Fall 2023

Towards Agile Academia: An Approach to Scientific Paper Writing Inspired by Software Engineering

Tyler Procko
Embry-Riddle Aeronautical University, prockot@my.erau.edu

TOWARDS AGILE ACADEMIA: AN APPROACH TO SCIENTIFIC PAPER WRITING INSPIRED BY SOFTWARE ENGINEERING

By

Tyler Thomas Procko

A Thesis Submitted to the Faculty of Embry-Riddle Aeronautical

University in Partial Fulfillment of the Requirements for the

Degree of Master of Science in Software Engineering

Embry-Riddle Aeronautical University

Daytona Beach, Florida

September – December 2023

TOWARDS AGILE ACADEMIA: AN APPROACH TO SCIENTIFIC PAPER WRITING INSPIRED BY SOFTWARE ENGINEERING

By

Tyler Thomas Procko

This Thesis was prepared under the direction of the candidate's Thesis Committee Chair, Dr. Omar Ochoa, Department of Electrical Engineering and Computer Science.

Acknowledgements

It occurs to me that tradition places a requirement, as it were, upon the author of any graduate research artifact, to acknowledge and give thanks to those individuals who contributed in some measure to the development of said artifact. While the present paper is not, indeed, my ultimate provision as a graduate student, i.e., my dissertation thesis, it is nonetheless proper to acknowledge certain important associates of mine who have affected me positively during the course of the present work. I am indebted, to inestimable degrees, and in various capacities, personal and professional, to my long-time advisor and close friend, Dr. Omar Ochoa, who has, above all others, guided me through muddy research with diligence and patience, and faithfully represented to me what it is to be a software engineer. To my mentor, Dr. Nicholas Del Rio, I owe much unreserved gratitude, insofar as he alone proffered to me a field of research that has shaped no insignificant part of my own work and personal belief; that is, the philosophical discipline of Ontology, and the Semantic Web. I would be woefully remiss if I were to forget my closest friend and colleague, Timothy Elvira, who has sat by my side, published alongside my name and become a source of humor in my presence since our undergraduate years. I am grateful as well for the camaraderie shared with my colleague, Anton Kiselev, whose own graduate research project I was honored to play some role in, and whose progress in this regard has informed the present work. To the various associates of Omar's research group, my appreciation is due. I must also thank the members of my family, particularly my mother and father, for supporting me in my various endeavours, and always remaining steadfast in a world of vacillation.

Science is built up of facts, as a house is built of stones; but an accumulation of facts is no more a science than a heap of stones is a house.

Jules Henri Poincaré, 1905

ABSTRACT

The construction of scientific papers is performed in service of the greater scientific community. This iterative process is, in effect, an academic economy, where all members benefit from well-written papers. However, many published scientific papers are poorly written; they often lack sufficient detail to allow replication, there is improper usage of citations or a lack of regard to relevant work, reporting is vague or without linked empirical data to allow verification, figures do not correspond to text or are non-sensical, literary elements, e.g., bulleted lists, are used ineffectively, formatting renders certain sections unreadable, and grammatical errors abound. The issues of paper quality are widespread and of varying concern. Similarly, the development of software systems is rife with many processual issues, from high-level architectural flaws to small developer errors, e.g., setting a Boolean value to true instead of false, which can be disastrous in large systems. As an answer to these longstanding concerns, software development methods have emerged over decades, most notably, the Waterfall and Agile approaches. These methods have established software engineering as a professional discipline backed by rigorous, empirical evaluation on many systems. A scientific paper is, conceptually, a system to be developed, much like a software system: it has a name, particular sections codified for different purposes, e.g., as the abstract summarizes and the conclusion concludes, it has an author or authors, it goes through several iterations of refinement, it may reference outside systems and it is eventually released to the public, and possibly maintained in future versions. It is posited that, due to the relatively small nature of most scientific papers (4-20 pages), the Agile method of software development can be used to produce more reliable scientific papers, in a more efficient manner and with better availability to readers, by employing the principles of open-source software, and a version control system, e.g., Git. Agile methods consistently provide deliverables of higher quality; this work intends to demonstrate that Agile can be adapted to streamline the scientific writing process and improve publication quality.

Table of Contents

List of Figures

List of Tables

1. Introduction

The quality of scientific publications has long been a topic of debate. As witness, that period of decades in the mid-20th century when epistemologists, historians and sociologists variously contested the issue of mismatch between what scientists actually do (typically following a hypothetico-deductive model) and how they reported it in their publications (typically misrepresented as a perfectly inductive process) (Schickore, 2008; Holmes, 1987). Even the "systematicism" of systematic literature reviews has been subject to debate (Boell & Cecez-Kecmanovic, 2016). Fine details of scientific papers, such as grammatical correctness, are often poorly executed, especially in papers published by non-native English writers (Madkur, 2013; Carrío-Pastor & Mestre-Mestre, 2014). Any method of scientific writing employed by some author(s) is subject to their own experience, thinking processes and various extraneous factors, such as funding sources and incentivizing outlets. A proper, structured method for the development of a system, such as a scientific paper, would provide the writer with a comprehensive, replicable means of factually reporting their scientific endeavours in an efficient manner.

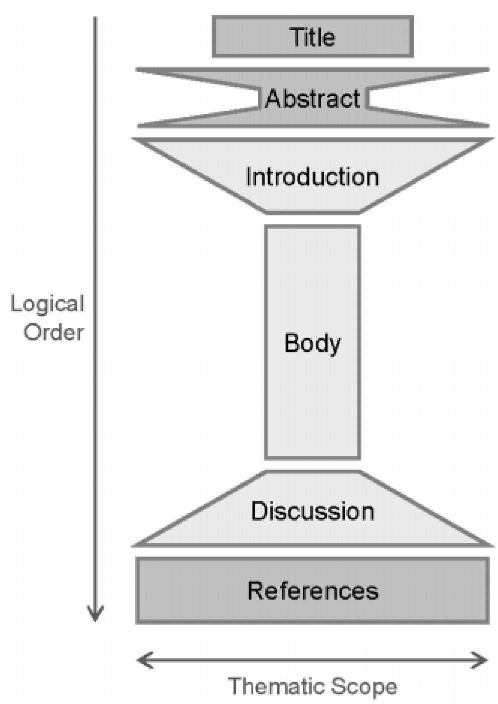

Fig. 1 The hourglass (or king) model of scientific writing. Corresponds loosely with the IMRaD model. From (Derntl, 2014).

1.1. Motivation

Although there are extant a number of proposed guides for the writing of scientific papers, e.g., Introduction, Methods, Results and Discussion (IMRaD) (Sollaci & Pereira, 2004) and the hourglass model (Derntl, 2014), such general outlines provide only frameworks for scientific writing, and do not consider the more granular, iterative, collaborative facets of scientific writing. While such frameworks may act as points of origin in seeking improved scientific publication, it is necessary to consider other project development methods that have greater empirical validity and explicitly measured success in project delivery. One such method is Agile development, from the domain of software engineering, which is ideal for the rapid development of small projects with small teams (Al-Saqqa, Sawalha, & AbdelNabi, 2020). As most scientific papers are rather short, e.g., most conference publications are between 4 and 8 pages, and most scientific and research teams have between 2 and 8 members, the Agile development method is quite amenable, conceptually, to scientific writing. The motivation behind the present paper is to ascertain the validity of this conception.

1.2. Main Objective

The purpose of this project is to establish a framework for the writing of scientific papers, by adhering to the Agile method. Every phase of the traditional Software Development Lifecycle (SDLC), from requirements elicitation to maintenance, will be mapped to the canonical phases of scientific writing, and the traditional constructs of Agile development employed to produce better papers more efficiently. Some of the aims of this project are to improve or otherwise engender:

1. Knowledge: as of yet, there is a very small quantity of literature extant on the use of Agile methods for writing scientific papers; from the preliminary literature review conducted for this proposal, there are fewer than five papers directly addressing the use of an Agile

method in scientific writing (Cruz, et al., 2021; Ramos, Ramos, Viana, Silva, & de Oliveira, 2016; Svoboda, 2021), with the closest subsuming topic being Scrum for research projects, but no focus on the writing and publication of papers themselves; there are general "techniques" for writing scientific papers, e.g., the IMRaD structure (Sollaci & Pereira, 2004) and the hourglass model (Derntl, 2014), but these are extremely simplistic outlining schemes, and do not provide writers with an iterative development lifecycle, like Agile

2. Performance: the turnaround time for scientific papers can not only be reduced, but normalized, using Agile, resulting in more consistent scientific paper development and rapid dissemination
3. Quality: the quality of scientific papers can be improved by adhering to Agile principles for development
4. Compliance: scientific papers are written without a defined standard for doing so; ensuring compliance with well-established Agile principles can lend to increased reliability and adoption, as well as the guarantee of a more standardized paper format, which is amenable to widespread understanding
5. Productivity: the productivity of scientific writers can be improved by adhering to Agile principles
6. Risk Management: by utilizing the well-established principles of Agile engineering, it is possible to reduce the risks inherent in scientific publication, e.g., reporting incorrect data, not providing enough information for the paper to be replicable, etc.

1.3. Research Questions

The following questions guide this research:

RQ1: How feasible is the adaptation of the Agile method to the construction, refinement and dissemination of scientific papers?

RQ2: How does implementing Agile influence the quality of scientific papers when compared to traditional paper development processes?

RQ3: How does employing the Agile method affect the efficiency of the paper construction process and the collaboration between co-authors and reviewers?

1.4. Methods Preview

In seeking an answer to these research questions, a two-fold approach is taken:

1. **Systematic Literature Review:** a SLR of scientific writing approaches/methods will be conducted
2. **Survey:** a survey of fourteen experienced engineers on their own methods of scientific writing will be conducted

The insights gained from these efforts will result in a systematic mapping between the canonical scientific paper writing processes and Agile SDLC phase techniques. First, however, it is necessary to consider relevant background knowledge, e.g., that on the SDLC, Agile development and Agile use outside of software engineering.

2. Background

The background needed for a proper understanding of the present work is necessarily brief, focusing on the history of scientific papers, software development and Agile methods.

2.1. A Brief History of The Canonical Scientific Paper

"Science is the orderly collection of scientific records–i.e., observations about the natural world made via well-defined procedures–and scientific records are archived in a standardized form, the scientific research paper" (Katz, 2009). The recorded history of the scientific paper, as it is regarded today, finds its beginning in the mid-17th century, in Europe, where educated gentlemen wrote detailed experimental reports in prosaic style, for the digestion of their peers (Sollaci & Pereira, 2004). The proliferation of scientific essays from Europe – London in particular – became more rigorous, with the establishment of the Royal Society, and the accepted "experimental essay" format of Robert Boyle (Moessner, 2009; Lareo Martín & Reyes, 2007). Scientific discourse over the decades progressed into the 20th century, leading to the unanimous adoption of the Introduction, Methods, Results and Discussion (IMRaD) scientific paper structure, driven primarily by the biomedical disciplines (Sollaci & Pereira, 2004). Today, the scientific paper is not so different from early experimental reports, although there are certainly more genres in the landscape of scientific literature, e.g., literature reviews, meta-analyses, perspective papers, and so on.

2.2. The Software Development Lifecycle

The software development lifecycle (SDLC) is an abstraction of typical system development progress particular to the profession of software engineering. Modeling of the SDLC began in the mid-1950s, with Benington's program production model, which was later adopted and modified by Royce in the 1970s, who produced the Cascade model (Benington, 1956; Royce, 1970). Ignoring models specific to the SDLC, like the Waterfall/Cascade model, the V-model, the Spiral model, and so forth, the general systems lifecycle for any product is composed of 6 stages, depending on the interpretation employed (Ruparelia, 2010; Sivess & Curtis, 1993) (see Figure 1):

1. Requirements elicitation

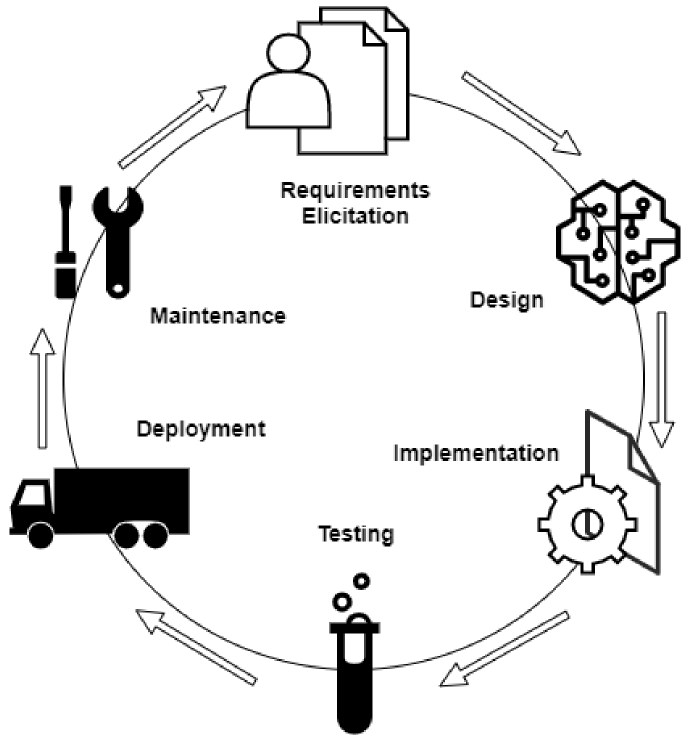

Fig. 2 The general software development lifecycle.

2. Design

3. Implementation

4. Testing

5. Deployment

6. Maintenance

These 6 stages are useful as a means of abstracting the generally encountered phases of software development, although some phases overlap and there are inevitable circumstances in which developers may return to previous steps to iterate once more. Starting in the late 1990s, such mutability led software practitioners to seek out more adaptive software development methods.

2.3. AGILE SOFTWARE DEVELOPMENT

Given the relatively formal, document-heavy, up-front deliberative methods that dominated software development work prior to the close of the 20th century, a growing number of software practitioners sought to employ more lightweight development methods, devoid of the encumbrance of rigid documentation and the resultant inflexibility to changing customer demands (Cohen, Lindvall, & Costa, 2003). Several methods emerged during the late 1980s-1990s, e.g., eXtreme programming (XP), SCRUM, lean software development, Crystal methods, and so on (Abrahamsson, Salo, Ronkainen, & Warsta, 2017; Dingsøyr, Nerur, Balijepally, & Moe, 2012). Akin to the universally adopted "heavyweight" models of software development used for the preceding 40+ years, what remained for the software practitioners of the new century was to reach a consensus on what lightweight software development was to be.

In February of 2001, seventeen software practitioners convened in a summit of like minds, ultimately producing the *Manifesto for Agile Software Development* (Beck, et al., 2001). Summarizing the core tenets of Agile can be achieved in four bullet points, where the Agile developer values:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

Additionally, the Agile manifesto prescribes 12 principles:

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity--the art of maximizing the amount of work not done--is essential.
11. The best architectures, requirements, and designs emerge from self-organizing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

This adaptability in the face of inevitable change is useful for small, rapidly-developed systems, such as scientific conference papers. For the purpose of the present paper, the Agile manifestation, SCRUM, will be used as the nominal Agile method, as it is deliberately generalized and capable of cross-domain use; i.e., in the present paper, SCRUM provides a suitable baseline method for Agile scientific paper writing. Other facets of the various Agile methods, e.g., pair programming from XP, will be considered as well.

2.3.1. SCRUM

Ken Schwaber, in the proposal paper for SCRUM, states: "The system development process is complicated and complex... Evolution favors those that operate with maximum exposure to environmental change and have optimised for flexible adaptation to change. Evolution deselects those who have insulated themselves from environmental change and have minimized chaos and complexity in their environment. An approach is needed that enables development teams to operate adaptively within a complex environment using imprecise processes. Complex system development occurs under rapidly changing circumstances. Producing orderly systems under chaotic circumstances requires maximum flexibility. The closer the development team operates to the edge of chaos, while still maintaining order, the more competitive and useful the resulting system will be" (Schwaber, Scrum development process, 1997). SCRUM consists of 4 primary "phases" that are iterative by nature, and that, taken together, engender further iterations, ultimately resulting in a final, refined deliverable (Deemer, Benefield, Larman, & Vodde, 2010). The "phases" of SCRUM are:

1. Architecture and product backlog: input from end-users, customers and other stakeholders results in a product backlog, i.e., a "to-do list"
2. Sprint planning meeting: the team selects how much from the product backlog can be done by the end of the next sprint
3. Sprint execution: for 1-4 weeks, the team works collaboratively to finish the objectives on the product backlog; daily meetings and updates are conducted to ensure communication between team members
4. Delivery, review and retrospective: the team delivers an incremented product (possibly a new, iterated-upon prototype or "draft" artifact) to stakeholders, and conducts the review and retrospective, where team effectiveness and product correctness, respectively, are discussed, to inform and improve the next iteration and future projects

SCRUM is a general project development framework suitable for any sort of collaborative work between members of small teams. As will be discussed later in the present paper, SCRUM is potentially quite applicable to the development of scientific papers.

