From Papers to Progress: Rethinking Knowledge Accumulation in Software Engineering

Anonymous Author(s)

Abstract

Software engineering research has experienced sustained growth in both output and participation over the past decades. Yet concerns persist about the field's ability to accumulate, integrate, and reuse knowledge in ways that support long-term progress. To better understand how the community itself perceives these challenges, we analyze responses from the ICSE 2026 Future of Software Engineering pre-survey, which captures perspectives from a globally distributed and highly experienced set of researchers. Our analysis reveals a tension between increasing research productivity and the limited mechanisms available for synthesizing results, tracking evolving claims, and supporting cumulative understanding over time.

Building on these observations, we diagnose four interrelated structural breakdowns: papers function as isolated knowledge units with claims embedded in prose; context and provenance are lost as knowledge moves through the publication pipeline; claims evolve without systematic tracking; and incentive structures favor novelty over consolidation. We argue that addressing these barriers requires rethinking the fundamental properties of research artifacts.

We articulate four technology-agnostic principles for future research artifacts: structured and interpretable representations of claims and evidence; inspectable and provenance-aware documentation of methodological decisions; long-lived and reusable substrates that evolve beyond publication; and governance mechanisms that align individual incentives with collective knowledge-building goals. We discuss implications for research practice, publication norms, and community infrastructure, positioning FOSE as a venue for experimenting with alternative artifact designs that support cumulative scientific progress.

CCS Concepts

• **Software and its engineering** → **Software creation and management**; **Empirical software validation**; • **General and reference** → *Empirical studies.*

Keywords

knowledge accumulation, research artifacts, software engineering, cumulative progress, research infrastructure

ACM Reference Format:

Anonymous Author(s). 2026. From Papers to Progress: Rethinking Knowledge Accumulation in Software Engineering. In . ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

Conference'17, Washington, DC, USA

2026. ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM

https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

Software engineering research has experienced remarkable growth over the past decades. Conference submissions have increased, publication venues have proliferated, and the community has become truly global. By many measures, the field is thriving: researchers are productive, techniques are advancing, and new areas of inquiry continue to emerge. Yet beneath this surface of activity lies a persistent challenge: the difficulty of building cumulative knowledge that connects, consolidates, and extends prior work in ways that support long-term scientific progress.

This challenge is not new, nor is it unique to software engineering. Across many scientific disciplines, researchers have observed that increasing publication rates do not automatically translate into deeper understanding or more integrated knowledge [8, 11]. In software engineering specifically, concerns about fragmentation, replication, and the ability to synthesize results across studies have been raised repeatedly [20, 21]. Despite these concerns, the structural factors that limit cumulative progress—and the properties that future research artifacts might need to address them—remain underexplored.

This paper offers a community-informed perspective on knowledge accumulation challenges in software engineering research. We begin by examining responses to the ICSE 2026 Future of Software Engineering track pre-survey, which reveals a mature, productive, and globally distributed community. This demographic profile is significant: if knowledge accumulation remains limited despite high levels of expertise, experience, and output, then the barriers to progress are unlikely to be individual. They are structural—embedded in how research artifacts are designed, how knowledge is represented, and how incentives shape research practice.

Building on this observation, we diagnose four interrelated structural breakdowns that constrain cumulative knowledge building:

1. **Papers as isolated knowledge units:** Claims and evidence are embedded in prose, making direct comparison and synthesis difficult as the literature grows.
2. **Loss of context and provenance:** Motivation, assumptions, and methodological decisions fade over time, increasing the cost of building on prior work.
3. **Claims evolve without tracking:** Refinements, contradictions, and relationships between claims are documented in prose but not systematically tracked or resolved.
4. **Incentive structures favor novelty over accumulation:** Publication systems reward new contributions while undervaluing replication, consolidation, and infrastructure work.

These breakdowns are not independent—they reinforce one another, creating a system in which knowledge accumulates slowly and inefficiently despite the community's best efforts.

In response, we articulate four principles for designing research artifacts that better support cumulative progress:

(1) **Structured and interpretable:** Artifacts should make claims, evidence, and context explicit and directly accessible.
(2) **Inspectable and provenance-aware:** Artifacts should preserve the full provenance of results and make it transparent.
(3) **Long-lived and reusable:** Artifacts should support evolution and reuse beyond the moment of publication.
(4) **Governed with human oversight:** Artifacts and infrastructures should be managed by community processes that ensure quality, integrity, and alignment with collective goals.

These principles are intentionally technology-agnostic. We do not prescribe specific tools, platforms, or workflows. Instead, we describe properties that any system aiming to support knowledge accumulation should strive for, leaving room for diverse implementations across domains and research communities.

1.1 Contributions

This paper makes the following contributions:

- A community-informed diagnosis of structural barriers to knowledge accumulation in software engineering, grounded in evidence from the ICSE 2026 FOSE pre-survey.
- An articulation of four interrelated breakdowns that explain why cumulative progress remains limited despite high levels of expertise and productivity.
- Four principles for future research artifacts that address these breakdowns, presented as technology-agnostic guidelines rather than prescriptive solutions.
- A discussion of implications for research practice, publication norms, community infrastructure, and the role of FOSE as an experimentation venue for alternative artifact types.

1.2 Scope and Positioning

This is not a tooling paper. We do not present a new system, platform, or methodology. Nor do we claim to have solved the challenges we diagnose. Instead, we offer a reflective, agenda-setting perspective appropriate for the FOSE track: a synthesis of community observations, a diagnosis of structural limitations, and a set of principles to guide future work. Our goal is to articulate why cumulative knowledge progress remains constrained and what properties future research artifacts might need to address these constraints.

We position this work as a call to rethink the fundamental design of research artifacts—not because current practices are wrong, but because they were optimized for different goals (dissemination, novelty, individual credit) and may be insufficient for the cumulative, long-term knowledge building that scientific progress requires.

1.3 Paper Organization

The remainder of this paper is organized as follows. Section 2 situates our work within related perspectives on knowledge accumulation, reproducibility, and research infrastructure. Section 3 presents findings from the ICSE 2026 FOSE pre-survey, establishing the community profile that motivates our structural diagnosis. Section 4 diagnoses four interrelated breakdowns that limit cumulative progress. Section 5 articulates four principles for future research artifacts. Section 6 discusses implications for research practice, publication, and infrastructure. Section 7 acknowledges limitations and open questions. Section 8 concludes with reflections on the path forward.

2 Background and Related Perspectives

The challenges of cumulative knowledge building are not unique to software engineering, nor are they new. Across scientific disciplines, researchers and meta-researchers have documented tensions between publication growth and knowledge synthesis, raised concerns about reproducibility and replication, and called for better infrastructure to support long-term scientific progress. This section situates our work within these broader conversations.

