 22nd International Conference on Evaluation and Assessment in Software Engineering 2018*. ACM, Christchurch, New Zealand, 151–156.

[35] Ashley Williams. 2018. Using reasoning markers to select the more rigorous software practitioners' online content when searching for grey literature. In *Proceedings of the 22nd International Conference on Evaluation and Assessment in Software Engineering 2018*. ACM, Christchurch, New Zealand, 46–56.

[36] A. Williams and A. Rainer. 2017. The analysis and synthesis of previous work on credibility assessment in online media: technical report. https://www.researchgate.net/publication/324765770

[37] Ashley Williams and Austen Rainer. 2017. Toward the use of blog articles as a source of evidence for software engineering research. In *Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering*. ACM, Karlskrona, Sweden, 280–285.

[38] Ashley Williams and Austen Rainer. 2019. Do software engineering practitioners cite software testing research in their online articles?: A larger scale replication. In *Proceedings of the Evaluation and Assessment on Software Engineering*. ACM, Copenhagen, Denmark, 292–297.

[39] Ashley Williams and Austen Rainer. 2019. How do empirical software engineering researchers assess the credibility of practitioner–generated blog posts?. In *Proceedings of the Evaluation and Assessment on Software Engineering*. ACM, Copenhagen, Denmark, 211–220.

[40] Claes Wohlin. 2013. An evidence profile for software engineering research and practice. In *Perspectives on the Future of Software Engineering*. Springer, Berlin, Heidelberg, 145–157.