2.3.2. Agile Outside of Software

Agile methods are not relegated to software development alone; in fact, Agile is used for many projects outside of the domain of software. In particular, SCRUM, as it is more generally focused on project management, and not software development, is commonly used as a cross-disciplinary project development method in various domains (Al-Saqqa, Sawalha, & AbdelNabi, 2020).

Some of the domains where Scrum and other Agile methods are commonly employed are: sales, education, manufacturing, marketing, healthcare, finance, human resources, geology and communication, where iterative sprints and sprint retrospectives are among the most important Agile techniques used (Oprins, Frijns, & Stettina, 2019). Notwithstanding, there have emerged several concerns regarding the transfer of Agile methods outside of software development, e.g., the difficulty in integrating Agile practices in domains using physical materials, as software is not precisely bound by physical constraints, and the slowness of Agile adoption in other domains as opposed to its rapid adoption in software development, which opens questions about organizational practices and institutional inertia (Niederman, Lechler, & Petit, 2018).

Agile methods have been mapped to pedagogical environments, to improve the cooperative learning of students (Stewart, DeCusatis, Kidder, Massi, & Anne, 2009). See Table I.

TABLE I Agile values mapped to education.

Agile Value	**Agile Education Value**
Individuals and interactions over process and tools	Students over traditional processes and tools
Working software over comprehensive documentation	Working projects over comprehensive documentation
Customer collaboration over contract negotiation	Student and instructor collaboration over rigid course syllabi
Responding to change over following a plan	Responding to feedback rather over following a plan

3. METHOD

To effectually delineate the applicability of the Agile method, SCRUM, for the writing of scientific papers, a proper literature review of scientific paper writing techniques, methods, frameworks and approaches, is necessary. The providence of such a review informs the later analysis and mapping of Agile software development techniques to scientific paper writing. Considering the landscape of scientific writing methods provides a concrete way of gauging how scientific papers are written, *in general*.

Additionally, following the literature review, a supplementary survey of software practitioners provides further insight into the potential for Agile methods as means of writing scientific papers.

3.1. SYSTEMATIC LITERATURE REVIEW

The protocol of this systematic literature review (SLR) is reported in adherence to, as closely as possible, the prescriptions set forth for SLRs in software engineering (Kitchenham & Charters, 2007), and the Preferred Reporting Items for Systematic Reviews and Meta-Analysis Protocols (PRISMA-P) statement (Moher, et al., 2015).

The SLR was conducted from October 2th – October 27th, 2023. These dates are given insofar as the author cannot assert absolute replicability beyond the given timeframe.

3.1.1. Search Strategy

Searches were conducted using a combination of traditional, regular expression-like searching through Google Scholar and other search engines, use of the reference snowballing technique (Wohlin, 2014), and inquisitive searching by prompts fed to the GPT-4 model of the ChatGPT interface. The use of GPT-4 for literature searching in the present work was limited precisely to listing potential literature to review through traditional Google Scholar searches. As large language models are prone to hallucination (Ji, et al., 2023), every article recommended by GPT-4 had to be individually verified as legitimate, e.g., as GPT-4 often imagined realistic conference titles, paper titles and author names, insofar as some provided "articles" appeared, at first exposure, to be real, but upon investigation in Google Scholar, such "articles" were forged. In any case, GPT-4 was useful in ensuring coverage of "fringe" literature by recommending related topics or domains of interest, potentiating searches for articles that would have otherwise been totally missed in traditional Google Scholar searches.

The search process was composed of two steps:

1. Search for other literature reviews: this step is essential, as the providence of a similarly interested literature review would greatly inform that of the present paper
2. Search for relevant articles: due to the rather qualitative nature of the scientific writing process, this search considers any type of article in which some unique technique, method, framework or approach to scientific writing is reported, e.g., primary/case studies, secondary studies, theoretical papers, methodological papers and academic books

3.1.1.1. Eligibility Criteria

The eligibility criteria for the present SLR are necessarily relaxed, as the location of articles on unique scientific paper writing methods is an exercise in patience; in initial searches, very few articles of relevance were located, and search phrases had to be modified (discussed later). Inclusion/exclusion criteria were as follows:

- Language: English only, or in a readily available format, e.g., PDF, that is able to be transcribed by uploading the whole document at once, e.g., with Google Scholar or ChatGPT
- Publication status: any publication status acceptable, e.g., under review, preprint, published, etc.
- Publication year: any publication year is acceptable
- Peer review status: articles published to a peer-reviewed outlet (e.g., a conference, journal, workshop, etc.), reviewed by a publishing agency (e.g., books), or hosted on a scientific outlet (e.g., SSRN or arXiv); for example, pseudo-scientific outlets, such as blogs and news sites, are excluded

3.1.1.2. Information Sources

Four types of information sources were utilized in the present SLR: scientific search engines and databases, open search engines, snowballed reference lists, and GPT-4. These information sources were used in descending order of the list presented below.

- Scientific search engines and databases
  - Google Scholar
  - ScienceDirect
  - ACM Digital Library

    - SpringerLink
    - Wiley Online Library
    - JSTOR
    - IEEExplore
    - WorldCat
- Open search engines
    - Google
    - Yandex
- Snowballed reference lists
- GPT-4 (ChatGPT)

Searches began in Google Scholar, then ScienceDirect, and progressed through the remaining scientific search engines and databases; searches then progressed to the open search engines, Google and Yandex, to potentially locate non-indexed literature, e.g., books; then, the literature search began anew within the bibliographies of the selected literature, following a technique referred to as snowballing (Wohlin, 2014); finally, GPT-4 was prompted in various ways to provide novel search recommendations. Snowballing is perhaps one of the more efficacious means of augmenting a SLR, inasmuch as it allows the author to quickly encounter relevant, cited literature that other authors have labored to synthesize, thereby reducing the possibility of missing relevant literature.

3.1.1.3. SEARCH PHRASES

In general, the search phrases employed are informed by the research questions defined earlier, in Section 1.3. However, these questions are specific to the objective of the present paper, which is to determine the applicability of Agile for scientific writing. This is a much more directed and assumptive work than an SLR, and, no literature exists regarding the use of Agile methods for scientific paper writing. So, the following, more general research questions was used to guide this supporting SLR:

- What scientific writing methods are used?

This question is broken into individual pieces, then expanded with synonyms using a combination of manual effort and GPT-4 prompting, e.g., “scientific writing method” becomes:

- (scientific OR academic OR scholarly OR research OR manuscript OR thesis OR dissertation)
- (writing OR drafting OR publication OR recording OR communication OR authorship)
- (method OR approach OR technique OR framework OR style OR strategy OR guide OR guideline OR best practice OR model OR protocol OR standard OR principle OR procedure OR process OR methodology OR system OR standard OR convention OR structure)

With simple combinatorics, the possible number of unique search phrases is calculated to be 840 (). The search phrases were created by linking the three OR-separated strings with Boolean AND, insofar as a potential search phrase could be “scholarly writing protocol”.

3.1.1.4. Article Selection Process

Initial article searches invariably return large numbers of totally irrelevant papers (Kitchenham & Charters, 2007); these are not recorded in the selection/exclusion process here. Candidate papers were selected by reading their titles, abstracts and conclusions, and determining if their contents were relevant to the question (and its resultant search phrase) at hand; any articles not selected based on this preliminary evaluation were discarded as irrelevant non-candidates. Candidate articles were later, after compilation into an annotated bibliography (see Section 3.1.2.1), included or excluded based on the inclusion/exclusion criteria established earlier.

3.1.2. Search Documentation and Records

To support study replicability, the search and data collection processes of this SLR were explicitly recorded. The tasks of this recording effort are given in the following sections.

3.1.2.1. Data Management

Articles initially selected for final selection against the previously stated inclusion/exclusion criteria had their PDF versions downloaded and saved locally. Further, an annotated bibliography, in the form of a Microsoft Excel spreadsheet, was used to annotate the important information about each paper, as well as ultimately record the statement of acceptance/rejection of each paper.

3.1.2.2. Data Collection

For the selected articles, data were collected by reading the text entirely and noting the particular scientific writing method described, as well as transcribing any potential diagrammatic representations of said method in textual form, if need be. Metadata on the location of the articles were also recorded. See Section 3.1.3.

3.1.3. Data Items

Following are the data items collected for each candidate article entered into the annotated bibliography.

- Metadata:
    - What exact search phrase located the article?
    - What outlet was the article published in?

- Content specific data:
    - What facet of scientific writing does the method seek to improve, e.g., study execution, literature reviews, whole-paper structuring, holistic writing-to-publication work, etc.?
    - In what field(s) is/are the method proposed for, e.g., biomedical research?
    - Does the method extend on or diverge from any other noted method(s)?
    - What is the title of the method, if it has one?

3.1.4. Risk of Bias

Given the expected nature of the individual articles surveyed, which are likely to describe particular, unique scientific writing methods or techniques, there was an expected inherent bias for each article, as authors may present their method in a better light than others. In the data collection of the present paper, this bias was reduced by simply noting the unique facets or steps of each method, without regard to the methodological discourse of the providing authors. So, in the synthesis of the SLR here, broad comparisons across all encountered methods were given, to consider each method with respect to the others, without attempting to propose the advantages or disadvantages of them.

3.1.5. Data Synthesis

The data of the SLR is synthesized as a large table containing several organization columns, and thereafter broken down in narrative manner in attendant paragraphs, with analysis across the various themes of the selected papers.

3.2. SURVEY OF ENGINEERING PROFESSIONALS

As the SLR provides general insight into scientific paper writing methods that may be rather antiquated, it is of interest to the present paper to conduct a supplementary survey of software practitioners, to ascertain how software professionals conduct their scientific paper writing process. As most of the software practitioners surveyed have used, or are deeply familiar with, Agile methods, this survey provides a novel, first-hand accounting of potentially Agile-inspired scientific paper writing and the effectiveness of Agile in this context. Moreover, it provides a contemporary, albeit limited in sample size, summarization of common scientific paper writing techniques.

3.2.1. SURVEY FORMATION

TABLE II Pilot study respondent characteristics.

Respondent	Age	Job Title	Pilot Study Response Time (minutes)
1	26	ML Engineer	5
2	27	Systems Engineer	14
3	25	Software Engineer	6

The survey (see Appendix A) consists of 3 demographic questions (numerical slider and free response), 7 basic questions about scientific writing practices (multiple-choice), 12 specific questions about scientific writing practices (multiple-choice) and 4 personal questions about scientific writing practices (free response).

Although primitive, a method for estimating the total time to complete the survey was utilized, to allow the provision of a time estimate to participants. A pilot study was conducted on three engineers, who were given the survey and asked to report their start and end times to the nearest minute, rounded up. The demographics of these three engineers are represented in Table II. This allowed for the calculation of an average time per question (), where represents a participant and represents the number of questions:

$$T_{avgQ} = \frac{T_1 + T_2 + T_3}{3N}$$

That in turn allowed for the calculation of an estimated total time for the survey. Let represent the estimated total time to complete the survey:

$$T_{total} = T_{avgQ} \times N$$

After calculation, the average time to complete the survey was estimated to be 8.3 minutes, or, roughly, 8 minutes and 20 seconds. This is an acceptably short length for a survey, as longer survey times result in less cooperation and completion by respondents (Galesic & Bosnjak, 2009).

3.2.2. Survey Execution

The survey was administered using Google Forms. Respondents were emailed a link to the survey and asked to complete it at their leisure. Google Forms compiled the answers and allowed direct analysis.

4. Results

This section presents the results of the SLR and the survey. It is followed by a brief summarizing analysis.

4.1. SLR Results

In total, 559 papers were surveyed. Of these, 81 papers were included for candidacy in final selection. Of these 81 candidate papers, less than 20 were selected for final inclusion and analysis in the following sections. The primary reason for this large reduction in relevant literature is that very few novel, named or otherwise unique methods, techniques or approaches to scientific writing

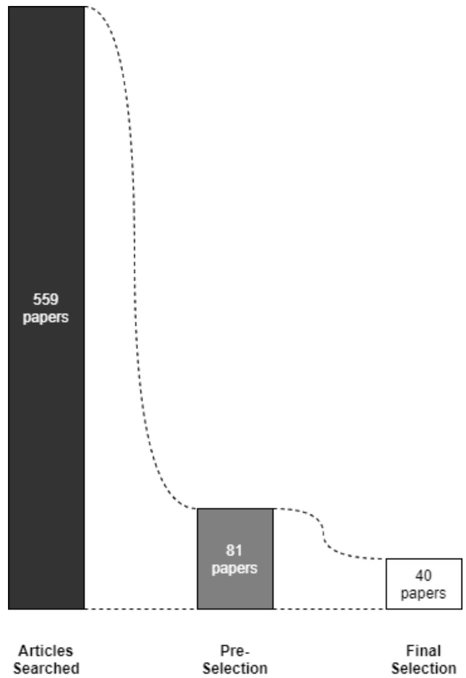

Fig. 3 Flowdown visualization of the paper selection for the conducted SLR.

were encountered, even with the very extensive combinatoric search phrase detailed in the earlier Methods section. The snowballing technique resulted in no new included literature outside of a handful of articles relevant in the background section of the present paper, or as hedging elsewhere. In order to provide a more substantial SLR, non-named, published *books* on scientific writing were included as well (i.e., the final inclusion criteria were relaxed), bringing the total number of included articles to 40.

As such, the results of the SLR were quite surprising. Most included papers referenced either the IMRaD or hourglass models. Many of the papers surveyed were particular to the biomedical, surgical or clinical domains; in fact, IMRaD has its roots in biomedical writing (Sollaci & Pereira, 2004). Although generally the accepted form (and, potentially, operative order) of scientific writing, some have suggested that IMRaD be rearranged, e.g., so that the sequence of writing is MRDaI (Pollock, Evans, Wiggin, & Balch, 1991). As discussed earlier, these are essentially prescriptive frameworks for the canonical form a scientific paper should take, bound up in decades of social form and custom, stemming primarily from the modern zeitgeist of science which proliferated with the establishment of the Royal Society in the mid-17th century. IMRaD and hourglass are not, however, *process* models, i.e., they only fully consider the continuant pieces of the artifacts (papers and their composite sections), and not the occurrents that spawn them (research and writing). In other words: the form of the scientific paper is well-established and unlikely to change; but the process of arriving at this form is insufficiently researched. This is why, in the present paper, Agile methods of product development are evaluated for their feasibility in facilitating scientific research and writeup.

There were encountered many articles pertaining to teaching scientific writing methods to undergraduate students in various domains (Niemitz & Potter Jr., 1991; Phadtare, Bahmani, Shah, & Pietrobon, 2009; Jerde & Taper, 2004; Woodford, 1967; Holstein, Steinmetz, & Miles, 2015; Clabough & Clabough, 2016; Cuschieri, Grech, & Savona-Ventura, 2018). These were not included in the SLR as candidate papers unless they contained an evaluation of an identifiable, novel method.

Many papers that were rejected for final inclusion pertained to the communication facet of scientific papers, e.g., empirical investigations into word choice, rhetorical style (such as passive vs. active voice), linguistic complexity, text parsimony, text recycling, etc. (Moskovits, 2019; Ping Alvin, 2014; Lu, et al., 2019; Hebb & Bindra, 1952; Bizzoni, Degaetano-Ortlieb, Fankhauser, & Teich, 2020). Many of these papers employed machine learning or other data analysis techniques upon large corpora of scientific paper texts to derive insight for discussion. In any case, these were not included for final analysis because no explicitly identifiable method for scientific writing was found in any of them.

4.1.1. Synthesis of Results

The results of the conducted SLR are synthesized below. In order to delineate the contribution of the SLR more clearly, the selected papers are organized in tabular form, using three high-level facets or focuses that are fiat, i.e., defined here arbitrarily for better understanding: Structure, Process and Communication. The first pertains to the canonical "pieces" of scientific articles, or those prescribed by, or built on, IMRaD, i.e., "what to write"; the second, to the time-bound actions of scientific writing, i.e., "how to manage writing"; and the latter, to the more granular, technical elements of scientific writing, i.e., "how to write".

- Structure: the continuant elements of scientific papers
    - Title
    - Abstract
    - Introduction
    - Materials, Methods
    - Results
    - Discussion
    - Conclusion
    - Meta-structure, e.g., thesis articles
- Process: the occurrent or temporal-bound portions of research, writeup and publication
- Communication: the technical elements of scientific papers, e.g., sentence case, perspective, pacing, paragraph structure, etc.

TABLE III Articles included for final analysis in the SLR.