2.1 Knowledge Accumulation in Science

The challenge of accumulating knowledge has been a recurring theme in philosophy of science and meta-research. Kuhn's notion of paradigm shifts highlighted periods of consolidation and synthesis [16]. More recently, concerns about the "reproducibility crisis" have drawn attention to the difficulty of verifying and building on prior results [3, 11]. These discussions emphasize that scientific progress depends not only on producing new findings but also on mechanisms for validating, integrating, and reusing prior work.

In software engineering, similar concerns have been raised. Studies have documented challenges in replication [10, 14], challenges in synthesizing results across empirical studies [15], and the difficulty of comparing techniques when methodologies and datasets vary [4]. Researchers have also noted that the field's rapid growth, while positive in many ways, has made it harder to maintain shared understanding and cumulative building across subfields [21].

2.2 Research Infrastructure and Artifacts

A growing body of work has explored how research infrastructure can support cumulative progress. Cyberinfrastructure initiatives [2] have emphasized the need for shared computational resources, data repositories, and collaborative platforms. In software engineering specifically, efforts like artifact evaluation tracks [10], benchmark repositories [6], and replication packages [9] aim to make research more transparent and reusable.

Beyond software engineering, knowledge graph initiatives in other scientific domains offer relevant examples. The Open Research Knowledge Graph (ORKG) [13] structures contributions from scientific papers as semantic triples, enabling comparison and aggregation across studies. Similar efforts in scientific knowledge extraction [17] and academic graph construction [24] aim to represent scholarly knowledge in machine-readable, queryable forms. These systems demonstrate the potential of structured representations to support synthesis, but they also highlight challenges: creating and maintaining such infrastructures requires sustained effort, community coordination, and alignment of incentives [5].

2.3 Incentive Structures and Community Practices

Several researchers have examined how incentive structures shape scientific practice. Publication pressures, the emphasis on novelty in review processes, and the undervaluation of negative results and replication have all been identified as barriers to cumulative progress [18, 22]. In software engineering, Shull et al. [20] discussed the role of community infrastructure and incentives in supporting empirical research. More recent work has called for recognizing contributions to datasets, benchmarks, and infrastructure as valuable scholarly outputs [1].

The tension between individual incentives (publishing novel results to advance one's career) and collective needs (consolidating knowledge, maintaining infrastructure) is a well-recognized collective action problem [19]. Addressing it requires not only technical solutions but also changes in how contributions are evaluated, how credit is assigned, and how community resources are governed.

2.4 Meta-Research and Science of Science

The emerging field of meta-research, or the "science of science," studies the processes and structures that shape scientific work [8, 12]. This research has documented patterns in citation networks, collaboration structures, and the lifecycle of scientific ideas. It has also examined how publication systems, peer review, and funding mechanisms influence what research gets done and how it is communicated [23].

One relevant finding is that scientific communities differ in how well they support cumulative building. Fields with well-established benchmarks, standardized evaluation protocols, and shared datasets often exhibit faster cumulative progress than fields where each study uses custom methodologies [7]. This suggests that infrastructure for comparison and reuse can have measurable impacts on the pace of knowledge accumulation.

2.5 Positioning This Work

Our work builds on these perspectives but focuses specifically on structural factors that limit cumulative progress in software engineering research. We differ from prior work in three ways:

First, we ground our analysis in community perspectives gathered through the ICSE 2026 FOSE pre-survey. Rather than relying solely on bibliometric analysis or case studies, we use survey data to establish that the community itself recognizes accumulation challenges and that these challenges persist despite high levels of expertise and productivity.

Second, we provide a structured diagnosis of four interrelated breakdowns, connecting individual challenges (isolated papers, lost context, untracked claims, misaligned incentives) into a coherent account of why progress remains constrained.

Third, we articulate technology-agnostic principles for future research artifacts rather than proposing specific tools or platforms. Our goal is not to advocate for any particular system but to describe properties that any infrastructure aiming to support cumulative progress should strive for. This framing is intentionally broad, allowing for diverse implementations across different subfields and research communities.

We do not claim that these challenges are unique to software engineering, nor that solutions developed in other fields are irrelevant. Instead, we offer a community-informed synthesis tailored to the software engineering context, with the hope that the principles we articulate can inform both local efforts within the field and broader conversations about the future of scientific infrastructure.

Table 1: FOSE 2026 pre-survey respondent demographics (n=280)

Characteristic	Count	Percentage
Years in SE Research Community		
0–1 years	12	4.3%
2–3 years	26	9.3%
4–5 years	29	10.4%
6–10 years	50	17.9%
11–20 years	82	29.3%
21+ years	78	27.9%
Papers Authored (Past 3 Years)		
0 papers	8	2.9%
1–3 papers	54	19.3%
4–10 papers	91	32.5%
11–20 papers	73	26.1%
21–40 papers	32	11.4%
41+ papers	19	6.8%
Geographic Distribution		
Europe	139	49.6%
North America	70	25.0%
Asia	41	14.6%
South America	14	5.0%
Oceania	11	3.9%
Middle East	2	0.7%

3 Community Signals from the ICSE 2026 FOSE Pre-Survey

To ground our discussion in community perspectives, we draw on responses from the ICSE 2026 Future of Software Engineering track pre-survey. This survey was designed to understand how members of the software engineering research community perceive current challenges and opportunities in the field. We received 280 completed responses from researchers representing diverse roles, experience levels, and geographic regions.

Our purpose in examining this survey is not to conduct a comprehensive empirical study, but rather to establish context and legitimacy for the structural challenges we diagnose in this paper. The survey data reveals a community profile that is critical to our central argument: if knowledge accumulation remains limited despite significant expertise and participation, then the barriers must be structural rather than individual.

3.1 Community Profile

The survey respondents represent a mature, active, and globally distributed research community. Table 1 summarizes key demographic characteristics.

Experience. The majority of respondents (57.1%) have been involved in software engineering research for over a decade, with 29.3% reporting 11–20 years of experience and 27.9% reporting 21 or more years. This is not a community of newcomers struggling to establish research programs, but rather one populated by seasoned researchers who have observed and participated in the field's evolution over substantial periods.

Productivity. The community is actively engaged in producing research outputs. Nearly half of respondents (44.3%) reported authoring 11 or more paper submissions in the past three years alone—an average of at least 3–4 papers per year. An additional 32.5% reported 4–10 submissions. Only 2.9% reported zero submissions. This high level of productivity suggests a community that is not constrained by lack of effort or output capacity.

Global Reach. Respondents represent six geographic regions spanning Europe, North and South America, Asia, Oceania, and the Middle East. While Europe is most heavily represented (49.6%), North American (25.0%) and Asian (14.6%) researchers form substantial communities. This global distribution indicates that the challenges we discuss are not artifacts of a single research culture or institutional context, but rather reflect patterns that transcend geographic boundaries.

3.2 Implications for Knowledge Accumulation

The demographic profile of survey respondents carries a critical implication for our argument. The software engineering research community is not lacking in *expertise*—over half the respondents have more than a decade of experience. It is not lacking in *participation*—nearly half produce at least 11 paper submissions every three years. And it is not limited to a narrow geographic or cultural context—six regions are represented, with substantial participation from multiple continents.