Title	Citation	Article Type	Field	Article Facet / Focus	Method Title (if applicable)	Mentions
Scientific Writing and Communication in Agriculture and Natural Resources	(Nair & Nair, 2014)	Book	Natural Sciences	All	-	IMRaD
A Scientific Approach to Scientific Writing	(Blackwell & Martin, 2011)	Book	Unspecified	All	-	-
Scientific Writing = Thinking in Words	(Lindsay, 2010)	Book	Unspecified	All	-	IMRaD
Writing up Research: Experimental Research Report Writing for Students of English	(Weissberg & Buker, 1990)	Book	Unspecified	All	-	Hourglass Model
Communicating in Science: Writing a Scientific Paper and Speaking at Scientific Meetings	(Booth, 1993)	Book	Unspecified	All	-	-
An Outline Of Scientific Writing: For Researchers With English As A Foreign Language	(Yang, 1995)	Book	Unspecified	All	-	-
The art of scientific writing: from student reports to professional publications in chemistry and related fields	(Ebel, Bliefert, & Russey, 2004)	Book	Natural Sciences	All	-	-
Writing for science	(Goldbort, 2006)	Book	Unspecified	All	-	IMRaD
Writing scientific research articles: Strategy and steps	(Cargill & O'Connor, 2021)	Book	Unspecified	All	-	IMRaD, AIRDaM, AIM(RaD)C, AIBC
Successful Scientific Writing A Step-by-Step Guide for the Biological and Medical Sciences	(Matthews & Matthews, 2014)	Book	Agnostic	All, Process	-	IMRaD
From Research to Manuscript: A Guide to Scientific Writing	(Katz, 2009)	Book	Unspecified	All, Structure	-	-
Research Methodology and Scientific Writing	(Thomas, 2021)	Book	Unspecified	All, Thesis	-	IMRaD
Scientific writing skills: Guidelines for writing theses and dissertations	(Lourens, 2007)	Book	Unspecified	All, Thesis	-	-
The Artof Science Writing	(Worsley & Mayer, 1989)	Book	Education	Process	-	-
Scientific writing for psychology: Lessons in clarity and style	(Kail, 2018)	Book	Medical Sciences	Structure	-	-
An Investigation of Students' Science Writing Processes Using Think-aloud Method	(You, Kang, Kim, & Noh, 2013)	Empirical	Education	All	Think-Aloud Method	-

Structuring the composition process in scientific writing	(Patterson, 2001)	Empirical	Education	All	Context Mapping*	-
Writing Research Article Introductions in Software Engineering: How Accurate is a Standard Model?	(Anthony, 1999)	Empirical	Engineering	All, Introduction	-	Create a Research Space (CARS)
Mind-to-paper is an effective method for scientific writing	(Rosenberg, Burcharth, Pommergaard, & Danielsen, 2013)	Empirical	Education	Process	Mind-to-Paper	-
Experimental evidence for diagramming benefits in science writing	(Barstown, Fazio, Schunn, & Ashley, 2017)	Empirical	Education, Medical Sciences	Structure, Communication	Argument Diagramming	-
Writing Good Abstracts	(Alexandrov & Hennerici, 2007)	Theoretical	Medical Sciences	Abstract	AB(solutely) STR(aightforward), ACT(actual data & interpretation)	IMRaD
Scientific Writing 3.0: A Reader and Writer's Guide	(Lebrun & Lebrun, 2021)	Theoretical	Agnostic	All	-	-
Algorithm for Writing a Scientific Manuscript	(O'Connor & Holmquist, 2009)	Theoretical	Agnostic, Medical Sciences	All, Communication	Algorithm for an Initial Draft*	-
Basics of research paper writing and publishing	(Derntl, 2014)	**Theoretical**	**Unspecified**	**All, Structure**	**The King Model**	**Hourglass Model**
A Conceptual Framework for Scientific Writing in Nursing	(Regan & Pietrobon, 2010)	Theoretical	Medical Sciences	All, Structure	Four-Phase Model	-
Integrating Writing Frames into Inquiry-Based Instruction	(Subramaniam, 2010)	Theoretical	Education	All, Structure	Frame-Based Writing*	-
The Principles of Biomedical Scientific Writing: Citation	(Bahadoran, Mirmiran, Kashfi, & Ghasemi, The principles of biomedical scientific writing: citation, 2020)	Theoretical	Agnostic, Medical Sciences	Citations	-	IMRaD
The Principles of Biomedical Scientific Writing: Materials and Methods	(Ghasemi, Bahadoran, Zadeh-Vakili, Montazeri, & Hosseinpanah, 2019)	Theoretical	Agnostic, Medical Sciences	Materials and Methods	-	IMRaD
Enhancing research publications and advancing scientific writing in health research collaborations: sharing lessons learnt from the trenches	(Li, et al., 2018)	Theoretical	Medical Sciences	Process	Experience Sharing*	-
The Logic of Scientific Writing	(Volpato, 2011)	Theoretical	Unspecified	Process, Communication	Logical Method for Scientific Writing*	-
"Grab" and Good Science: Writing Up the Results of Qualitative Research	(Gilgun, 2005)	Theoretical	Unspecified	Process, Communication	Grab vs. Good Science*	-

Writing Science: What Makes Scientific Writing Hard and How to Make It Easier	(Grogan, 2021)	Theoretical	Unspecified	Process, Thesis	-	Specific, Measurable, Achievable, Relevant and Time-Bound (SMART), Pomodoro technique
The Principles of Biomedical Scientific Writing: Results	(Bahadoran, Mirmiran, Zadeh-Vakili, Hosseinpanah, & Ghasemi, 2019)	Theoretical	Agnostic, Medical Sciences	Results	-	IMRaD
The introduction, methods, results, and discussion (IMRAD) structure: a fifty-year survey	(Sollaci & Pereira, 2004)	**Theoretical**	**Agnostic, Medical Sciences**	**Structure**	**IMRaD**	**-**
Improving the writing of research papers: IMRAD and beyond	(Wu, 2011)	Theoretical	Agnostic	Structure	Evolutionary IMRaD*	IMRaD
Scientific Writing & the Scientific Method: Parallel "Hourglass" Structure in Form & Content	(Schulte, 2003)	Theoretical	Agnostic	Structure	Parallel Hourglass	Hourglass Model
On Scientific Writing in the Information Era: Tailoring Papers for Internet Searching and Other 21st Century Realities	(Hamby, 2015)	Theoretical	Unspecified	Structure, Communication	-	-
How to avoid common errors in writing scientific manuscripts	(Maiorana & Mayer, 2018)	Theoretical	Agnostic	Structure, Communication	Errors-First*	IMRaD
The essentials of effective scientific writing – A revised alternative guide for authors	(Sayer, 2019)	Theoretical	Agnostic	Structure, Communication	Audience-First*	-
The Principles of Biomedical Scientific Writing: Title	(Bahadoran, Mirmiran, Kashfi, & Ghasemi, The Principles of Biomedical Scientific Writing: Title, 2019)	Theoretical	Agnostic, Medical Sciences	Title	-	IMRaD
* indicates a method title provided here but not in the source article, for the purpose of synthesizing said article						

4.1.2. Individual Article Analysis

The papers by Derntl and Sollaci et al. present the seminal publications on hourglass and IMRaD formats, respectively (Derntl, 2014; Sollaci & Pereira, 2004). These entries are bolded in Table III.

Fifteen articles did not specify any field of relevance, and eleven were pointedly agnostic with respect to application field, so their contents are likely applicable to any domain of interest. Otherwise, the natural sciences (e.g., agriculture), medical sciences and education were the most commonly specified application areas.

Fifteen published books on scientific writing were identified by the SLR. Five of these fourteen referenced IMRaD, and one referenced the hourglass model. All but one book pertained to all facets of scientific writing (Structure, Process and Communication), although one had a sub-focus on structure (Katz, 2009), and two had the sub-focus of theses writing (Thomas, 2021; Lourens, 2007); the remaining book had a specific focus on the process of writing (Worsley & Mayer, 1989). As mentioned previously, none of these books presented a named method for scientific writing, but they were included to supplement the rather small SLR, as their content, often 200+ pages in length, may be considered to constitute unique scientific writing methods on their own.

Four papers by the same group of authors were revealed by the SLR; each one pertained to a different IMRAD-derived element, to wit, Materials and Methods, Results, Citations and Title (Ghasemi, Bahadoran, Zadeh-Vakili, Montazeri, & Hosseinpanah, 2019; Bahadoran, Mirmiran, Zadeh-Vakili, Hosseinpanah, & Ghasemi, 2019; Bahadoran, Mirmiran, Kashfi, & Ghasemi, The principles of biomedical scientific writing: citation, 2020; Bahadoran, Mirmiran, Kashfi, & Ghasemi, The Principles of Biomedical Scientific Writing: Title, 2019). The authors have papers for the other sections, but these were not revealed by the SLR. These papers are included because they provide general insight to scientific writers in addition to advice specific to biomedical writers.

You et al. present the results of the Think-Aloud method, which encourages young researcher to express their thoughts verbally before writing (You, Kang, Kim, & Noh, 2013). Rosenberg et al. present the Mind to Paper (MTP) technique, which, similar to the previous paper, uses verbal dictation to assist authors in creating a first draft from an outline (Rosenberg, Burcharth, Pommergaard, & Danielsen, 2013). Li et al. propose several strategies to enhance publication in the health sciences, culminating with the idea that experience sharing and "embracing the iterative process of writing" is central to success (Li, et al., 2018). Volpato presents a "logical" method for scientific writing, that encourages authors to structure their articles and report research according to a consistent logic, in order to produce more cogent papers (Volpato, 2011). Schulte presents an addition to the hourglass method that implores authors to consider the components of the hourglass structure as parallel to the stages of the scientific method, e.g., observation to introduction, hypothesis to objectives, experimentation to methods/results, conclusion to discussion, and the iterative process as further publication (Schulte, 2003). Maiorana and Mayer present an "errors-first" scientific writing method which encourages writers to consider common writing errors up-front, and additionally how to address them in the most efficacious manner (Maiorana & Mayer, 2018). Sayer poses the scientific paper as an author-centric artifact, with specific focus on keeping the reader interested so the paper can be fully utilized (Sayer, 2019). Grogan presents an interesting discussion on scientific writing, mentioning the use of the Pomodoro technique, which is a means of working that employs periods of rest, e.g., 40-minutes work and 20-minutes rest, or 20-minutes work and 5-minutes rest (Grogan, 2021).

In general, the articles of the SLR unsurprisingly present scientific writing as an iterative process, but there were no truly codified, systematized project management methods, such as software engineers have with Agile, or Waterfall.

4.2. SURVEY RESULTS

This section contains the results of the survey conducted on engineering professionals, with the goal of ascertaining a general understanding of their method of scientific writing, in order to compare such with that gathered from the SLR conducted previously. The questions of the survey were worded to be as general as possible, with no mention of Agile, to mitigate the elicitation of responses with bias.

4.2.1. SURVEY DEMOGRAPHICS

In total, fourteen individuals responded to the survey. Every one of them was employed or enrolled at an engineering university at the time of the survey. The most common bachelor's degree held by respondents was Software Engineering (n = 5), followed by Computer Science (n = 2). See Figure 5. Artificial Intelligence / Machine Learning was the most popular research topic (n = 5), followed by Systems Engineering (n = 3) and Education (n = 2). See Figure 6. The weighted average of the number of years the respondents had experience in scientific writing and publication was calculated to be 5.64 years (see Figure 4).

4.2.2. SURVEY SYNTHESIS

The following sections contain the results of the survey; it is broken down into three parts:

- Multiple Choice Questions – Basic
- Multiple Choice Questions – Processual
- Free Response Questions

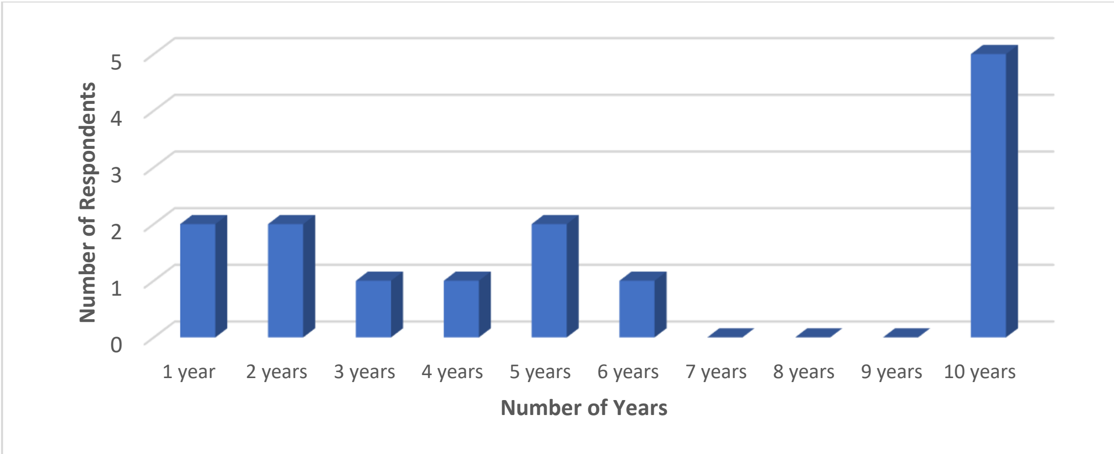

Fig. 4 Survey responses to the question *“How many years of experience do you have in publishing scientific papers?”*.

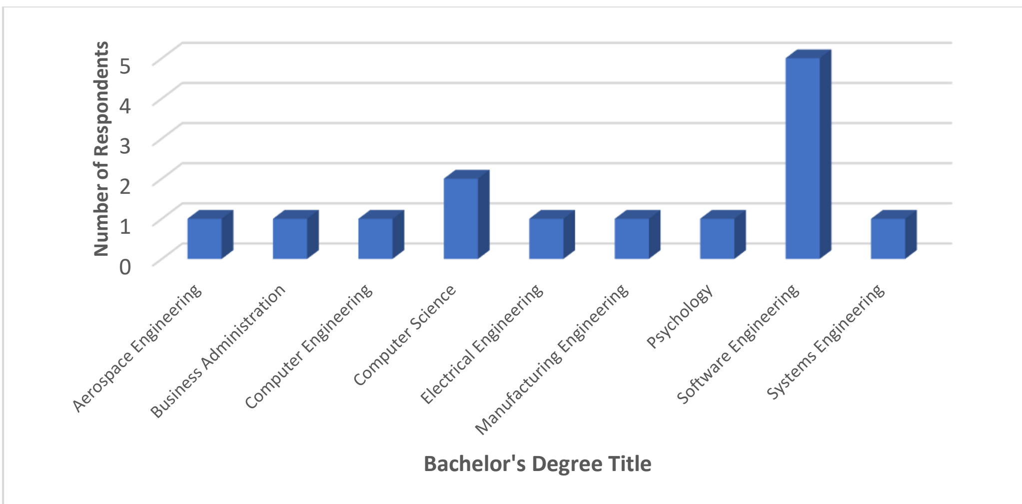

Fig. 5 Survey responses to the question *“What is the title of the bachelor's degree you graduated with?”*.

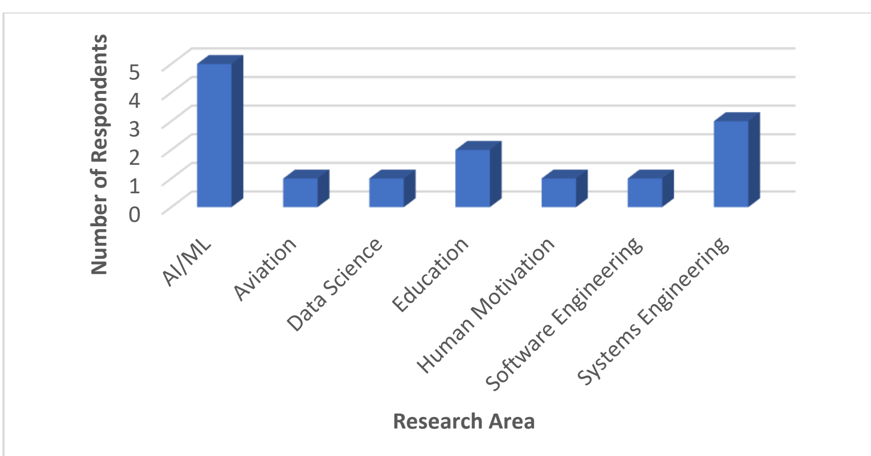

Fig. 6 Survey responses to the question *“What is your primary field of research?”*.

4.2.2.1. MULTIPLE CHOICE QUESTIONS – BASIC

The response data for the basic multiple-choice questions can be viewed in Figures 7-13. For basic question 2 (see Figure 8), one respondent stated that they used Curvenote, an online collaborative scientific writing application built on top of Jupyter Notebook, to write their scientific papers (Basu, 2023). For basic question 4 (see Figure 10), one respondent noted that their revision time for papers varies, stating "… some papers take longer than others. And how do you measure it? Is in one sitting? No paper I've ever spent editing for one week, usually in bursts. But if you consider bursts, some journal papers have taken definitely more than a month.". For basic question 5 (see Figure 11), two respondents asserted that their selection of a publication outlet is performed through a combination of general Internet searches, consulting with colleagues and using personal experience/references.