If knowledge accumulation remains fragmented, if claims and evidence remain weakly linked, and if rediscovery costs remain high despite this level of experience, productivity, and global engagement, then the barriers to cumulative progress must be *structural*. They cannot be attributed to insufficient expertise, inadequate effort, or limited participation. Instead, they must arise from the ways in which research artifacts are produced, represented, and connected to one another over time.

3.3 Recognized Challenges

Beyond demographics, the survey responses reveal that the community itself recognizes systemic challenges. When asked what aspects of the software engineering research community do not work well, respondents frequently pointed to issues related to knowledge synthesis and cumulative building:

> *"Reviewing process: quality of reviews is not always good—too many papers get submitted (and resubmitted)."*

> *"We're not inclusive or sustainable enough insofar as the remote participation is lousy. I'd do away with all of our program committees and just do journal first for everything. I'd like to see people have more flexibility in which and whether to attend conferences and less waste and resubmission and re-review of papers."*

> *"Adopting new technologies. Let's say NLP community already adopted LLM based annotation with human supervision... yet in SE papers 30% of the justification goes to why this technology works well. It's like doing two research in a single paper."*

These comments hint at deeper structural issues: cycles of resubmission that fragment results across venues and versions, difficulties in building on prior work efficiently, and challenges in tracking how ideas evolve as they move through the publication pipeline. Respondents also expressed concerns about review quality, the stress of managing high submission volumes, and the difficulty of preparing mentees for a system that rewards novelty over consolidation.

3.4 Transition to Structural Diagnosis

The survey data establishes that the software engineering research community possesses significant expertise, maintains high productivity, and operates across global contexts. Yet respondents themselves recognize persistent challenges in how knowledge is synthesized, reviewed, and built upon. These observations set the stage for a more detailed examination of the structural barriers that limit cumulative knowledge accumulation—the subject of the next section.

4 Where Knowledge Accumulation Breaks Down

The survey data establish that the software engineering research community is experienced, productive, and globally distributed. Yet respondents recognize persistent challenges in synthesizing results, building on prior work, and tracking evolving ideas. If expertise and participation are not the bottleneck, what structural factors limit cumulative knowledge progress? In this section, we diagnose four interrelated breakdowns that arise from how research artifacts are currently produced, represented, and connected.

4.1 Papers as Isolated Knowledge Units

Research papers are designed as self-contained documents. They package claims, evidence, and context into narrative prose optimized for human reading and conference presentation. While this format has served the community well for disseminating new ideas, it creates challenges for cumulative building.

Claims are embedded in prose. A research paper may make multiple claims—about the performance of an approach, the generalizability of findings, or the limitations of prior work. But these claims are woven into paragraphs, interleaved with motivation, related work, and methodological details. Extracting and comparing claims across papers requires reading entire documents, interpreting language, and reconciling terminological differences. There is no structured representation that makes claims directly comparable or composable.

Evidence is not structured. The evidence supporting a claim—datasets, experimental configurations, evaluation metrics, statistical analyses—is typically described in text, presented in tables and figures, and sometimes supplemented with artifacts in repositories. But the linkage between claim and evidence remains implicit. A reader must infer which results support which claims, under what assumptions, and with what constraints. When attempting to build on prior work, researchers must reconstruct these connections from prose, often leading to misinterpretation or incomplete understanding.

Results are not directly reusable. Even when artifacts are shared, the results reported in a paper are fixed at the time of publication. If a dataset is updated, a metric is refined, or a baseline is improved, the original paper does not reflect these changes. Follow-on work must cite the paper, describe the difference, and report new numbers—but the original claim remains static in the literature. Over time, the paper becomes a historical artifact rather than a living record of what is currently known.

The consequence is fragmentation. Each paper is an island of knowledge, connected to others only through citations and prose references. Building cumulative understanding requires repeatedly reading, interpreting, and synthesizing isolated documents—a process that does not scale as the literature grows.

4.2 Loss of Context and Provenance

Research is inherently contextual. Every study makes assumptions, operates under constraints, and reflects decisions about what to prioritize and what to defer. Yet this context is often lost or obscured as knowledge moves through the publication pipeline.

Motivation and assumptions fade. A paper's introduction typically explains why a problem is important and what assumptions underlie the proposed approach. But as the work is cited and summarized by others, this context erodes. Later papers reference a result without fully capturing the conditions under which it holds. A finding that was carefully qualified in the original work becomes an unqualified fact in subsequent literature. Over time, the original motivation and boundary conditions become difficult to recover.

Methodological decisions are not preserved. Research involves countless decisions: which dataset to use, how to split training and test sets, which baselines to compare against, how to handle edge cases. Papers describe these decisions in methods sections, but the rationale—why this choice rather than another—is often implicit or omitted due to space constraints. When researchers attempt to replicate or extend the work, they must reverse-engineer these decisions, sometimes discovering that subtle choices had significant impact on results. The provenance of decisions is lost.

Evolution of ideas is not tracked. Ideas evolve as they move through the research process. An initial hypothesis may be refined, a technique may be adapted for a new domain, or a negative result may shift the direction of inquiry. But papers present polished, final versions of ideas. The path from initial conception to published result—the false starts, pivots, and accumulated insights—remains invisible. This evolutionary context could be valuable for others working on related problems, but it is rarely captured in the artifact of record.

When context and provenance are lost, later researchers must either accept prior work at face value or invest substantial effort reconstructing the reasoning behind it. This friction slows cumulative progress and increases the risk of misapplication or misinterpretation.

4.3 Claims Evolve Without Tracking

Scientific claims are not static. They are refined, qualified, contradicted, and superseded as new evidence accumulates. Yet the publication system provides limited mechanisms for tracking these changes over time.

Contradictions go unnoticed. The literature may contain conflicting claims: one paper reports that technique A outperforms technique B, while another finds the opposite. These contradictions may arise from differences in datasets, experimental setups, or evaluation criteria—but they are not systematically flagged or resolved. Researchers discovering these conflicts must investigate the causes themselves, often finding that subtle differences in methodology account for divergent results. Without structured tracking of claims and their relationships, contradictions remain latent in the literature.

Refinements are implicit. A follow-on paper may refine or qualify a prior claim: "Technique A works well for problem class X but not for problem class Y." This refinement is documented in the new paper, but the original claim remains unchanged. A reader consulting the older paper will not know that its findings have been refined unless they also discover the newer work. The relationship between the original claim and its refinement is captured only through citation and prose—mechanisms that do not scale to tracking fine-grained updates across a growing corpus.

Relationships between claims are unclear. Does one claim extend, contradict, depend on, or generalize another? These relationships are expressed in natural language: "building on [12]," "in contrast to [34]," "our results suggest that..." But extracting and formalizing these relationships requires interpretation. Without explicit, structured representations, it is difficult to trace how a claim has evolved through multiple papers, which claims support or refute each other, and which gaps remain to be addressed.