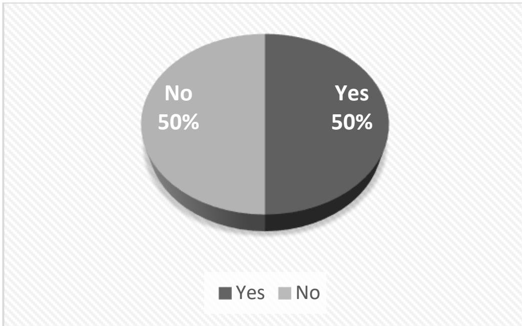

Fig. 7 Survey responses to the question “Have you ever taken an academic course specific to scientific writing?”.

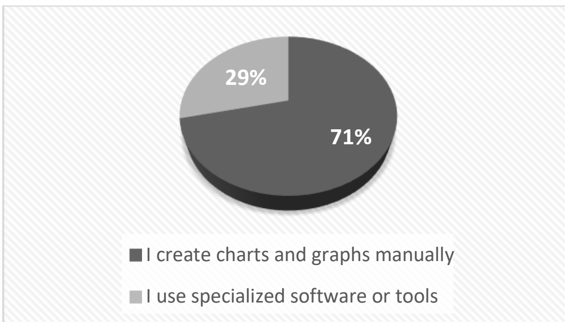

Fig. 9 Survey responses to the question “How do you usually handle data visualization and chart creation in your papers?”.

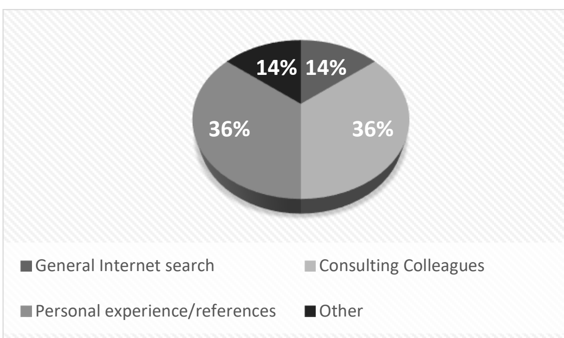

Fig. 11 Survey responses to the question “How do you select a publication outlet?”.

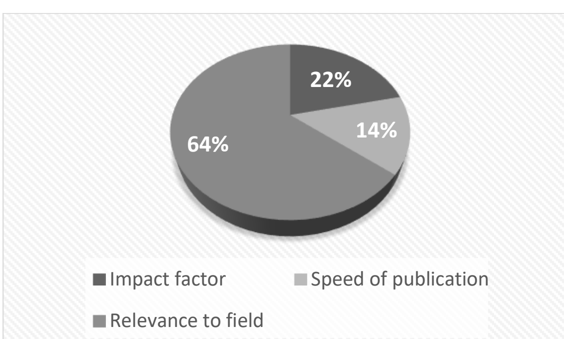

Fig. 13 Survey responses to the question “Which factor is most important to you when selecting a journal for submission?”.

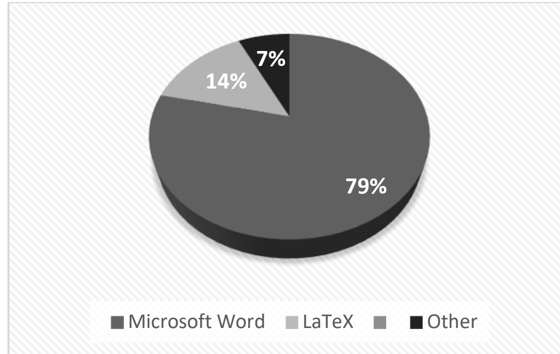

Fig. 8 Survey responses to the question “Which software/tools do you primarily use for drafting your scientific papers?”.

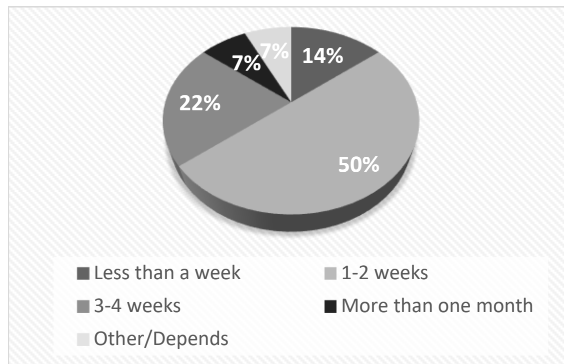

Fig. 10 Survey responses to the question “How much time do you usually spend on revising and editing a paper before submission?”.

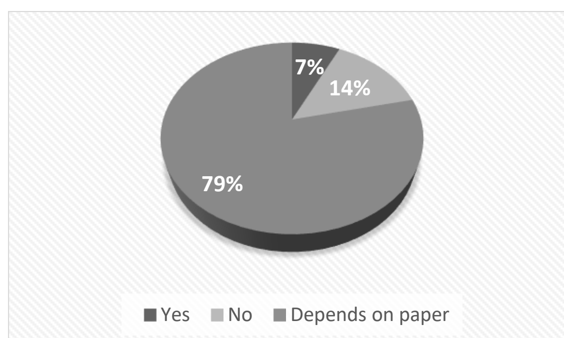

Fig. 12 Survey responses to the question “Do you use preprint servers for early dissemination of your research?”.

4.2.2.2. Multiple Choice Questions – Processual

The response data for the processual multiple-choice questions can be viewed in Figures 14-25. For processual question 1 (see Figure 8), one respondent answered that their approach to starting a scientific paper was to outline first. For processual question 3 (see Figure 16), one respondent noted that their approach to collaboration with co-authors was a combination of real-time co-writing with online tools, dividing/combining sections and discussing ideas verbally while one person writes. For processual question 4 (see Figure 17), one respondent noted that they organized paper writing tasks with Trello (an online Kanban board), and another noted that they used Overleaf (an online LaTeX collaborative writing tool). For processual question 6 (see Figure 19), one respondent noted their prioritization in detail, stating "Hard to describe. I feel it out. Sometimes I am motivated to write a really interesting introduction section to catch the reader. Sometimes I have results and can start there. Typically I write what feels right, first. If I'm struggling on a section, I move to another. So, I guess the second answer 'ease of completion', is what I'm saying.". For processual question 7 (see Figure 20), one respondent noted that they typically consider a section of their paper complete "when they cannot stand to look at it anymore". For processual question 11 (see Figure 24), one respondent stated that they balance writing papers with other responsibilities by setting a minimum requirement for words written per day. For processual question 12 (see Figure 25), one respondent said that their use of iterative/cyclical methods in their writing process involves "[going] back and forth with my co-authors [to] point out particularly weak areas that I know they (with their expertise) can maybe fix.".

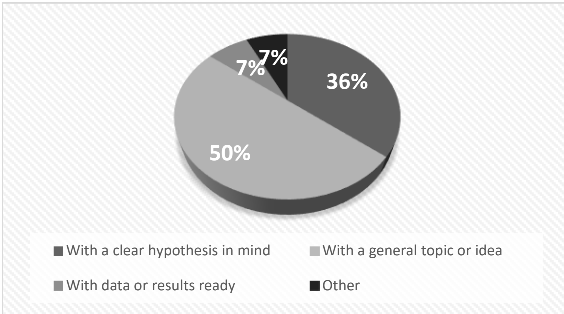

Fig. 14 Survey responses to the question “How do you typically begin writing a scientific paper?”.

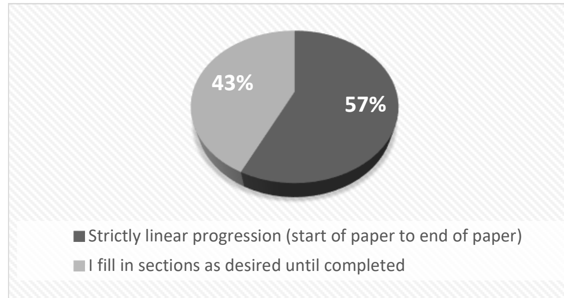

Fig. 15 Survey responses to the question “When writing a scientific paper, do you progress linearly, or do you fill in sections as you progress in the understanding of your topic?”.

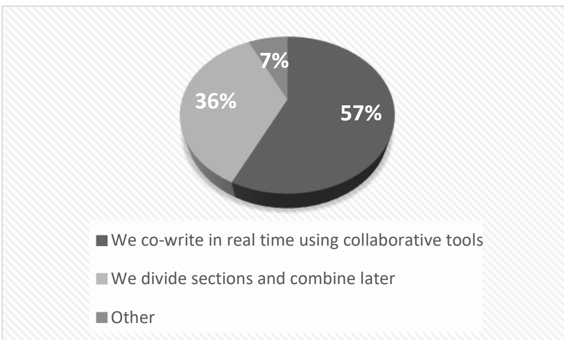

Fig. 16 Survey responses to the question “How do you approach collaboration with co-authors?”.

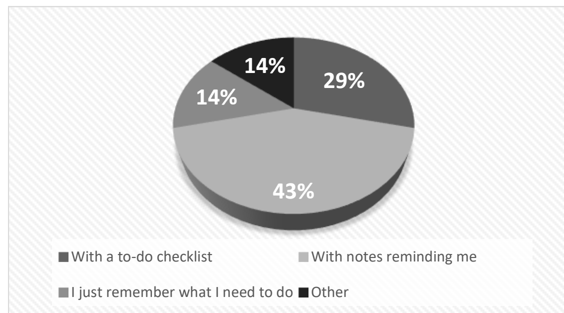

Fig. 17 Survey responses to the question “How do you organize tasks during the writing of a paper?”.

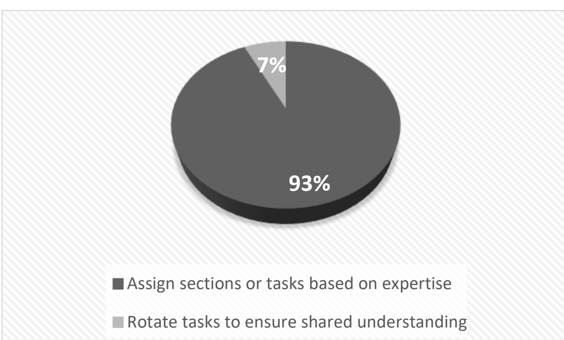

Fig. 18 Survey responses to the question “How do you approach task delegation when collaborating on a paper?”.

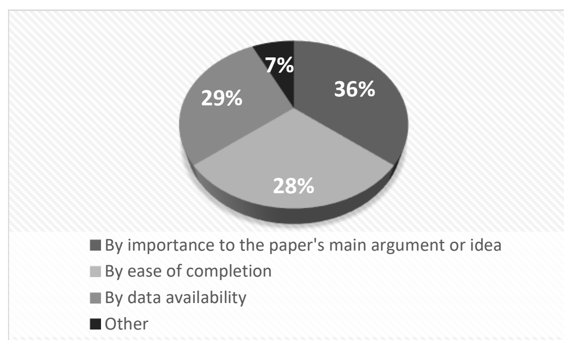

Fig. 19 Survey responses to the question “How do you usually prioritize sections or elements in your papers?”.

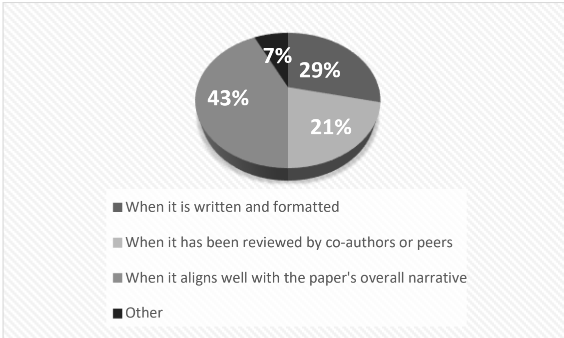

Fig. 20 Survey responses to the question “When do you typically consider a section of your paper "complete"?”.

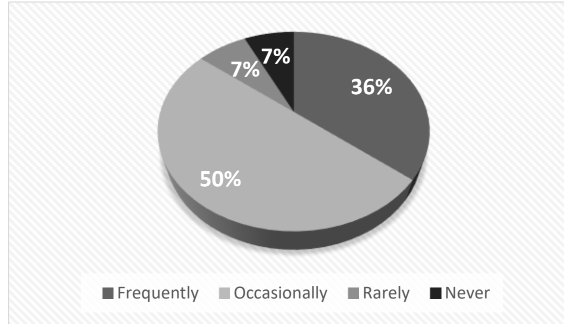

Fig. 21 Survey responses to the question “How frequently do you revisit and revise sections after considering them "complete?”.

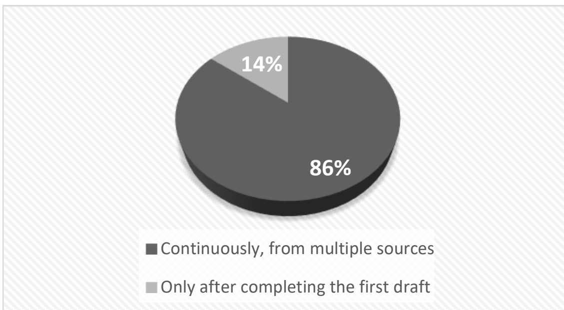

Fig. 22 Survey responses to the question “How do you handle unexpected changes or challenges, e.g., new data, conflicting results, etc., during your writing process?”.

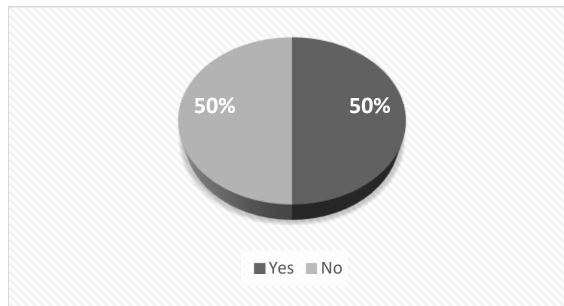

Fig. 23 Survey responses to the question “Do you set specific milestones or deadlines for different sections or stages of your paper?”.

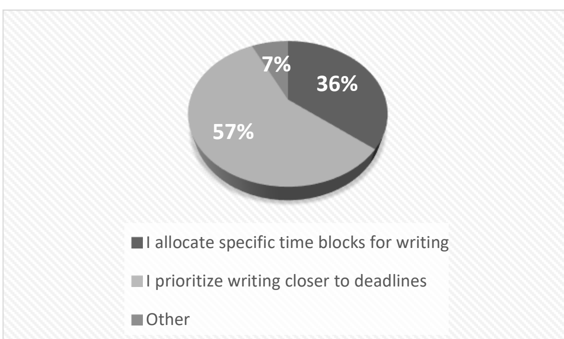

Fig. 24 Survey responses to the question “How do you balance writing the paper with other responsibilities?”.

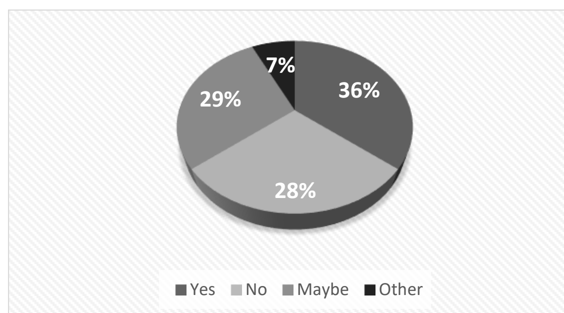

Fig. 25 Survey responses to the question “Do you incorporate any iterative or cyclical methods in your writing process (e.g., Write-Review-Revise cycles)?”.

4.2.2.3. Free Response Questions

This section contains respondent answers to 4 free response questions. These free-form answers give greater insight into the particular processes engineering writers abide by when creating a scientific paper for publication. The more notable responses are reproduced in turn.

TABLE IV Survey responses to free response question 1.

Can you describe any challenges you frequently encounter during the writing and publication process for scientific papers?	
Respondent	**Answer**
Respondent 1	If it's a niche topic, finding sources or similar works is challenging
Respondent 2	new info about the subject forces revision of the main ideas in a paper
Respondent 3	Waiting too long for peer-review and getting rejected is a pain. This is why I almost always publish open-access, like with SSRN, so regardless, the paper is out there and getting views/citations, building credibility so even if rejected, we can try again with an already recognized paper.
Respondent 4	Sometimes I'm halfway through writing paper before I realize that the idea doesn't work or is trivial.
Respondent 5	I also sometimes struggle to expand on topics when writing without sounding repetitive.
Respondent 6	LITERATURE REVIEWS. I'm always missing an important reference.
Respondent 7	Disconnect or lack of flow from chapter to chapter
Respondent 8	Writer's block. Sometimes there isn't inspiration for a particular idea even though I know it is relevant/important, or my inspiration comes and goes continuously in waves.
Respondent 9	Lack of time to focus on writing, slow response or review time from journals
Respondent 10	Waiting for review comments
Respondent 11	Publication fees can sometimes be more than is available and you must find a source of funds from department, college, or university to cover the fees.
Respondent 12	Time availability
Respondent 13	The lead time to publish is too long. Relinquishing the rights to your work just to publish is morally challenging.
Respondent 14	Formatting is a big issue for us. In particular certain publications require specific formats

For free response question 1 (see Table IV), there is a general, negative feeling towards publication, particularly publication fees, licensing and review timeframe. Four respondents noted that waiting on journal/conference reviewers is a particular point of contention for their work.