The result is that the state of knowledge remains fragmented and ambiguous. Researchers must perform extensive manual synthesis to determine what is currently believed, under what conditions, and with what confidence.

4.4 Incentive Structures Favor Novelty Over Accumulation

The final breakdown is not purely technical—it is embedded in the incentive structures that shape research practice. Publication venues, hiring and promotion criteria, and funding mechanisms reward novelty and originality. Consolidation, replication, and infrastructure work, while valuable for cumulative progress, are often undervalued.

Publication rewards new contributions. Conference and journal review processes prioritize papers that present new techniques, findings, or insights. Replication studies, negative results, and incremental refinements face higher bars for acceptance. A paper that carefully synthesizes prior work, resolves contradictions, or provides infrastructure for cumulative building may be seen as insufficiently novel, even if it would significantly advance the field's collective understanding. The system incentivizes producing new papers over consolidating existing knowledge.

Replication and consolidation are marginalized. Researchers who invest time in replicating prior work, curating datasets, or building shared infrastructure face opportunity costs. These efforts may not yield publications in top venues, and they may not be credited as highly in tenure and promotion reviews. The result is a collective action problem: everyone would benefit from better knowledge infrastructure, but individual researchers are disincentivized from contributing to it.

Infrastructure work is undervalued. Building systems that support cumulative knowledge—repositories, ontologies, quality standards, interoperability layers—requires sustained, long-term effort. Such work is often led by a small number of dedicated individuals or groups, without commensurate recognition or support. The broader community benefits from these infrastructures, but the incentives to create and maintain them remain weak relative to the incentives to publish novel research.

These structural incentives shape behavior at every level: what researchers choose to work on, what papers reviewers accept, and what contributions institutions reward. As long as novelty is privileged over accumulation, knowledge will remain fragmented, and progress will be constrained.

4.5 Interconnected Challenges

These four breakdowns are not independent. They reinforce one another. Isolated, prose-based knowledge units make it difficult to track evolving claims. Loss of context and provenance makes it harder to synthesize results across papers. Incentive structures discourage the sustained effort required to address these barriers. Together, they create a system in which knowledge accumulates slowly and inefficiently, despite the community's expertise, productivity, and good intentions.

The implication is clear: incremental fixes—better citation practices, more replications, improved repositories—are necessary but insufficient. Addressing these structural barriers requires rethinking the fundamental properties of research artifacts themselves. What would it take to design artifacts that support cumulative, interpretable, and reusable knowledge building? We turn to this question in the next section.

5 Rethinking Research Artifacts for Cumulative Progress

The structural barriers diagnosed in the previous section are not inevitable. They arise from design choices embedded in how research artifacts are currently produced and shared. If papers-as-documents create fragmentation, and if current incentive structures discourage consolidation, then addressing these challenges requires reconsidering the fundamental properties that research artifacts should have. In this section, we articulate four principles to guide the design of artifacts that support cumulative, interpretable, and reusable knowledge building.

These principles are intentionally abstract. We do not prescribe specific technologies, platforms, or workflows. Instead, we describe properties that any system, tool, or practice aiming to support knowledge accumulation should strive for. Concrete implementations will vary across domains and communities, but the underlying principles provide a shared foundation for evaluating and improving research infrastructure.

5.1 Principle 1: Structured and Interpretable

Principle: Research artifacts should make claims, evidence, and context explicit and directly accessible, not only embedded in prose.

The first breakdown—papers as isolated units—stems from the fact that claims and evidence are woven into narrative text. While prose is valuable for explaining motivation and argumentation, it is not optimal for supporting cumulative synthesis. A reader wishing to compare claims across papers must extract and interpret statements from natural language, a process that does not scale as the literature grows.

What this principle requires. Artifacts should represent claims, evidence, and context as structured, first-class entities. A claim should be identifiable: "Technique X improves metric Y on dataset Z by δ under conditions C." The evidence supporting that claim—experimental configuration, data, results, analysis—should be explicitly linked. The context—assumptions, limitations, methodological decisions—should be preserved alongside the claim, not buried in prose.

Structured representations enable direct comparison, aggregation, and reasoning over results. If two papers make conflicting claims about the same technique, the conflict becomes visible and resolvable by examining the structured evidence and context. If a dataset is updated or a metric refined, claims referencing that dataset or metric can be systematically identified and reevaluated.

Instantiation examples. Structured representations could take many forms:

- *Semantic annotations:* Papers supplemented with machine-readable metadata describing claims, experimental setups, and results.
- *Knowledge graph representations:* Claims, techniques, datasets, and metrics represented as entities with typed relationships, enabling queries like "Which papers claim improvements on dataset D using technique family T?"
- *Computational notebooks with assertions:* Executable analyses in which key claims are formalized as assertions that can be tested, versioned, and reused.

These are illustrative, not prescriptive. The principle is that structure should complement prose, making knowledge directly accessible for synthesis while preserving the explanatory power of narrative.

5.2 Principle 2: Inspectable and Provenance-Aware

Principle: Research artifacts should preserve the full provenance of claims—from raw data through methodological decisions to final results—and make this provenance inspectable.

The second breakdown—loss of context and provenance—occurs because the reasoning behind decisions fades as knowledge moves through the publication pipeline. A paper reports final results but often omits the path taken to reach them: why this dataset, why this baseline, why this evaluation protocol. When later researchers attempt to build on the work, they must reconstruct this reasoning, often discovering that subtle choices had significant consequences.

What this principle requires. Artifacts should document not only what was found, but how and why. Every claim should trace back to its sources: the data, the code, the configuration, the assumptions. Methodological decisions should be explicitly recorded and justified: "We used dataset D because it contains feature F relevant to hypothesis H." When results are updated—due to new data, corrected analyses, or refined methods—the provenance chain should track these changes, preserving the history of how understanding evolved.

Provenance enables trust and reproducibility. A researcher can inspect the lineage of a claim, verify that it rests on sound evidence, and understand the conditions under which it holds. When claims conflict, provenance helps diagnose the source of divergence: different data, different preprocessing, different evaluation criteria.

Instantiation examples. Provenance tracking could be realized through:

- *Versioned computational artifacts:* Code, data, and configurations stored in version control systems with clear lineage from raw inputs to published results.
- *Provenance graphs:* Explicit representations of how results were derived, linking claims to experiments, experiments to data, and data to sources.
- *Decision logs:* Structured records of methodological choices, their rationales, and their impacts, made part of the permanent artifact.

Again, these are examples. The principle is that artifacts should be inspectable—transparent about their origins and decisions—so that future work can be built on solid, well-understood foundations.

5.3 Principle 3: Long-Lived and Reusable

Principle: Research artifacts should support evolution and reuse, not remain static at the moment of publication.

The third breakdown—claims evolve without tracking—and aspects of the first breakdown arise because papers are snapshots frozen in time. Once published, a paper does not update when new evidence emerges, when datasets are revised, or when methods are improved. Follow-on work cites the paper and describes differences in prose, but the original claim remains unchanged in the literature.

What this principle requires. Artifacts should be living substrates that can be updated, extended, and reused. When a dataset is corrected, claims depending on that dataset should be re-evaluable, not obsolete. When a technique is refined, prior results should be comparable to new results using updated methods. When claims are qualified or contradicted, these relationships should be reflected in the artifact, not scattered across disconnected papers.

This does not mean papers should be continuously rewritten. Rather, the underlying structured representations—claims, evidence, configurations—should be decoupled from narrative documents. Narratives can remain stable as historical records, while the structured substrates evolve as understanding progresses.

Instantiation examples. Long-lived artifacts could include:

- *Living knowledge bases:* Repositories where claims, datasets, and techniques are maintained, versioned, and updated as new evidence accumulates.
- *Executable benchmarks:* Evaluation frameworks that can be rerun with updated data or baselines, allowing claims to be retested and results to be continuously refined.
- *Modular artifact ecosystems:* Components (datasets, models, evaluation scripts) designed for composition and reuse, enabling new studies to build directly on prior artifacts rather than reimplementing from scratch.

The principle is that knowledge should outlive individual papers, accumulating in shared substrates that reduce redundancy and enable direct building.

5.4 Principle 4: Governed with Human Oversight

Principle: Research artifacts and infrastructures should be governed by community processes that ensure quality, integrity, and ethical responsibility.

The fourth breakdown—incentive structures favor novelty—reflects the tension between individual incentives and collective needs. Even if we had perfect technical infrastructure, cumulative progress requires coordination: standards for quality, mechanisms for resolving conflicts, and recognition for contributions that consolidate rather than merely add.

What this principle requires. Artifacts cannot govern themselves. Structured representations, provenance tracking, and living substrates are valuable only if the community trusts them, maintains them, and uses them responsibly. This requires human oversight: peer review for knowledge contributions, curation of shared resources, community processes for resolving disputes or establishing standards, and credit systems that value infrastructure work alongside novel research.

Governance also addresses ethical concerns. Structured knowledge artifacts raise questions about ownership, attribution, bias, and access. Who controls a shared knowledge base? How is credit assigned when multiple researchers contribute incrementally? How do we ensure that knowledge infrastructures do not perpetuate biases present in underlying data? These are social and ethical questions, not purely technical ones, and they require community deliberation and ongoing stewardship.

Instantiation examples. Governance mechanisms could include:

- *Community curation processes:* Peer-reviewed contributions to shared knowledge bases, with clear criteria for inclusion, updating, and deprecation.
- *Credit and attribution systems:* Mechanisms that recognize contributions to infrastructure, replication, and consolidation, not only novel results.
- *Ethical oversight boards:* Community structures responsible for addressing bias, fairness, and access issues in shared knowledge infrastructures.
- *Transparent governance models:* Decision-making processes for managing shared resources, resolving conflicts, and setting standards, with broad community input.

The principle is that cumulative progress is a collective endeavor. Technical solutions alone are insufficient; they must be embedded in social practices that align individual incentives with collective goals and ensure responsible stewardship of shared knowledge.

5.5 Mapping Principles to Breakdowns

These four principles are not independent—they address the interconnected breakdowns diagnosed earlier:

- **Structured and interpretable** artifacts address the isolation of knowledge by making claims and evidence directly comparable and composable.
- **Inspectable and provenance-aware** artifacts address the loss of context by preserving the reasoning and decisions behind results.
- **Long-lived and reusable** artifacts address the static nature of claims by enabling evolution and incremental refinement over time.
- **Governed with human oversight** addresses incentive misalignment by establishing community processes that value consolidation and ensure responsible stewardship.

Together, they describe a vision for research artifacts that support cumulative knowledge building: artifacts that are not isolated documents but interconnected, evolving substrates governed by community practices that align individual contributions with collective progress.

5.6 From Principles to Practice

These principles are aspirational. Realizing them fully would require significant changes in how research is conducted, reviewed, published, and rewarded. We do not claim that these changes are easy or that any single intervention will suffice. But the principles provide a framework for evaluating incremental steps: Does a proposed tool, platform, or practice make artifacts more structured, more inspectable, more reusable, or better governed? If so, it moves the community toward more cumulative progress.

Importantly, these principles are not tied to specific technologies. Knowledge graphs, computational notebooks, version control systems, and community repositories are all potential instantiations—but the principles themselves are technology-agnostic. They describe *properties* that artifacts should have, leaving room for diverse implementations across different domains and research communities.

The challenge is to move from principles to practice: to design systems, establish norms, and reform incentives in ways that make cumulative knowledge building not only possible but rewarded. We turn to the broader implications of this shift in the next section.

6 Implications for the Future of Software Engineering

The principles articulated in the previous section are aspirational. Realizing them would require changes not only in how research artifacts are designed, but also in how research is conducted, reviewed, published, and rewarded. In this section, we discuss implications for research practice, publication norms, community infrastructure, and the role of FOSE as a venue for experimentation.

6.1 Implications for Research Practice

If research artifacts were structured, inspectable, provenance-aware, and long-lived, how would research practice change?

Planning and documentation. Researchers would need to document not only final results but also the process by which those results were obtained. This includes recording methodological decisions, their rationales, and their impacts. Tools like computational notebooks, version control systems, and workflow management platforms could support this documentation, but the cultural shift—valuing process alongside outcomes—is equally important.

Structuring contributions. Instead of presenting results solely in prose, researchers would also produce structured representations of claims, evidence, and context. This does not mean abandoning narrative—papers would still tell stories, explain motivations, and argue for significance. But they would be supplemented by machine-readable metadata, semantic annotations, or knowledge graph entries that make key contributions directly accessible for synthesis and reuse.

Designing for reuse. Researchers would consider reusability at the outset, not as an afterthought. Datasets would be documented with schemas and provenance. Code would be modular and well-documented. Evaluation protocols would be reproducible. The goal would be to produce artifacts that others can directly build on, reducing the need for reimplementation and lowering the barrier to cumulative progress.

Collaborating across studies. Structured artifacts enable new forms of collaboration. Researchers could contribute incrementally to shared knowledge bases, updating claims as new evidence emerges, proposing refinements to existing techniques, or resolving contradictions by comparing structured evidence. This would shift the unit of contribution from the paper to the structured artifact, enabling continuous, distributed knowledge building.

These changes would require time and effort. But if the infrastructure and incentives were in place, they could become standard practice, much as sharing code and data have become increasingly expected in many research communities.

6.2 Implications for Publication and Review

If the principles were adopted, what would change in how research is published and reviewed?