Three respondents note difficulties with technical writing concerns, e.g., repetitive text, disconnect between paper sections and writer's block. Four respondents state that finding relevant sources, or incorporating new sources that may invalidate their current work, are common challenges. Other concerns include time availability (presumably of team members) and formatting for publication outlets.

TABLE V Survey responses to free response question 2.

What strategies or practices have you found most effective in successfully publishing your work?	
Respondent	**Answer**
Respondent 1	Write about a relevant topic
Respondent 2	printing and revising using hard copies instead of computer screen
Respondent 3	Not peer review in the sense of a conference/journal, but review by my actual peers in-office, especially my advisor, who has an uncanny ability to find appropriate publication outlets that nearly always accept our work.
Respondent 4	Clear writing that is easy to read and understand. When writing, I try to write for someone who has some background knowledge, but is not an expert in the field.
Respondent 5	Iterating over the paper. Start by just throwing text in a document, then forcing myself and others to read it. The feedback I generate and get from others helps me reorganize the paper and clarify important details.
Respondent 6	Clear outline and key ideas for each chapter or subchapter
Respondent 7	Setting aside time blocks to work on my papers rather than working on it as time becomes available, as time is never available.
Respondent 8	sticking to a deadline when the work needs to be completed; write with co-authors
Respondent 9	Use journals you are familiar with and have published in prior, so they know your reputation.
Respondent 10	Setting aside blocks of [sic] tiem
Respondent 11	Outline, fill in technical sections, review/revise, add front matter and back matter, review and revise again, and then add abstract. Submit and wait.
Respondent 12	Quality work
Respondent 13	Publish often. It helps understand how to write for scientific audience understanding.
Respondent 14	Persistence is the key to success

For free response question 2 (see Table V), respondent answers varied. Three respondents noted dependence on co-author review. Two respondents note the use of iterative, review/revise cycles.

TABLE VI Survey responses to free response question 3.

If you could describe your method for writing a scientific paper in one sentence, what would that sentence be?	
Respondent	**Answer**
Respondent 1	I write one section at a time
Respondent 2	revise-revise-revise
Respondent 3	Hypothesize, write title/abstract, write a "hook" introduction, give reader just enough background in that section (not a "walk through the zoo") to know the general topic, then write the meat (results, analysis, etc.), and discuss and conclude with forward-looking statements.
Respondent 4	I really try to stress the importance of the research and explain what the work can be used for.
Respondent 5	Put all thoughts onto the paper and revise into an accepted format for technical writing.
Respondent 6	Only the genius controls chaos
Respondent 7	Organized process consisting of sequential atomic tasks that are documented in Trello so that the current and next tasks at hand are clear.
Respondent 8	Get all my thoughts out and clean it up later
Respondent 9	Come up with a good idea based in theory, then collect data, write it up and publish.
Respondent 10	Write [sic] consistantly
Respondent 11	Write with the expectation to revise, start with what matters most, then add the "packaging/framing/background" after you have a solid foundation.
Respondent 12	Publish on topics that are new and potentially transformative.
Respondent 13	Idea (write abstract and conclusion 1st draft), literature review (write literature review) , hypothesis (reform abstract, conclusion, write experimentation plan), experimentation (write next steps to capture what you wish you could have done differently, what you wish you had time to do), results (write results, finish abstract and conclusion).
Respondent 14	I feel that my method for writing has become a repeatable proven process.

For free response question 3 (see Table VI), which was the most direct, yet general, attempt at extracting explicit writing processes from the survey respondents, some interesting insight was gained. Overall, there is a respondent emphasis on revision and iteration. Two respondents (3 and

13) provided very detailed, comma-separated workflows. Respondent 7 stated that they organized their scientific writing as an "organized process consisting of sequential atomic tasks that are documented in Trello…". This is amenable to the earlier discussion of Agile Kanban boards, because Trello is an online, collaborative Kanban board. Two respondents noted that they prefer to "get it all out on paper" before revising for publication, which is interesting from an Agile perspective, as this phase of "mind dumping" can be likened to software prototyping.

TABLE VII Survey responses to free response question 4.

How has your approach to writing and publishing changed over the course of your career?	
Respondent	**Answer**
Respondent 1	I became more organized on finding out what I wanted to write about before starting the writing. And now I also prioritize writing relevant information rather than filling out the space, the paper might be shorter but it's better quality
Respondent 2	unable to comment
Respondent 3	It's gotten easier. My first paper took me one year to write. Now I write about 10 papers per year. The main thing that got easier was my ability to find and read relevant sources to then cite (hedge) my own propositions with.
Respondent 4	As I gain experience and knowledge, I'm getting better at understanding the literature and using that to help form the basis for my papers.
Respondent 5	I've become less concerned about the quality of the first draft and rely on the feedback of others more.
Respondent 6	Slightly improved
Respondent 7	I have not been publishing for very long, but the process is becoming expedited with time.
Respondent 8	I used to be better at it when I had more time to [sic] rea , think, and write
Respondent 9	Yes, not as important now, I often give younger co-authors first author because they need to get tenure.
Respondent 10	Less frantic, more scheduling
Respondent 11	I am now more selective with the venues I publish.
Respondent 12	One constantly adapts to what is cutting edge research, so staying updated to the new developments is essential.
Respondent 13	I didn't put as much emphasis on the abstract and conclusion but now I spend most of my time making sure they clearly detail the paper. They are the tldr of the paper. You shouldn't need to read the whole paper to understand the paper.
Respondent 14	I've become more willing to not publish something perfect, stop obsession over small flaws

For free response question 4 (see Table VII), the general consensus between respondents is that scientific writing, on the level of the individual as well as the team, is a process of constant improvement. This mirrors the Agile principles about team and process improvement, and is relevant in the context of Agile sprint retrospectives, which are aimed at improving future spring efficacy and efficiency.

5. Analysis

Although the results of the SLR were limited, inasmuch as the identification of new or novel scientific writing techniques outside of the IMRaD or the hourglass models was not explicitly possible, the administration of a small survey of engineering professionals brings fresh insight to the topic of employing Agile techniques in scientific research, writing and publication. In the present paper, a later section addresses the size of the survey (n = 14) as the primary limitation, but regardless, the survey provides a digestible baseline for future research on the subject. Some respondent answers lend credence, although preliminary, to the hypothesis that Agile techniques, such as Kanban boards and write-review-revise iterations, are applicable to scientific writing.

Many of the Agile values/principles, such as continuous feedback and prioritizing team communication, are mirrored in some of the scientific writing techniques surveyed, e.g., Think-Aloud method and Mind to Paper (You, Kang, Kim, & Noh, 2013; Rosenberg, Burcharth, Pommergaard, & Danielsen, 2013).

This provides an initial validation of the hypothesis that Agile techniques may be beneficial to scientific writers, and an Agile framework, such as Scrum, can be used my scientific writing teams to produce papers of higher quality.

6. Related Work

As discussed earlier (see Section 2.3.2), Agile has been employed in various domains outside of software. Notwithstanding, very little literature on the use of Agile for research work are extant. These are detailed in turn.

Ramos et al. present the use of Scrum in research-oriented projects (Ramos, Ramos, Viana, Silva, & de Oliveira, 2016). Within the context of a student research group, the authors discuss issues and lessons learned with Scrum-based research projects. They note that, among other observations, the Scrum approach had to be adjusted, because the research group was focused on activities other than development, e.g., research itself. They present several suggestions for the improved use of Scrum in research settings, including fine-tuning the number of daily meetings each week, using cloud services to store sprint deliverables, creating groups in messaging applications, and choosing a Scrum Master with comprehensive research knowledge, Scrum experience and communication skills.

Twidale and Hansen seek an answer to the question "how can we apply the underlying philosophy of Agile software development and explore them in the context of research?" (Twidale & Hansen, 2019). Contrasting with Big Design Up Front (BDUF) methods, e.g., Waterfall, the authors highlight the applicability of Agile to research, noting several surveyed constructs, e.g., ethnographies, contextual inquiries, bodystorming and small-scale user testing. They discuss the use of sprints, breaking such into five stages: Map/understand, sketch, decide, prototype and validate. They delineate the similarities and differences between software development projects and research projects: "In both software development and research we may not be completely sure what the requirements are. It is certainly possible to have a nice precisely articulated requirements document and this will not change. But it does not always happen. It may be somewhat more

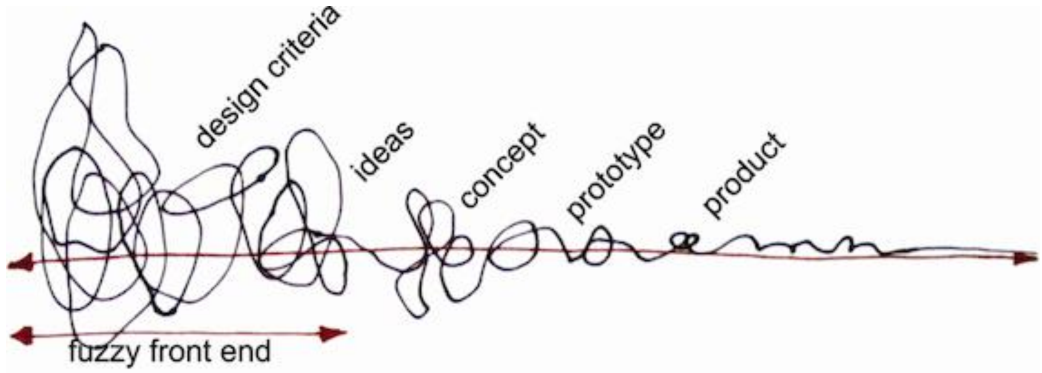

Fig. 26 The iterative co-design process.

common to have a nice precisely articulated set of research questions and these will not change. But that does not apply to all research… Furthermore, where did that precisely articulated research question come from? How was it constructed? Did it emerge fully formed, or was it itself the result of an iterative design process? Just as a software development project can fall into delays, cost overruns and unexpected complexities, so can a research project... In both software development and research we may wish that we had the foreknowledge to have been able to have specified everything perfectly at the outset through a [Big Design Up Front]… it would be nice if we could catch problems earlier through tighter iterations. That is the starting point to considering agile methods." (Twidale & Hansen, 2019). They draw a corollary with media design processes, which often rely on rapid prototyping; the notion of co-design, or the collective creativity of collaborating designers (see Figure 26), is applicable to the discussion of Agile research (Sanders & Stappers, 2008). The authors are cautious to note the following: "We do not want to imply that agile research methods… can work by simply directly mapping agile software development methods into a research setting, or to serve as an excuse to do fast sloppy work to bang out yet more publications. Nevertheless we think that there is great promise in further exploring alternative faster lighter more tightly iterative ways of doing exploratory research." (Twidale & Hansen, 2019).

Svoboda's Master's thesis presents an analysis of Agile implemented in an informatics laboratory, where there was witnessed an increase in quality of advisory meetings, student motivation, laboratory affinity and membership satisfaction (Svoboda, 2021). It is based on a survey of 6 previous case studies of Agile methods employed in research settings. Despite the promising results of the paper, there is no mention of scientific writing and publication.

Cruz et al. present the most applicable work: their paper reports the use of Scrum for writing a scientific article (Cruz, et al., 2021). Although the paper is rather informal (and only available in Spanish), it is nonetheless rife with valuable insight. For the formation of a product backlog, they begin by outlining Scrum roles: Product Owner (sponsor or advisor), Scrum Master (lead author) and Scrum Team (lead author and co-authors). Then, they used these roles to develop a collection of user stories, e.g., for the user story titled “Introduction”, the content is thus: “As the lead author, I want to write the introduction section of the article that contains prior information on the topic to contextualize the reader about its content”. After user stories are aggregated, they are prioritized using the MoSCoW technique (Must Have, Should Have, Could Have and Won’t Have) in combination with Planning Poker, or Scrum Poker, which is a gamified technique for estimating priority and time-to-complete for sprint tasks that factors in the estimates of all team members. These user stories, now estimated, are then detailed further with acceptance testing requirements, e.g., the user story “Keywords” was given the acceptance requirement “When three to five words are differentiated to detail the content and the reference to search for the article, ordered by importance or impact on the content, then distinction of keywords is concluded”. These detailed user stories are then broken apart and codified as tasks in an online Scrum tool with a Kanban board. Once the product backlog is complete, the authors note the use of two week sprints to complete the project, where each sprint is loaded with a manageable number of tasks from the total product backlog. The authors also discuss the use of a Definition of Done (DoD), daily Scrum meetings and sprint retrospectives.

7. Proposal: Agile Academia

Given the various scientific writing methods encountered through the SLR, and the contemporary insight gained from the survey of software practitioners, it is possible, now, to propose a mapping between Agile systems development and scientific paper writing: Agile Academia.

7.1. Mapping Agile Between Software Development and Scientific Writing

Modern scientific writing is typically performed with access to the Internet, using a computer device to write text, create diagrams and tables, keep track of references, and many other specific tasks. Seeking improved efficiency and efficacy in scientific writing, below are proposed the values for Agile Academia, mapped to the original Agile values for software development. While acknowledging the cautionary stance of earlier researchers in attempting to explicitly map Agile to research settings (Twidale & Hansen, 2019), it is posited that, to establish a baseline for evaluating the feasibility of Agile Academia in future case studies, it is necessary to perform such a mapping, insofar as later researchers may use it as a guide. What is provided below is not so much a prescription as it is a preliminary description of the mapping. This roughly follows the work set forth by Stewart et al. for the domain of education (Stewart, DeCusatis, Kidder, Massi, & Anne, 2009).

TABLE VIII Mapping of Agile software values to scientific paper writing.

Agile Software Value	**Scientific Paper Writing Corollary**
Individuals and interactions over processes and tools	Synergy among researchers and interactive discussions over strict adherence to writing tools and processes
Working software over comprehensive documentation	Concise and effective communication of findings over overly detailed documentation
Customer collaboration over contract negotiation	Active engagement and feedback from the academic community over rigid adherence to writing and publishing protocols
Responding to change over following a plan	Responding to feedback and unexpected results over following a plan

The same can be done for the Agile software principles, by drawing corollaries between the two disciplines. The ultimate essence of this mapping is to maximize adaptability through quick iterations and continuous development, in pursuit of more efficient and effective dissemination of scientific knowledge.

TABLE IX Mapping of Agile software principles to scientific paper writing.

Agile Software Principle	Scientific Paper Writing Corollary
Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.	Our highest priority is to faithfully record experimentation and present the most cogent, quickly disseminated research articles to other researchers; the "customer" is, ideally, not a publication outlet, but other researchers.
Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.	Researchers welcome unexpected findings, even late in the reporting process. Agile scientific writing uses new problems to incorporate more relevant citations and improve current methods, resulting in more complete and useful papers.
Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.	Prioritizing early drafts and iterative feedback from researchers on the team, as well as selected reviewers, e.g., professorial advisors, allows constant improvement of papers.
Business people and developers must work together daily throughout the project.	Researchers, writers and reviewers (ideally, reviewers selected prior to outlet submission *and* those assigned during/after submission) must work iteratively together throughout the lifecycle of a paper.
Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.	With a topic they have interest in, most researchers are highly motivated to augment their knowledge into a paper; give them the environment and support needed to progress.
The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.	While online collaboration tools are useful, insofar as it is possible, encourage face-to-face, same-desk interaction between researchers and writers.
Working software is the primary measure of progress.	Coordinated sections of cogent text, as well as relevant citations, tables and diagrams, are the primary measure of progress, not necessarily a paper formatted for some particular outlet.
Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.	Agile scientific writing intends to be sustainable. All team members should be able to work at a constant pace, from paper inception to publication and maintenance.
Continuous attention to technical excellence and good design enhances agility.	Persistent focus on rigorous scientific method and clear, thoughtful structuring of papers improves the adaptability and effectiveness of academic research.
Simplicity--the art of maximizing the amount of work not done--is essential.	Simplicity is key; granting that appropriate citations to relevant, outside bodies of work are given, the more direct a scientific paper is, the more easily it can be digested and used by other researchers.
The best architectures, requirements, and designs emerge from self-organizing teams.	Research groups should self-organize and assign tasks to the right people, but all members should contribute equally to the paper.
At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.	Periodically, the team should evaluate their writing and collaboration processes to identify improvements, adapting their strategies and practices to enhance overall research productivity and paper quality.

Reiterating the soul of the Agile manifesto: Agile Academia does not prescribe any one Agile variant, e.g., Scrum (although Scrum is a well-established and suitable project framework for many fields). Those intending to research and write in an Agile way should experiment with several

Agile variants and their innumerable activities, e.g., pair programming from XP, daily standup meetings from Scrum, and so on.

The following sections consider the six stages of the product development lifecycle (requirements elicitation, design, implementation, testing, deployment and maintenance) and Agile activities specific to each of them, with conceptual mappings to the domain of scientific paper writing.

7.1.1. Scientific Paper Requirements Elicitation

Requirements elicitation begins the lifecycle of software development: customers are interviewed and explicit product requirements are Agile practices would inherently bind team members closer to one another, resulting in better communication, feedback, revision, etc. As such, the paper construction process would be improved, because both the natures of science (or scientific writing) and software development, are iterative.

(Pacheco, Garcia, & Reyes, 2018). These requirements form the basis of work, and the contract between developer and client.

Scientific papers also have requirements driven by client wishes, although this is not so salient as with software systems. For instance, future readers of a scientific paper (probably) wish for it to be complete, cogent, digestible, linked to other work, replete with diagrams and tables, etc. Although these clients are not typically conversed with in the typical sense, the wishes of a scientific writer, or their experience with others, manifest in tacit requirements for their written papers; i.e., no scientific writer wishes to write a *bad* paper, or to read one. Drawing from Agile software engineering, it is possible to conceptualize requirements elicitation activities for beginning the lifecycle of a scientific paper.

7.1.1.1. User Stories

A user story "describes functionality that will be valuable to either a user or purchaser of a system..." (Cohn, 2004). Further, a user story embodies three key items:

1. A written description of the story (plan)
2. Conversations about the story (details)
3. Tests to determine when a story is complete (acceptance criteria)

User stories typically manifest as cards on a project board, and can be broken down into tasks and sub-tasks, that developers choose to work on, or are assigned to, for a given iteration of a project. The fulfillment of user stories implies project progression.

In the context of writing a scientific paper, user stories should pertain to three user groups:

1. Readers
2. Reviewers
3. Self

In the first case, it is important to address the needs and wants of other researchers who may be reading an article, e.g., as it is beneficial for them to have an understandable, cogent, well-coordinated paper. For reviewers, principally if a journal or conference outlet is decided upon in advance, it is important to create user stories making their review efforts easier, and adhering to any stated guidelines. For the self, or the team, writing the paper, it is important to optimize credibility, by writing exhaustively yet cogently, considering a good volume of outside literature, addressing potential issues in the research and lending insight to future work.

7.1.1.2. PRODUCT BACKLOG, KANBAN BOARDS AND BURNDOWN CHARTS

For some software development project, the product backlog is a list of items to address and implement in software, e.g., user stories, bugs, “chores”, etc. (Sedano, Ralph, & Péraire, 2019); i.e., the product backlog contains the to-be-fulfilled wishes of the customer.

The Kanban board is a visual, interactive means of measuring project progress, by decomposing a project into a “to-do list”, allowing for an explicit limitation of project scope through a visual representation of the project workflow (Alaidaros, Omar, & Romli, 2021). The primitive, canonical Kanban board consists of three groups of tasks: TO DO, IN PROGRESS and DONE; more complex Kanban boards can include groups of finer granularity, e.g., BACKLOG, TESTING, etc. There are a great number of Kanban software tools available for use by teams (Corona & Pani, 2013), but the use of a physical room with a chalkboard, whiteboard or corkboard on a wall, whereupon papers containing backlog items are grouped with tape or tacks, is a well-established means of utilizing a Kanban board. Kanban boards provide development teams with a transparent, visible project progress measure.

Kanban boards are limited in visualizing overall project progress or sprint timeliness, as they essentially provide a “snapshot” of current progress (Alaidaros, Omar, & Romli, 2021). As such, the use of another project progress visualization is advisable: the burndown chart. Burndown charts track the work remaining against a fixed timeframe, e.g., a sprint or set deadline. Kanban burndown charts may also manifest as cumulative flow diagrams, which visualize the number of items in each state (TO DO, IN PROGRESS, DONE, etc.) over time. The burndown chart is useful for development teams in ensuring that they stay on track and on time.

Scrum makes use of boards, but Scrum boards are constructed per sprint, i.e., they cannot be added to during sprints, whereas Kanban boards contain the entire backlog and can update as needed. It is not the intent of the present paper to prescribe one over the other in the context of scientific paper writing, as both may be efficacious; in fact, a combined method, Scrumban, finds use in various fields (Reddy, 2015).

Nevertheless, it is incumbent on any scientific writing teams to decide which is best for their particular work: for more unpredictable projects, e.g., those still involving experimentation, a Kanban board is best, as it is more flexible; for more predictable projects, e.g., literature reviews, a Scrum board is better suited because of its structured sprints. In short: Kanban is less prescriptive, and therefore more adaptive than Scrum, where Scrum is more organizational, and therefore more rigorous; but both methods find success and their practices can be intertwined (Kniberg & Skarin, 2010). Teams may even find it beneficial to name their Kanban-style board groups after the IMRaD format, preceded by a backlog group and appended with a group named DONE. The survey of the present paper (see Section 4.2) revealed that at least one respondent used Trello, an online Kanban-style board, to manage paper tasks.

7.1.2. Scientific Paper Design

Scientific papers can benefit from efforts in conceptual design prior to writing. Just as software systems consist of interconnected modules serving particular purposes to fulfill some product goal, so too do scientific papers consist of coordinated sections of text, figures and tables, fulfilling the ultimate purpose of representing some experiment, relaying a theory, or some such insight, for the benefit of other researchers.

7.1.2.1. Diagrammatical Design

The SLR identified one unique paper that employed the use of argument diagrams to visually, and more concretely, represent the need, motivation and purpose of scientific papers, in addition to the traditional lexical argument style (Barstown, Fazio, Schunn, & Ashley, 2017). Such an affordance would typically manifest in the canonical introduction section of papers, as this is the section where the overall intent of a paper is typically delineated.

Argument diagramming has two primary components (Reed, Walton, & Macagno, 2007):

1. Propositions (premises or conclusions) represented as points or nodes
2. Inferences represented as lines

Translating the implications of scientific argument diagramming into the context of software diagramming can be done by considering a particular type of software diagram, known as the use case diagram. The use case diagram is, perhaps, the simplest of all typically realized software design diagrams, being composed of actors (users or external systems), a box delineating the system being developed, and use case bubbles inside, which represent system functionalities. Actors are connected to use cases with lines, and use cases may connect to one another for deeper relationships. Mapping between the two: premises may be thought of as actors (they initiate actions that lead to outcomes), conclusions as use cases (outcomes of actor interaction), relations as arguments (inferences) and finer relations (e.g., extends and includes) as counter-arguments and sub-arguments.

For the purpose of graphically delineating an Introduction section, which typically summarizes the purpose or need of some paper, a use case diagram inspired by the argument diagramming paradigm may be beneficial for the writing team, to cogently summarize the intent of their paper and keep their objective clear as their writing progresses. Within the published paper itself, however, a use case diagram is unlikely to be beneficial, as readers have their own reasons for interacting with said paper; so, the use of an actual argument diagram (without use case actors) within a paper may be beneficial to visually represent the paper's intent and reasoning.

Fig. 27 The general Scrum lifecycle (Permana, 2015).

7.1.3. Scientific Paper Implementation

After requirements have been documented, and some preliminary design performed, software systems are implemented iteratively until a satisfactory state is reached, insofar as the product can be delivered. Scientific papers go through an implementation phase where outlines are fleshed out, paragraphs written, graphics created, citations gathered, etc. The Agile notion of a sprint may find use here.

7.1.3.1. Sprints

Agile sprints are a fundamental component of the Agile variant, Scrum. A sprint is a short, time-boxed period, typically two to four weeks, during which a specific set of work must be completed and made ready for review (Schwaber & Sutherland, The 2020 Scrum Guide (TM), 2020). Each sprint begins with a planning meeting where the team identifies the work to be done, followed by the actual development work, daily Scrum meetings (check-ins), and ends with a sprint review and retrospective. The goal of a sprint is to produce a potentially shippable product increment, ensuring rapid and continuous delivery of valuable software. Sprints allow teams to break down complex projects into manageable chunks, enabling frequent reassessment and adaptation of plans.

As argued in the present paper, and as other authors have noted (Cruz, et al., 2021), scientific papers are iterative systems with a small set of requirements, and a deliverable timeframe, e.g., publication timing. So, short, Scrum-like sprints of 1-4 weeks are likely very amenable to scientific paper project management. This is reinforced by the findings of the survey in Section 4.2, which revealed that 50% of engineering respondents spent 1-2 weeks revising or editing papers before submission; additionally, 36% of the respondents stated that they explicitly used iterative write-review-revise cycles, and 29% thought they might, but were unsure. Although somewhat speculative, it is posited that Agile sprints may be very beneficial to scientific writing teams as a means of managing project workload and timing.

7.1.3.2. Code Re-use and Referencing

Software systems often make use of extant code libraries to achieve desired functionality. In much the same way, scientific papers rely on citations to reinforce narratives, hedge particularly extended statements, and generally provide context to readers.

The use of citations should be precise and not superfluous (Lalumière, 1993). Grogan implores scientific authors to refrain from the tendency to look for the "perfect citation", which can be an endless cycle of searching (Grogan, 2021). It may be asserted that, like software engineers make what is available to them work for their purposes, often with modified code, scientific authors should cite other works as needed, and introduce new text to contextualize their use of citations.

7.1.3.2. Pair Programming

Pair, or collaborative, programming, is one of the core activities of the XP Agile variant. Pair programming places two developers side by side at the same computer, where they work together on some code. Early literature on pair programming employed in software development indicated that, with a cost of slightly (15%) more development time as opposed to individual development, pair programming resulted in less software defects, higher software design quality, faster problem solving, improved communication and better learning (Cockburn & Williams, 2000).

In much the same way, pairs of researchers may write sections of scientific text together, catching one another's grammatical or technical mistakes, resulting in less cleanup during review. This engenders an environment of continuous review, similar to the desirable state of continuous code reviews in pair programming, resulting in fewer defects and better output quality.

7.1.4. Scientific Paper Testing

Although not regarded as tested systems, scientific papers indeed undergo extensive testing before release, both internally, with team member review, and externally, with journal/conference review.

7.1.4.1. Test-Driven Development

Test-driven development (TDD) is an Agile activity that uses automated tests with an iterative test-first mindset. Developers begin by creating an atomic automated unit test, then writing some small amount of code to pass the test; this is repeated until the system has been fully developed. Test-first teams have been evaluated as more productive than no-test and test-last teams (Janzen & Saiedian, 2006).

In the context of scientific paper creation, multiple automated unit tests are not practical. However, the work of Cruz et al. demonstrated efficacy when scientific paper user stories were appended with acceptance tests written in natural language, to be fulfilled by writing new sections of text, adding images, formatting paragraphs, etc. (Cruz, et al., 2021). Maiorana and Mayer present the "errors-first" method of scientific writing, which encourages authors to address common writing errors with immediacy (Maiorana & Mayer, 2018). With the provision of "test-driven writing", scientific papers may progress with greater reliability, allowing writers to be more productive.

7.1.4.2. Artifact Reviews and Walkthroughs

Peer code review typically occurs in-person at the desk of a developer requesting (or assigned) code review for some new module, with one or more reviewers. Many studies indicate that peer code review catches more defects than without peer review, resulting in improved software quality (Bavota & Russo, 2015; McIntosh, Kamei, Adams, & Hassan, 2016).

An indication of the efficacy of peer review in detecting defects can be found from the pilot study of the survey presented earlier. In response to the question "What strategies or practices have you found most effective in successfully publishing your work?", one respondent answered: "Review by lab mates or other students. My most successful papers were all reviewed by lab mates and friends before I even attempted to publish. Someone who is not involved in the project every day can more easily spot errors in the way I've presented information than I can. Their questions help me fill in conceptual gaps and/or streamline the material in the paper.", and another answered "Asking for help from my peers".

7.1.4.3. Rubber Duck Writing

Rubber duck debugging is an exaggeration of the tendency for programmers to "talk things out", even if to themselves, to aid in the act of debugging non-working code; the rubber duck (typically figurative, although not always) is a medium for the programmer to express their thoughts without judgement. Similar to pair programming, rubber duck programming may witness an increase in software quality (Bryant, Romero, & du Boulay, 2006). Scientific writing methods evaluated on young researchers, e.g., the Think-Aloud and Mind to Paper techniques, revealed that the verbal dictation of thoughts or paper outline material resulted in papers that were of easier to read, and of higher quality in general (You, Kang, Kim, & Noh, 2013; Rosenberg, Burcharth, Pommergaard, & Danielsen, 2013).

7.1.5. Scientific Paper Deployment

The deployment, or release, of scientific papers, may be well suited in the context of source control, or versioning, platforms, such as Git, which is commonly used by software development teams to track changes, issues and merge content from various team members. The popular online Git service, GitHub, has a built-in “Project” feature, that allows team members to track issues in digital Kanban-style boards and burndown charts. Issues can be directly assigned to team members, commented on for further clarification, closed if completed, re-opened if needed, etc. GitHub repositories can be made public, allowing anyone to view, interact with and propose changes to source artifacts.

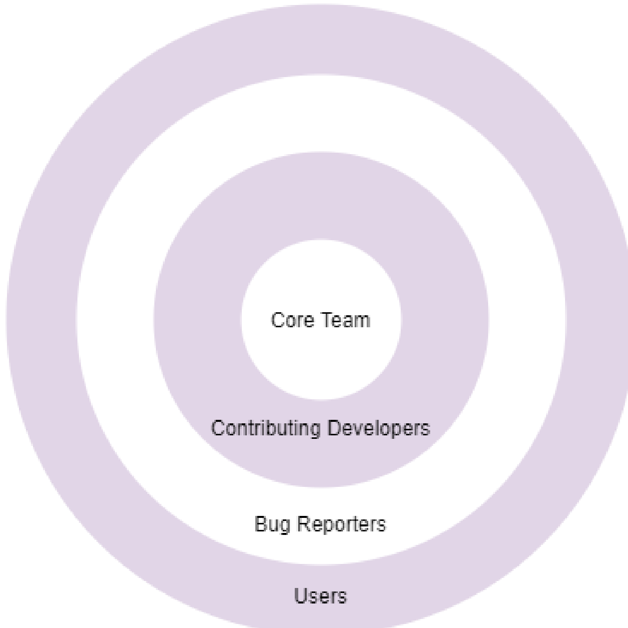

Fig. 28 The four-layer "onion model" of OSS. The closer to the center a user group is, the closer their control over the project.

7.1.5.1. OPEN-SOURCE PAPERS

Open-Source Software (OSS) is regarded as the movement responsible for very stable, high quality software, e.g., as the Linux operating system is OSS (Bretthauer, 2001). OSS can be defined, roughly, as software that has its source code freely and publicly available, with limited or no restriction on the modification and use of such (The Open Source Definition (Annotated), n.d.). OSS researchers have modeled the community interactions of OSS with a four-layer onion model (Bahamdain, 2015; Aberdour, 2007). See Figure 28. Each community group contributes in some measure to the development of some OSS project, with the core team having the tightest control, contributing developers having some control, bug reporters having slight control, and general users having very little control over project trajectory. The outer layers of the OSS community onion feed inward and inform project development.

In the same way, scientific papers have four community groups that inform the initial development, and potential maintenance, of their content:

1. Core author team
2. Advisors / colleague reviewers
3. Journal / conference reviewers
4. Readers

Conceptually framing scientific papers as open-source systems is beneficial for both writers, as they can benefit from reviews and testing, and for readers, as they receive a more cogent product of higher quality.

7.1.6. Scientific Paper Maintenance

Scientific papers are, ideally, maintained by the authors, if for no purpose other than to correct mistakes undetected during publication, resulting in new paper versions. Maintenance engenders continued quality assurance and benefits to readers.

7.1.6.1. Errata, Corrigenda

The accumulation of an errata, or a list of errors noted after the publication of some text, is typically informed by invested readers and performed for larger publications, i.e., books. However, both author errors (corrigenda) and publisher errors (errata) can manifest with scientific paper writing, as humans invariably make mistakes, and some publishing software, which convert working files, e.g., Microsoft Word files, can incorrectly augment resultant PDFs.

As the last canonical phase of the software development lifecycle, maintenance feeds forward into the first phase, requirements elicitation; i.e., new requirements from customer feedback during the maintained, deployed phase of some software enables developers to update their system. In much the same way, the maintenance of an errata for a scientific paper after publication can be a means of generating new requirements for the paper, such that it can be iterated upon briefly to correct its errors. However, unlike software systems, scientific papers should not be "updated", insofar as their original intent is lost; at most, maintenance errata for scientific papers should be relegated to grammatical, formatting and other presentation-oriented concerns.

These scientific errata and corrigenda can manifest as new user stories or tasks adding to a backlog to be iterated upon at some later date, under the framework of an Agile sprint or Kanban cycle.

Any content-related corrigenda should be preserved in a separate list, forming a body of requirements for a *new* scientific paper to be written by the authors at a later date. Such lists can engender new iterations for subsequent scientific papers that are informed by the previous, as many authors naturally tend to do as their research progresses and they encounter new articles of relevance. Schulte presents the hourglass model as parallel to the scientific method, with the iterative process of the scientific method on the same level as further publication (Schulte, 2003).