Evaluating contributions beyond novelty. Review processes would need to recognize contributions that consolidate, synthesize, or replicate prior work. A paper that resolves contradictions in the literature, curates a benchmark dataset, or provides infrastructure for cumulative building should be valued alongside papers that propose new techniques. This requires updating review criteria and training reviewers to assess different types of contributions.

Artifact expectations. Conferences and journals could establish expectations that submissions include not only papers but also structured artifacts: semantic annotations, provenance documentation, reusable components. Artifact evaluation tracks have made progress in this direction, but the principles suggest going further—treating structured artifacts as first-class contributions, not optional supplements.

Living publications. If artifacts evolve beyond publication, then papers could be viewed as snapshots of ongoing work rather than final records. A paper might introduce a technique and present initial results, while a living artifact continues to accumulate evidence, track refinements, and document extensions. This would require rethinking version control for scholarly communication and establishing norms for how to cite evolving artifacts.

Community review and curation. Governance mechanisms imply that some contributions would be reviewed not only at submission time but also post-publication, as they are integrated into shared knowledge bases. Community curation processes—similar to those used in some open-access repositories or collaborative platforms—could ensure that contributions meet quality standards, resolve conflicts, and maintain consistency over time.

These changes would be significant, but they are not without precedent. Fields like machine learning have experimented with leaderboards, benchmark repositories, and living evaluation frameworks. The principles suggest extending these experiments more broadly and embedding them in the publication process itself.

6.3 Implications for Community Infrastructure

If the principles were realized, what infrastructure would be needed?

Shared knowledge repositories. The community would need platforms for storing, querying, and updating structured research artifacts. These could take many forms: knowledge graphs representing claims and evidence; benchmark repositories hosting datasets and evaluation scripts; collaborative platforms where researchers propose and refine contributions. The key is that these repositories would be long-lived, maintained, and governed by community processes.

Standards and interoperability. For artifacts to be reusable across studies, the community would need shared standards: ontologies for representing claims and techniques, schemas for documenting datasets, protocols for reporting results. Establishing these standards requires coordination, but the benefits—reduced friction in synthesis and reuse—could be substantial.

Credit and attribution systems. Infrastructure contributions, replication efforts, and incremental refinements would need to be recognized and credited. This could involve alternative metrics (e.g., how often a dataset is reused, how many claims reference a benchmark), changes in tenure and promotion criteria, or platforms that track contributions beyond traditional papers. Aligning individual incentives with collective knowledge-building goals is essential for sustainability.

Governance and stewardship. Shared infrastructures require stewardship: maintaining repositories, resolving disputes, updating standards, addressing ethical concerns. The community would need governance structures—committees, boards, or distributed processes—to manage these responsibilities. Transparency, inclusivity, and accountability would be critical to ensuring that infrastructures serve the collective good.

Building and maintaining such infrastructure is not trivial. It requires sustained investment, coordination across institutions and subfields, and a willingness to prioritize collective needs alongside individual research agendas. But without infrastructure, the principles remain aspirational.

6.4 FOSE as a Venue for Experimentation

The Future of Software Engineering track is uniquely positioned to support experimentation with alternative artifact types and practices. FOSE papers do not need to conform to the same evaluation criteria as empirical studies or tool papers. They can be reflective, speculative, and forward-looking. This creates space for contributions that would not fit traditional tracks but that could inform the evolution of the field.

Prototyping alternative formats. FOSE could encourage submissions that experiment with structured artifacts, living documents, or collaborative knowledge-building platforms. These contributions would not be judged solely on novelty or empirical rigor but on their potential to demonstrate new ways of organizing and sharing knowledge.

Community dialogues. FOSE sessions could serve as forums for discussing infrastructure needs, governance models, and incentive alignment. The track could facilitate conversations that bridge research practice, publication systems, and community infrastructure—conversations that are difficult to have within the constraints of traditional paper sessions.

Long-term tracking. FOSE could track experiments over time, documenting what works, what fails, and what lessons emerge. This would provide evidence for evaluating alternative practices and informing broader community decisions about adopting new norms or infrastructures.

By embracing experimentation, FOSE can help the software engineering community move from principles to practice, testing ideas, refining approaches, and building momentum for structural change.

6.5 Challenges and Trade-offs

Realizing these implications would not be without challenges. Structured artifacts require effort to produce and maintain. Standards risk becoming rigid or exclusionary. Governance processes can be contentious. Infrastructure requires resources that may not be available or equitably distributed.

Moreover, there are trade-offs. Emphasizing structure and reusability could discourage exploratory, early-stage work that does not yet fit established frameworks. Focusing on consolidation could slow the introduction of genuinely novel ideas. Balancing these tensions—supporting both exploration and consolidation, novelty and cumulation, individual creativity and collective building—is an ongoing challenge.

The goal is not to replace current practices wholesale but to expand the range of valued contributions and supported practices. Papers will remain important for storytelling and argumentation. But they could be complemented by structured artifacts, living repositories, and community infrastructures that make cumulative progress more feasible and more rewarded.

7 Limitations and Open Questions

This paper offers a diagnosis of structural barriers to knowledge accumulation and a set of principles for future research artifacts. But it does not provide solutions, nor does it claim that realizing these principles will be straightforward. In this section, we acknowledge limitations of our approach and identify open questions that require further investigation.

7.1 Limitations of the Survey Data

Our analysis draws on responses from the ICSE 2026 FOSE pre-survey. While these data provide valuable community perspectives, they have limitations.

Scope and representativeness. The survey captured 280 responses, representing a small fraction of the global software engineering research community. Respondents self-selected, which may introduce bias: those who chose to respond may have stronger opinions or greater engagement with meta-research questions than the broader population. We cannot claim that the findings generalize to all researchers or all subfields within software engineering.

Descriptive, not causal. We use survey data to establish that the community is experienced, productive, and globally distributed. This supports our argument that accumulation challenges are structural rather than due to lack of expertise. But the survey data do not establish causation, nor do they provide direct evidence for the specific breakdowns we diagnose. Our diagnosis is interpretive, grounded in the data but also informed by broader observations and prior work.

Snapshot in time. The survey reflects perspectives at a particular moment. Community concerns and priorities evolve. What is salient in 2026 may differ from what was salient a decade ago or what will be salient a decade hence. Our findings should be understood as contextual, not universal.

7.2 Generality Versus Specificity

The principles we articulate are intentionally abstract, describing properties that future artifacts should have without prescribing specific implementations. This generality has both strengths and weaknesses.

Strengths. Technology-agnostic principles allow for diverse implementations across subfields and research communities. They provide a framework for evaluating proposals without committing to any single approach. They avoid the risk of advocating for tools or platforms that may become obsolete.

Weaknesses. Abstract principles can be difficult to operationalize. How does one know when an artifact is "sufficiently structured" or "adequately provenance-aware"? Without concrete instantiations, principles may be too vague to guide action. Moreover, the tension between generality and specificity risks either being so broad as to be unhelpful or so narrow as to exclude valid alternatives.