7.2. AGILE ACADEMIA IN ACTION

Every Agile activity conceptually mapped to the act of scientific writing, discussed hitherto, can be, if desired, implemented by the scientific author, or a team of them, to organize and accelerate the lifecycles of their papers being labored upon.

Although the IMRaD model is a basis for the canonical scientific paper structure, it is rather simplified and lacking many of the more specific facets of scientific papers. Procko et al. present, in their paper about the use of GPT-4 in scientific writing, a more detailed taxonomy of the scientific paper, including such things as sub-titles, keywords, sub-sections, tables, figures, acknowledgements, etc., in addition to formatting concerns, such as font sizing, spacing, etc. (Procko, Davidoff, Elvira, & Ochoa, 2023). Combined with Derntl's hourglass variant, the king model (Derntl, 2014), the two provide the structure and backlog, respectively, for scientific papers.

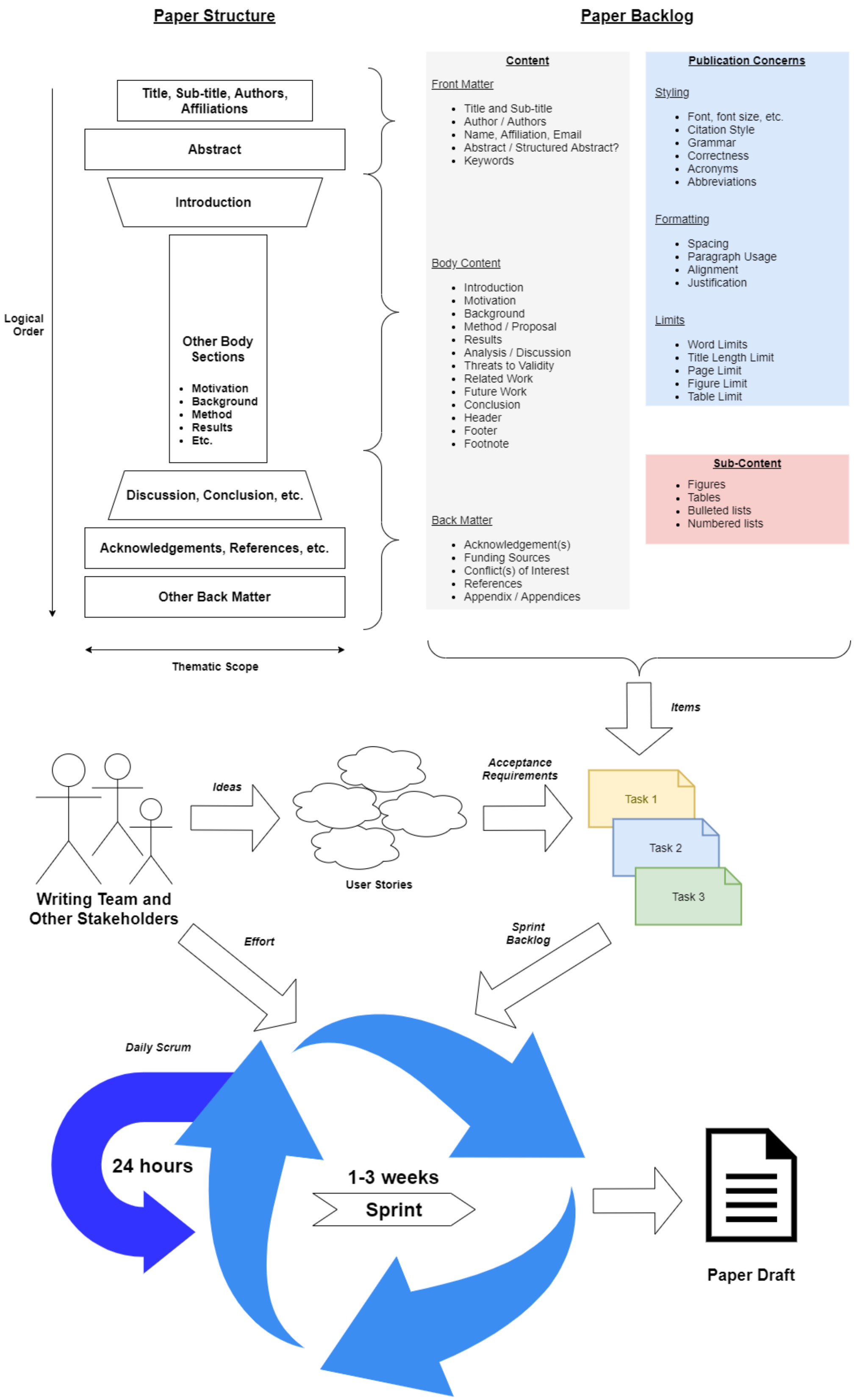

Fig. 29 The proposed Agile Scrum scientific paper development lifecycle.

As stated previously in the delineation of Agile Academia, no specific Agile variant is specified, but Scrum is included because of its definition of sprints and longstanding use in various domains. Specific Agile practices, e.g., pair programming, test-driven development, kanban boards, user stories, etc., can be used and weaved into the Agile Academia lifecycle as needed by scientific writing teams, depending on their needs and goals.

8. DISCUSSION

The path to Agile Academia is clear. Although there was not witnessed a single, well-codified scientific writing method in the SLR, some related works show the potential of Agile methods, such as Scrum, in research settings (Ramos, Ramos, Viana, Silva, & de Oliveira, 2016; Twidale & Hansen, 2019; Svoboda, 2021; Cruz, et al., 2021). Many of the Agile activities, e.g., use of Kanban project boards, sprints and pair programming, have obvious parallels to traditional writing processes. These were detailed in turn for the six stages of the SDLC: requirements elicitation, design, implementation, deployment, maintenance and testing. The preliminary survey conducted for the present paper as supplement to the SLR revealed that, in fact, a handful of respondents used Kanban boards and iterative write-review-revise cycles, both of which are central to the Agile variant, Scrum. So, it is possible to answer the guiding research questions presented earlier.

8.1. ANSWERING RESEARCH QUESTIONS

RQ1: How feasible is the adaptation of the Agile method to the construction, refinement and dissemination of scientific papers?

Answer 1: The feasibility of this goal is high, given the similarity between Agile activities and those typically conducted when writing scientific papers, e.g., code reviews and co-author reviews. While maintaining an awareness of the witnessed difficulties of employing Agile in fields outside of software development (Niederman, Lechler, & Petit, 2018), scientific paper writing typically occurs on computers, with small teams iterating over a single artifact, much like Agile software development teams do; so, Agile adaptation to scientific paper writing is quite feasible.

RQ2: How does implementing Agile influence the quality of scientific papers when compared to traditional paper development processes?

Answer 2: Quality may be increased because scientific papers are relatively small, constrained systems developed by small, close teams; and, as Agile practices, e.g., pair programming, test-driven development, sprints and Kanban boards, consistently demonstrate improved programmer productivity and code quality, it may be inferred that, if applied appropriately in the context of scientific writing, the resultant scientific papers may be of higher quality than if written with traditional practices. Cruz et al. show preliminary results of Agile in scientific writing (Cruz, et al., 2021).

RQ3: How does employing the Agile method affect the efficiency of the paper construction process and the collaboration between co-authors and reviewers?

Answer 3: Agile practices would inherently bind team members closer to one another, resulting in better communication, feedback, revision, etc. As such, the paper construction process would be improved, because both the natures of science (or scientific writing) and software development, are iterative.

8.2. IMPROVING SCIENTIFIC WRITING WITH AGILE

The SLR conducted reduced the included papers from 559 papers to 40. Many of the rejected papers pertained to a more complicated aspect of scientific writing, which is the quality of the writing itself. Although some works in the SLR addressed this explicitly, e.g., with verbal dictation techniques (You, Kang, Kim, & Noh, 2013; Rosenberg, Burcharth, Pommergaard, & Danielsen, 2013), most of the articles included in the SLR pertained to "tips" or "rules of thumb" for putting together scientific papers for publication.

What is good writing? More specifically: what constitutes good scientific writing? A good scientific article is precise, clear and brief, and "as much an exercise in clear and focused thinking as it is in clear and accurate writing" (Lindsay, 2010). The Japanese have a phrase, *fusoku-furi*, which means "to maintain a neutral attitude", but the term is bound up in cultural feelings about writer competence, clarity of thought and syntactic structure, which undoubtedly vary significantly from the feelings of Occidental scientists (Kaplan & Grabe, 1991).

In any case, as has been posited in the present paper, if Agile can be beneficial to scientific writers for improving their project lifecycle productivity and overall paper quality, it may be inferred that the quality of their writing, at the lowest level, may be improved as well, especially if Agile techniques like pair programming and rubber duck debugging are used.

8.3. LIMITATIONS AND THREATS TO VALIDITY

The primary limitation of the present work rests with the conducted SLR, as it is not feasible to ensure actual systematicism in surveying large quantities of literature (Boell & Cecez-Kecmanovic, 2016), although a protocol like PRISMA-P hopes to do so. Notwithstanding, the search protocol is clearly defined, so the SLR, while unable to claim absolutely perfect systematicism, is indeed perfectly replicable.

Regarding the survey employed, the sample size was limited to 14 participants, which very likely reduces the statistical significance of the findings. Moreover, all those surveyed were experienced software engineers or disciplined in a related field, e.g., computer science or systems engineering, save for two in business/psychology. The use of self-reported, free response answers in the survey may also introduce biases in response. Also, the multiple-choice questions used pre-determined answer "bins", typically followed by a catchall "Other (please specify)" answer, that may engender bias or limitation of respondent answering. The purpose of this survey was to establish if computer-related professionals, all of whom have at least a passing understanding of Agile, employ any Agile-like techniques when writing their scientific papers. Future research should address the aforementioned limitations by drawing upon larger and more diverse populations of scientific writers, principally ensuring that the members of various disciplines are included, e.g., biomedical researchers, social science researchers, etc.

So saying, a notable threat to the validity of the proposal of the present paper is the possibility that applying Agile methods to a field outside of software development may not be so easily done, as has been noted for other fields such as manufacturing (Niederman, Lechler, & Petit, 2018). In any case, as Agile software development, and modern scientific paper writing, are both performed in small iterations, and digitally with computers, it is posited that Agile methods can be applied to scientific paper writing. Future research in this regard should invariably include empirical studies of Agile-driven scientific paper development, measured, even if qualitatively, against controls, e.g., traditional scientific paper writing practices. The definition of an Agile writing method, for instance, based on Scrum, would ideally be informed by the mappings presented in this paper; such future work could follow the experimental technique demonstrated by Cruz et al., in their evaluation of Scrum as a scientific writing method (Cruz, et al., 2021).

Additionally, the present paper is necessarily limited in its scope, as other project development frameworks are not considered. This is due, primarily, to the author's education as a software engineer and longstanding involvement with various Agile software development teams. Future researchers would be prudent to consider other project development frameworks aside from Agile, to be potentially applied to the act of scientific paper writing. E.g., the Project Management Body of Knowledge (PMBoK) has been compared to Agile (Fitsilis, 2008), and the Projects in Controlled Environments (PRINCE2) method has been compared to the PMBoK (Wideman, 2002).

9. Conclusion

It is the sincerest hope of the author that, through the discourse potentiated by so novel a work, researchers across the diverse domains of scientific interest may begin to think of the scientific writing process not only as the reporting of data and results, or the hypothesizing of an idea, but as a drawing of order out of chaos; as an iterative, often collaborative, development of encapsulated systems with directed purposes, namely, to faithfully report theory and experimentation and further the web of interconnected knowledge that is Internet-bound science. As the research accumulated in the present work indicates, there are clear corollaries between Agile software development and scientific paper writing, inasmuch as the latter results in the provision of small systems iterated upon with small teams, having an objective purpose and an eventual release to the public. Therefore, it is posited that Agile methods may find great efficacy in the context of scientific paper writing.

Specifically, the systems development Agile variant, Scrum, is most noteworthy in the present context. Every stage of the canonical software development lifecycle is delineated with respect to scientific writing; and particular Agile activities, e.g., pair programming and Kanban boards, are transmuted into the application area of scientific writing. Backing the supposition that Agile may be beneficial to scientific writers are both a systematic literature review on scientific writing methods, and a survey of fourteen engineering writing professionals. The latter revealed that scientific writing is regarded by most as an iterative, collaborative, adaptive process, and, moreover, that some writers already employ Agile techniques, e.g., Kanban boards and write-review-revise cycles.

The fulfillment of Agile Academia remains entirely feasible. Future researchers would be wise to evaluate Agile practices on scientific writing with empirical case studies.

10. Acknowledgements

This work is partially supported by the National Science Foundation under Grant No. 1920780.

References

Aberdour, M. (2007). Achieving quality in open-source software. *IEEE Software, 24*(1), 58-64.

Abrahamsson, P., Salo, O., Ronkainen, J., & Warsta, J. (2017). Agile software development methods: Review and analysis. *arXiv preprint arXiv:1709.08439*.

Alaidaros, H., Omar, M., & Romli, R. (2021). The state of the art of agile kanban method: challenges and opportunities. *Independent Journal of Management & Production, 12*(8), 2535-2550.

Alexandrov, A. V., & Hennerici, M. G. (2007). Writing good abstracts. *Cerebrovascular Diseases, 23*(4), 256-259.

Al-Saqqa, S., Sawalha, S., & AbdelNabi, H. (2020). Agile software development: Methodologies and trends. *International Journal of Interactive Mobile Technologies, 14*(11).

Anthony, L. (1999). Writing research article introductions in software engineering: How accurate is a standard model? *IEEE transactions on Professional Communication, 42*(1), 38-46.

Bahadoran, Z., Mirmiran, P., Kashfi, K., & Ghasemi, A. (2019). The Principles of Biomedical Scientific Writing: Title. *International Journal of Endocrinology and Metabolism, 17*(4).

Bahadoran, Z., Mirmiran, P., Kashfi, K., & Ghasemi, A. (2020). The principles of biomedical scientific writing: citation. *International Journal of Endocrinology and Metabolism, 18*(2).

Bahadoran, Z., Mirmiran, P., Zadeh-Vakili, A., Hosseinpanah, F., & Ghasemi, A. (2019). The principles of biomedical scientific writing: results. *International Journal of Endocrinology and Metabolism, 17*(2).

Bahamdain, S. S. (2015). Open source software (OSS) quality assurance: A survey paper. *Procedia Computer Science, 56*, 459-464.

Barstown, B., Fazio, L., Schunn, C., & Ashley, K. (2017). Experimental evidence for diagramming benefits in science writing. *Instructional Science, 45*, 537-556.

Basu, A. (2023). How to use Curvenote for Writing Papers. *Qeios*.

Bavota, G., & Russo, B. (2015). Four eyes are better than two: On the impact of code reviews on software quality. *2015 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, 81-90.

Beck, K., Beedle, M., Bennekum, A. V., Cockburn, A., Cunningham, W., Fowler, M., . . . Thomas, D. (2001). Manifesto for agile software development.

Benington, H. D. (1956). Production of Large Computer Programs. *Proceedings of the ONR Symposium on Advanced Programming Methods for Digital Computers*, 15-27.

Bizzoni, Y., Degaetano-Ortlieb, S., Fankhauser, P., & Teich, E. (2020). Linguistic variation and change in 250 years of English scientific writing: A data-driven approach. *Frontiers in Artificial Intelligence, 3*(73).

Blackwell, J., & Martin, J. (2011). A scientific approach to scientific writing.

Boell, S. K., & Cecez-Kecmanovic, D. (2016). On being ‘systematic’ in literature reviews. *Formulating Research Methods for Information Systems*, 48-78.

Booth, V. (1993). Communicating in science: writing a scientific paper and speaking at scientific meetings. *Cambridge University Press*.

Bretthauer, D. (2001). Open source software: A history. *UConn Library*.

Bryant, S., Romero, P., & du Boulay, B. (2006). The collaborative nature of pair programming. *International Conference on Extreme Programming and Agile Processes in Software Engineering*, 53-64.

Cargill, M., & O'Connor, P. (2021). Writing scientific research articles: Strategy and steps. *John Wiley & Sons*.

Carrío-Pastor, M. L., & Mestre-Mestre, E. M. (2014). Lexical errors in second language scientific writing: Some conceptual implications. *International journal of English studies, 14*(1), 97-108.

Clabough, E. B., & Clabough, S. W. (2016). Using Rubrics as a Scientific Writing Instructional Method in Early Stage Undergraduate Neuroscience Study. *Journal of Undergraduate Neuroscience Education, 15*(1), 85-93.

Cockburn, A., & Williams, L. (2000). The costs and benefits of pair programming. *Extreme programming examined, 8*, 223-247.

Cohen, D., Lindvall, M., & Costa, P. (2003). Agile software development. *Dacs Soar Report, 11*.

Cohn, M. (2004). *User stories applied: For agile software development.* Addison-Wesley Professional.

Corona, E., & Pani, F. E. (2013). A review of lean-kanban approaches in the software development. *WSEAS transactions on information science and applications, 10*(1), 1-13.

Cruz, L. M., Cosgaya, H. M., Canul, R. C., Alvarez, D. C., Ortegón, J. R., & Vega, Y. G. (2021). Scrum for life, methodology applied in the scientific production of the teacher: "Writing a

Paper". *2021 16th Iberian Conference on Information Systems and Technologies (CISTI)*, 1-6.

Cuschieri, S., Grech, V., & Savona-Ventura, C. (2018). WASP (Write a Scientific Paper): How to write a scientific thesis. *Early human development, 127*, 101-105.

Deemer, P., Benefield, G., Larman, C., & Vodde, B. (2010). The scrum primer.

Derntl, M. (2014). Basics of research paper writing and publishing. *International Journal of Technology Enhanced Learning, 6*(2), 105-123.

Dingsøyr, T., Nerur, S., Balijepally, V., & Moe, N. B. (2012). A decade of agile methodologies: Towards explaining agile software development. *Journal of systems and software, 85*(6), 1213-1221.

Ebel, H. F., Bliefert, C., & Russey, W. E. (2004). The art of scientific writing: from student reports to professional publications in chemistry and related fields. *John Wiley & Sons*.

Fitsilis, P. (2008). Comparing PMBOK and Agile Project Management software development processes. *Advances in Computer and Information Sciences and Engineering*, 378-383.

Galesic, M., & Bosnjak, M. (2009). Effects of questionnaire length on participation and indicators of response quality in a web survey. *Public opinion quarterly, 73*(2), 349-360.

Ghasemi, A., Bahadoran, Z., Zadeh-Vakili, A., Montazeri, S. A., & Hosseinpanah, F. (2019). The principles of biomedical scientific writing: Materials and methods. *International Journal of Endocrinology and Metabolism, 17*(1).

Gilgun, J. F. (2005). "Grab" and good science: Writing up the results of qualitative research. *Qualitative health research, 15*(2), 256-262.

Goldbort, R. (2006). Writing for science. *Yale university press*.

Grogan, K. E. (2021). Writing Science. *Bulletin of the Ecological Society of America, 102*(1), 1-8.

Hamby, S. (2015). On scientific writing in the information era: Tailoring papers for Internet searching and other 21st century realities. *Psychology of Violence, 5*(2), 103-111.

Hebb, D. O., & Bindra, D. (1952). Scientific writing and the general problem of communication. *American Psychologist, 7*(10).

Holmes, F. L. (1987). Scientific Writing and Scientific Discovery. *Isis, 78*(2), 220-235.

Holstein, S. E., Steinmetz, K. R., & Miles, J. D. (2015). Teaching science writing in an introductory lab course. *Journal of Undergraduate Neuroscience Education, 13*(2).

Janzen, D. S., & Saiedian, H. (2006). On the influence of test-driven development on software design. *19th Conference on Software Engineering Education & Training (CSEET'06)*, 141-148.

Jerde, C. L., & Taper, M. L. (2004). Preparing Undergraduates for Professional Writing: Evidence Supporting the Benefits of Scientific Writing within the Biology Curriculum. *Journal of College Science Teaching, 33*(7), 34-37.

Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., . . . Fung, P. (2023). Survey of hallucination in natural language generation. *ACM Computing Surveys, 55*(12), 1-38.

Kail, R. V. (2018). Scientific writing for psychology: Lessons in clarity and style. *SAGE Publications*.

Kaplan, R. B., & Grabe, W. (1991). The fiction in science writing. In *Subject-Oriented Texts.* Berlin: Walter de Gruyter & Co.

Katz, M. J. (2009). *From research to manuscript: A guide to scientific writing.* Springer Science & Business Media.

Kitchenham, B., & Charters, S. (2007). Guidelines for performing systematic literature reviews in software engineering.

Kniberg, H., & Skarin, M. (2010). *Kanban and Scrum-making the most of both.* C4Media.

Lalumière, M. L. (1993). Increasing the precision of citations in scientific writing. *American Psychologist, 48*(8), 913.

Lareo Martín, I., & Reyes, A. M. (2007). Scientific writing: following Robert Boyle's principles in experimental essays-1704 and 1998. *Revista alicantina de estudios ingleses, 20*, 119-137.

Lebrun, J.-L., & Lebrun, J. (2021). Scientific writing 3.0: a reader and writer's guide.

Li, G., Jin, Y., Mbuagbaw, L., Dolovich, L., Adachi, J. D., Levine, M. A., . . . Thabane, L. (2018). Enhancing research publications and advancing scientific writing in health research collaborations: sharing lessons learnt from the trenches. *Journal of Multidisciplinary Healthcare, 11*, 245-254.

Lindsay, D. (2010). *Scientific writing = thinking in words.* Australia: Csiro Publishing.

Lourens, A. (2007). Scientific writing skills: Guidelines for writing theses and dissertations. *African Sun Media*.

Lu, C., Bu, Y., Wang, J., Ding, Y., Torvik, V., Schnaars, M., & Zhang, C. (2019). Examining scientific writing styles from the perspective of linguistic complexity. *Journal of the Association for Information Science and Technology, 70*(5), 462-475.

Madkur, A. (2013). GRAMMATICAL ANALYSIS ON THE ABSTRACT OF SCIENTIFIC WRITING. *Pedagogy: Journal of English Language Teaching, 1*(1).

Maiorana, F. A., & Mayer, H. F. (2018). How to avoid common errors in writing scientific manuscripts. *European Journal of Plastic Surgery, 41*, 489-494.

Matthews, J. R., & Matthews, R. W. (2014). Successful Scientific Writing: a step-by-step guide for the biological and medical sciences.

McIntosh, S., Kamei, Y., Adams, B., & Hassan, A. E. (2016). An empirical study of the impact of modern code review practices on software quality. *Empirical Software Engineering, 21*, 2146-2189.

Moessner, L. (2009). The influence of the Royal Society on 17th-century scientific writing. *ICAME journal, 33*, 65-87.

Moher, D., Shamseer, L., Clarke, M., Ghersi, A. L., Petticrew, M., Shekelle, P., & Stewart, L. A. (2015). Preferred reporting items for systematic review and meta-analysis protocols (PRISMA-P) 2015 statement. *Systematic reviews, 4*(1), 1-9.

Moskovits, C. (2019). Text recycling in scientific writing. *Science and engineering ethics, 25*(3), 813-851.

Nair, P. R., & Nair, V. D. (2014). Scientific writing and communication in agriculture and natural resources. *15530*.

Niederman, F., Lechler, T., & Petit, Y. (2018). A research agenda for extending agile practices in software development and additional task domains. *Project Management Journal, 49*(6), 3-17.

Niemitz, J. W., & Potter Jr., N. (1991). The scientific method and writing in introductory landscape-development laboratories. *Journal of Geological Education, 39*(3), 190-195.

O'Connor, T. R., & Holmquist, G. P. (2009). Algorithm for writing a scientific manuscript. *Biochemistry and Molecular Biology Education, 37*(6), 344-348.

Oprins, R. J., Frijns, H. A., & Stettina, C. J. (2019). Evolution of scrum transcending business domains and the future of agile project management. *Agile Processes in Software Engineering and Extreme Programming: 20th International Conference, XP 2019, 20*, 244-259.

Pacheco, C., Garcia, I., & Reyes, M. (2018). Requirements elicitation techniques: a systematic literature review based on the maturity of the techniques. *IET Software, 12*(4), 365-378.

Patterson, E. W. (2001). Structuring the composition process in scientific writing. *International journal of science education, 23*(1), 1-16.

Permana, P. A. (2015). Scrum method implementation in a software development project management. *International Journal of Advanced Computer Science and Applications, 6*(9), 198-204.

Phadtare, A., Bahmani, A., Shah, A., & Pietrobon, R. (2009). Scientific writing: a randomized controlled trial comparing standard and on-line instruction. *BMC Medical Education, 9*(1), 1-9.

Ping Alvin, L. (2014). The passive voice in scientific writing. The current norm in science journals. *Journal of Science Communication, 13*(1).

Pollock, A. V., Evans, M. E., Wiggin, N. J., & Balch, C. M. (1991). Writing Your First Scientific Paper. In *Principles and Practice of Research: Strategies for Surgical Investigators* (pp. 396-403). New York: Springer US.

Procko, T. T., Davidoff, A., Elvira, T., & Ochoa, O. (2023). Towards Improved Scientific Knowledge Proliferation: Leveraging Large Language Models on the Traditional Scientific Writing Workflow. *Available at SSRN 4594836*.

Ramos, D. B., Ramos, I. M., Viana, W. d., Silva, G., & de Oliveira, E. (2016). On the use of Scrum for the management of researchoriented projects. *Nuevas Ideas en Informática Educativa, 12*, 589-594.

Reddy, A. (2015). *The Scrumban [R]Evolution: Getting the Most Out of Agile, Scrum, and Lean Kanban.* Addison-Wesley Professional.

Reed, C., Walton, D., & Macagno, F. (2007). Argument diagramming in logic, law and artificial intelligence. *The Knowledge Engineering Review, 22*(1), 87-109.

Regan, M., & Pietrobon, R. (2010). A conceptual framework for scientific writing in nursing. *Journal of Nursing Education, 49*(8), 437-443.

Rosenberg, J., Burcharth, J., Pommergaard, H. C., & Danielsen, A. K. (2013). Mind-to-paper is an effective method for scientific writing. *Danish Medical Journal, 60*(3).

Royce, W. W. (1970). MANAGING THE DEVELOPMENT OF LARGE SOFTWARE SYSTEMS . *Proceedings, IEEE WESCON*, 1-9.

Ruparelia, N. B. (2010). Software Development Lifecycle Models. *ACM SIGSOFT Software Engineering Notes, 35*(3), 8-13.

Sanders, E. B.-N., & Stappers, P. J. (2008). Co-creation and the new landscapes of design. *Co-design, 4*(1), 5-18.

Sayer, E. J. (2019). The essentials of effective scientific writing–A revised alternative guide for authors. *Functional Ecology, 33*(9), 1576-1579.

Schickore, J. (2008). Doing science, writing science. *Philosophy of Science, 75*(3), 323-343.

Schulte, B. A. (2003). Scientific writing & the scientific method: Parallel “hourglass” structure in form & content. *The American Biology Teacher, 65*(8), 591-594.

Schwaber, K. (1997). Scrum development process. *Business Object Design and Implementation: OOPSLA’95 Workshop Proceedings*.

Schwaber, K., & Sutherland, J. (2020). *The 2020 Scrum Guide (TM)*. Retrieved December 2023, from Scrum Guides: https://scrumguides.org/scrum-guide.html

Sedano, T., Ralph, P., & Péraire, C. (2019). The product backlog. *2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)*, 200-211.

Sivess, V. J., & Curtis, M. C. (1993). The software life-cycle model: An alternative perspective. *Transactions on Information and Communications Technologies, 4*, 345-360.

Sollaci, L. B., & Pereira, M. G. (2004). The introduction, methods, results, and discussion (IMRAD) structure: a fifty-year survey. *Journal of the medical library association, 92*(3), 364-371.

Stewart, J. C., DeCusatis, C. S., Kidder, K., Massi, J. R., & Anne, K. M. (2009). Evaluating agile principles in active and cooperative learning. *Proceedings of Student-Faculty Research Day*.

Subramaniam, K. (2010). Integrating writing frames into inquiry-based instruction. *Science Educator, 19*(2), 31-34.

Svoboda, B. F. (2021). Adapting Agile for Research Laboratory. *Masaryk University*.

The Open Source Definition (Annotated). (n.d.). Retrieved December 2023, from Open Source Initiative: https://opensource.org/osd-annotated

Thomas, C. G. (2021). Research methodology and scientific writing.

Twidale, M., & Hansen, P. (2019). Agile Research. *First Monday*.

Volpato, G. L. (2011). The logic of scientific writing. *Revista de Sistemas de Informação da FSMA, 7*, 2-5.

Weissberg, R., & Buker, S. (1990). Writing up research.

Wideman, R. M. (2002). Comparing PRINCE2® with PMBoK®. *AEW Services*, 13-16.

Wohlin, C. (2014). Guidelines for snowballing in systematic literature studies and a replication in software engineering. *18th International Conference on Evaluation and Assessment in Software Engineering*, 1-10.

Woodford, F. P. (1967). Sounder Thinking Through Clearer Writing: A graduate course on scientific writing can, if appropriately designed, strengthen scientific thinking. *Science, 156*(3776), 743-745.

Worsley, D., & Mayer, B. (1989). The Art of Science Writing. *Teachers and Writers Collaborative*.

Wu, J. (2011). Improving the writing of research papers: IMRAD and beyond. *Landscape Ecology, 26*, 1345-1349.

Yang, J. T. (1995). An Outline Of Scientific Writing: For Researchers With English As A Foreign Language. *World Scientific Publishing Company*.

You, J., Kang, S., Kim, J., & Noh, T. (2013). An investigation of students' science writing processes using think-aloud method. *Journal of the Korean Association for Science Education, 33*(5), 881-892.

Appendix A: Survey Questions

Demographic Questions

1. How many years of experience do you have in publishing scientific papers?

2. What is the title of the bachelor's degree you graduated with?

3. What is your current primary field of research?

Multiple Choice Questions – Basic

1. Have you ever taken an academic course specific to scientific writing?
   - Yes
   - No

2. Which software/tools do you primarily use for drafting your scientific papers?
   - Microsoft Word
   - LaTeX
   - Google Docs
   - Other (please specify)

3. How do you usually handle data visualization and chart creation in your papers?
   - I create charts and graphs manually
   - I use specialized software or tools
   - I delegate this task to others
   - Other (please specify)

4. How much time do you usually spend on revising and editing a paper before submission?
   - Less than a week
   - 1-2 weeks
   - 3-4 weeks
   - More than one month
   - Other (please specify)

5. How do you select a publication outlet?
   - General Internet search
   - Consulting colleagues
   - Personal experience/references
   - Other (please specify)

6. Do you use preprint servers for early dissemination of your research?
   - Yes
   - No

- Depends on paper
- Other (please specify)

7. Which factor is most important to you when selecting a journal for submission?
   - Impact factor
   - Speed of publication
   - Open access
   - Relevance to field
   - Other (please specify)

Multiple Choice Questions – Processual

1. How do you typically begin writing a scientific paper?
   - With a clear hypothesis in mind
   - With a general topic or idea
   - With data or results ready
   - With an anticipated publication outlet
   - Other (please specify)

2. When writing a scientific paper, do you progress linearly, or do you fill in sections as you progress in the understanding of your topic?
   - Strictly linear progression (start of paper to end of paper)
   - Mostly linear progression
   - I fill in sections as desired until completed
   - Depends on paper
   - Other (please specify)

3. How do you approach collaboration with co-authors?
   - We co-write in real time using collaborative tools
   - We divide sections and combine later
   - We discuss ideas verbally and one person writes
   - Other (please specify)

4. How do you organize tasks during the writing of a paper?
   - With a to-do checklist
   - With notes reminding me
   - I just remember what I need to do
   - Other (please specify)

5. How do you approach task delegation when collaborating on a paper?
   - Assign sections or tasks based on expertise
   - Assign sections or tasks based on availability
   - Rotate tasks to ensure shared understanding
   - Other (please specify)

6. How do you usually prioritize sections or elements in your papers?
   - By importance to the paper's main argument or idea
   - By ease of completion
   - By data availability
   - Other (please specify)

7. When do you typically consider a section of your paper "complete"?
   - When it is written and formatted
   - When it has been reviewed by co-authors or peers
   - When it aligns well with the paper's overall narrative
   - Other (please specify)

8. How frequently do you revisit and revise sections after considering them "complete"
   - Frequently
   - Occasionally
   - Rarely
   - Never

9. How do you handle unexpected changes or challenges, e.g., new data, conflicting results, etc., during your writing process?
   - Continuously, from multiple sources
   - At specific milestones, e.g., after completing each section
   - Only after completing the first draft
   - Other (please specify)

10. Do you set specific milestones or deadlines for different sections or stages of your paper?
    - Yes
    - No
    - Other (please specify)

11. How do you balance writing the paper with other responsibilities?
    - I allocate specific time blocks for writing
    - I write when inspiration strikes
    - I prioritize writing closer to deadlines
    - Other (please specify)

12. Do you incorporate any iterative or cyclical methods in your writing process (e.g., Write-Review-Revise cycles)?
    - Yes
    - No
    - Maybe
    - Other (please specify)

FREE RESPONSE QUESTIONS

1. Can you describe any challenges you frequently encounter during the writing and publication process for scientific papers?

2. What strategies or practices have you found most effective in successfully publishing your work?

3. If you could describe your method for writing a scientific paper in one sentence, what would that sentence be?

4. How has your approach to writing and publishing changed over the course of your career?