We have attempted to balance this tension by providing illustrative examples alongside each principle. But translating principles into practice will require experimentation, iteration, and community deliberation.

7.3 Feasibility and Adoption Challenges

Even if the principles are sound, realizing them faces practical challenges.

Effort and resources. Producing structured, provenance-aware, long-lived artifacts requires effort. Researchers already face time pressures, competing priorities, and limited resources. Asking them to invest more in documentation, structuring, and curation may be unrealistic without corresponding reductions in other demands or increases in support.

Infrastructure gaps. The infrastructure needed to support cumulative knowledge building—shared repositories, standards, governance processes—does not yet exist in many areas of software engineering. Building it requires sustained investment, coordination, and institutional commitment. It is unclear where this investment will come from or how coordination will be achieved.

Incentive misalignment. Even with infrastructure in place, adoption depends on incentives. If tenure and promotion decisions continue to prioritize novel papers over infrastructure contributions, researchers will have little reason to invest in cumulative building. Changing incentives requires action at multiple levels: funding agencies, universities, conferences, and journals. Coordination across these actors is difficult.

Cultural inertia. Research communities have established norms and practices. Changing them requires not only technical solutions and policy reforms but also shifts in culture and expectations. Such shifts take time and may face resistance from those who succeed under current systems.

We do not underestimate these challenges. But acknowledging them is a first step toward addressing them.

7.4 Ethical and Social Considerations

Structured knowledge infrastructures raise ethical and social questions that we have not fully addressed.

Ownership and control. Who owns shared knowledge bases? Who decides what is included, updated, or removed? Centralized control risks gatekeeping and exclusion. Distributed control risks fragmentation and inconsistency. Finding governance models that balance these concerns is an open problem.

Bias and fairness. Knowledge artifacts reflect the data, assumptions, and priorities of their creators. If those creators are not diverse, the artifacts may perpetuate biases. Ensuring that knowledge infrastructures are inclusive, representative, and fair requires ongoing attention and deliberate effort.

Access and equity. Building and maintaining infrastructure requires resources. If these resources are concentrated in well-funded institutions or regions, the benefits may accrue unevenly. Ensuring equitable access to shared infrastructures and equitable participation in their governance is critical but not straightforward.

Privacy and consent. Structured artifacts may involve data about research processes, collaborations, or individuals. Ensuring that such data are used ethically, with appropriate consent and privacy protections, is essential.

These are not purely technical issues. They require community deliberation, ethical oversight, and ongoing stewardship. We do not claim to have answers, but we emphasize that any future infrastructure must address these concerns.

7.5 Open Research Questions

Beyond the limitations above, several open questions remain:

- **What forms of structure are most useful?** Different subfields may benefit from different representational frameworks. How do we balance standardization (for interoperability) with flexibility (for domain-specific needs)?
- **How do we evaluate progress?** If artifacts evolve over time, how do we measure whether they are improving? What metrics indicate successful cumulative building versus fragmentation or stagnation?

- **What governance models work best?** Centralized, federated, or fully distributed? Community-driven or institution-led? Different models have different trade-offs, and the optimal choice may vary by context.
- **How do we transition from current practices?** Incremental adoption is likely more feasible than wholesale replacement. But what incremental steps are most effective? How do we avoid creating parallel systems that fragment the community further?
- **What role should automation play?** Tools like large language models could potentially help structure knowledge, extract claims, or track provenance. But they also introduce risks of error, bias, and over-reliance on automation. How do we balance human oversight with scalable automation?

These questions do not have easy answers. They will require empirical investigation, experimentation, and ongoing community dialogue.

7.6 Reflections on Scope

Finally, we acknowledge that this paper focuses on structural and infrastructural challenges. We do not address all barriers to cumulative progress. Individual researcher behavior, institutional policies, funding priorities, and broader societal factors all play roles. Structural changes alone will not suffice if researchers are not motivated to adopt new practices, if institutions do not reward infrastructure contributions, or if funding does not support long-term knowledge-building efforts.

Our goal is not to provide a complete solution but to articulate one important dimension of the challenge: the design of research artifacts and the infrastructures that support them. We hope this diagnosis and these principles can inform broader efforts to make software engineering research more cumulative, transparent, and impactful.

8 Conclusion

Software engineering research is thriving by many measures. The field is productive, the community is experienced, and participation spans the globe. Yet this success in producing research outputs has not translated fully into cumulative knowledge progress. Claims remain fragmented across disconnected papers. Context and provenance are lost as knowledge moves through the publication pipeline. Synthesis and reuse remain effortful and error-prone. The mechanisms we have for accumulating knowledge have not kept pace with the mechanisms we have for producing it.

This paper has argued that the barriers to cumulative progress are structural, not individual. They arise from how research artifacts are designed, how knowledge is represented, and how incentives shape practice. If papers function as isolated documents, if provenance is not preserved, if claims evolve without tracking, and if novelty is rewarded over consolidation, then accumulation will remain limited no matter how skilled or motivated the community.

We have diagnosed four interrelated breakdowns and proposed four principles to address them. Structured and interpretable artifacts make claims and evidence directly accessible. Inspectable and provenance-aware artifacts preserve the reasoning behind results. Long-lived and reusable artifacts support evolution beyond publication. Governed artifacts align individual incentives with collective knowledge-building goals.

These principles are aspirational. Realizing them would require changes in research practice, publication norms, community infrastructure, and incentive systems. We do not underestimate the challenges. Building shared infrastructures, establishing standards, reforming review processes, and changing institutional priorities are difficult, time-consuming, and require coordination across many actors.

But the alternative—continuing with practices optimized for dissemination and novelty rather than accumulation and reuse—will perpetuate the fragmentation and inefficiency we currently experience. As the field continues to grow, these challenges will only intensify. What is manageable today may become unmanageable tomorrow.

The Future of Software Engineering track is an ideal venue for this conversation. FOSE papers are not expected to provide complete solutions or empirical validation. They are expected to be reflective, forward-looking, and agenda-setting. This paper has aimed to meet that expectation: to articulate why cumulative progress remains constrained, to propose principles for addressing these constraints, and to discuss implications for how the community might move forward.

We see several potential next steps. First, the community could experiment with alternative artifact types and practices, using FOSE and other venues as spaces for prototyping and evaluation. Second, infrastructure initiatives—knowledge repositories, benchmark collections, collaborative platforms—could be developed with explicit attention to the principles we have articulated. Third, conferences, journals, and institutions could revisit policies and criteria to recognize contributions that consolidate, replicate, and support cumulative building alongside those that introduce novelty.

Most importantly, the community could engage in dialogue about what cumulative progress means, what it requires, and what trade-offs are acceptable. This paper offers one perspective, grounded in survey data and informed by related work. But it is not the final word. Other perspectives, critiques, and refinements are essential.

The software engineering research community has demonstrated its capacity for growth, adaptation, and innovation. The field has evolved in response to changing technologies, expanding application domains, and new research methods. Rethinking how we design research artifacts and build knowledge infrastructures is the next evolution—one that could make the difference between sustained productivity and sustained progress.

The question is not whether cumulative knowledge building is possible. It is whether the community will prioritize it, invest in it, and organize around it. The principles we have articulated provide a starting point. The path from principles to practice will require effort, experimentation, and commitment. But the potential reward—a research community that not only produces knowledge but also accumulates, integrates, and reuses it effectively—is worth pursuing.

The future of software engineering depends not only on what we discover but on how we organize and preserve what we know. Let us build infrastructures and practices that make cumulative progress not just possible but rewarded, supported, and celebrated.

Let us ensure that the next generation of researchers inherits not just a literature to read but a knowledge base to build upon.

References

[1] Liz Allen, Jo Scott, Amy Brand, Marjorie Hlava, and Micah Altman. 2014. Publishing: Credit where credit is due. *Nature* 508, 7496 (2014), 312–313. doi:10.1038/508312a

[2] Daniel E. Atkins, Kelvin K. Droegemeier, Stuart I. Feldman, Hector Garcia-Molina, Michael L. Klein, David G. Messerschmitt, Paul Messina, Jeremiah P. Ostriker, and Margaret H. Wright. 2003. Revolutionizing Science and Engineering Through Cyberinfrastructure: Report of the National Science Foundation Blue Ribbon Advisory Panel on Cyberinfrastructure. *National Science Foundation* (2003).

[3] Monya Baker. 2016. 1,500 scientists lift the lid on reproducibility. *Nature* 533, 7604 (2016), 452–454. doi:10.1038/533452a

[4] Victor R. Basili, Forrest Shull, and Filippo Lanubile. 1999. Building knowledge through families of experiments. *IEEE Transactions on Software Engineering* 25, 4, 456–473. doi:10.1109/32.799955

[5] Christian Collberg and Todd A. Proebsting. 2016. Repeatability in Computer Systems Research. *Commun. ACM* 59, 3, 62–69. doi:10.1145/2812803

[6] Hyunsook Do, Sebastian Elbaum, and Gregg Rothermel. 2005. Supporting controlled experimentation with testing techniques: An infrastructure and its potential impact. *Empirical Software Engineering* 10, 4, 405–435. doi:10.1007/s10664-005-3861-2

[7] David Donoho. 2017. 50 Years of Data Science. *Journal of Computational and Graphical Statistics* 26, 4 (2017), 745–766. doi:10.1080/10618600.2017.1384734

[8] Santo Fortunato, Carl T. Bergstrom, Katy Börner, James A. Evans, Dirk Helbing, Staša Milojević, Alexander M. Petersen, Filippo Radicchi, Roberta Sinatra, Brian Uzzi, Alessandro Vespignani, Ludo Waltman, Dashun Wang, and Albert-László Barabási. 2018. Science of science. *Science* 359, 6379 (2018), eaao0185. doi:10.1126/science.aao0185

[9] Jesús M. González-Barahona and Gregorio Robles. 2012. On the reproducibility of empirical software engineering studies based on data retrieved from development repositories. *Empirical Software Engineering* 17, 1-2 (2012), 75–89. doi:10.1007/s10664-011-9181-9

[10] Ben Herman, Emily Winter, Michail D. Papamichail, Georgios Gousios, and Andy Zaidman. 2020. A 2020 perspective on the replicability of software engineering research. In *Proceedings of the 14th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*. 1–12. doi:10.1145/3382494.3410652

[11] John P. A. Ioannidis. 2005. Why Most Published Research Findings Are False. *PLOS Medicine* 2, 8 (2005), e124. doi:10.1371/journal.pmed.0020124

[12] John P. A. Ioannidis. 2018. Meta-research: Why research on research matters. *PLOS Biology* 16, 3 (2018), e2005468. doi:10.1371/journal.pbio.2005468

[13] Mohamad Yaser Jaradeh, Allard Oelen, Kheir Eddine Farfar, Manuel Prinz, Jennifer D'Souza, Gábor Kismihók, Markus Stocker, and Sören Auer. 2019. Open Research Knowledge Graph: Next Generation Infrastructure for Semantic Scholarly Knowledge. In *Proceedings of the 10th International Conference on Knowledge Capture*. 243–246. doi:10.1145/3360901.3364435

[14] Natalia Juristo and Sira Vegas. 2011. Using differences among replications of software engineering experiments to gain knowledge. *Empirical Software Engineering* 16, 4 (2011), 511–545. doi:10.1007/s10664-010-9146-1

[15] Barbara Kitchenham, Pearl Brereton, David Budgen, Mark Turner, John Bailey, and Stephen Linkman. 2009. Systematic literature reviews in software engineering – A systematic literature review. *Information and Software Technology* 51, 1 (2009), 7–15. doi:10.1016/j.infsof.2008.09.009

[16] Thomas S. Kuhn. 1962. *The Structure of Scientific Revolutions* (1st ed.). University of Chicago Press.

[17] Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. Multi-Task Identification of Entities, Relations, and Coreference for Scientific Knowledge Graph Construction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 3219–3232. doi:10.18653/v1/D18-1360

[18] Brian A. Nosek, Jeffrey R. Spies, and Matt Motyl. 2012. Scientific Utopia: II. Restructuring Incentives and Practices to Promote Truth Over Publishability. *Perspectives on Psychological Science* 7, 6 (2012), 615–631. doi:10.1177/1745691612459058

[19] Margit Osterloh and Bruno S. Frey. 2020. How to avoid borrowed plumes in academia. *Research Policy* 49, 1 (2020), 103831. doi:10.1016/j.respol.2019.103831

[20] Forrest Shull, Jeffrey C. Carver, Sira Vegas, and Natalia Juristo. 2008. The role of replications in Empirical Software Engineering. In *Empirical Software Engineering and Verification*. Springer, 211–244. doi:10.1007/978-3-540-71301-2_8

[21] Dag I. K. Sjøberg, Tore Dybå, Bente C. D. Anda, and Jo E. Hannay. 2008. Building Theories in Software Engineering. *Guide to Advanced Empirical Software Engineering* (2008), 312–336. doi:10.1007/978-1-84800-044-5_12

[22] Paul E. Smaldino and Richard McElreath. 2016. The natural selection of bad science. *Royal Society Open Science* 3, 9 (2016), 160384. doi:10.1098/rsos.160384

[23] Dashun Wang and Albert-László Barabási. 2013. The Science of Science. *Physics Reports* 510, 3 (2013), 87–149. doi:10.1016/j.physrep.2012.02.006

[24] Kuansan Wang, Zhihong Shen, Chiyuan Huang, Chieh-Han Wu, Yuxiao Dong, and Anshul Kanakia. 2020. Microsoft Academic Graph: When experts are not enough. *Quantitative Science Studies* 1, 1, 396–413. doi:10.1162/qss_a_00021