From Papers to Progress: Rethinking Knowledge Accumulation in Software Engineering

Jason Cusati
Virginia Tech
Blacksburg, VA, USA
djjay@vt.edu

Chris Brown
Virginia Tech
Blacksburg, VA, USA
dcbrown@vt.edu

Abstract

Software engineering research has experienced rapid 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 280 globally distributed and highly experienced 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:
Jason Cusati and Chris Brown. 2026. From Papers to Progress: Rethinking Knowledge Accumulation in Software Engineering. In *2026 IEEE/ACM 48th International Conference on Software Engineering: Future of Software Engineering (ICSE-FoSE), April 12–18, 2026, Rio de Janeiro, Brazil.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3793657.3793888

ICSE-FoSE, Rio de Janeiro, Brazil

ACM ISBN 979-8-4007-2479-4/2026/04
https://doi.org/10.1145/3793657.3793888

1 Introduction

Software engineering research is thriving by most measures. Conference submissions keep rising, publication venues have multiplied, and the community has gone global. Researchers are productive, techniques keep advancing, and new areas of inquiry emerge every year. But a persistent challenge lurks beneath all this activity: we struggle to build cumulative knowledge that actually connects, consolidates, and extends prior work.

This is not a new concern, nor one unique to our field. Across science, more publications have not automatically meant deeper understanding [10, 13]. In software engineering specifically, fragmentation and the difficulty of synthesizing results have been raised repeatedly [8, 19, 20]. Yet the structural factors limiting cumulative progress remain underexplored.

This paper offers a community-informed perspective on knowledge accumulation in software engineering. Drawing on 280 ICSE 2026 FOSE pre-survey responses from an experienced (57% with 10+ years), productive (44% with 11+ papers in 3 years), globally distributed community, we diagnose four structural breakdowns: papers as isolated units, lost context and provenance, untracked claim evolution, and incentives favoring novelty over consolidation. We then propose four principles for research artifacts: structured and interpretable, inspectable and provenance-aware, long-lived and reusable, and community-governed. We do not prescribe specific tools or claim to solve these challenges; instead, we synthesize community observations into a diagnosis and principles to guide future work.

2 Community Signals from the ICSE 2026 FOSE Pre-Survey

We ground our diagnosis in the ICSE 2026 Future of Software Engineering pre-survey (n=280) [23]. Two researchers performed a lightweight qualitative analysis, examining comments for insights about knowledge-building in software engineering. The respondents span six geographic regions (Europe 50%, North America 25%, Asia 15%, others 10%) and represent a mature, productive community. This profile matters for our argument: if knowledge accumulation remains limited despite all this expertise and output, the barriers are *structural*, not individual. The community is not lacking effort. Respondents themselves recognize systemic challenges, noting fragmentation across resubmissions, difficulties building on prior work, and incentives that reward novelty over consolidation. These observations motivate our diagnosis.

3 Where Knowledge Accumulation Breaks Down

If the community is experienced, productive, and global, yet still struggles to synthesize results and build on prior work, the barriers are structural. We identify four interrelated breakdowns in how research artifacts are produced, represented, and connected. Anonymous participant identifiers appear in parentheses.

3.1 Papers as Isolated Knowledge Units

Research papers package claims, evidence, and context into narrative prose optimized for dissemination. This works well for presenting new ideas but creates barriers to cumulative building. One participant noted that "*15 years ago [it was the norm to] present and discuss a research idea at a workshop then go back home, build and evaluate the thing, then publish at a conference. These days, reviewers repeatedly reject such submissions 'because the work has already appeared at this and that workshop'*" (195811076). Claims get buried in paragraphs alongside motivation and methods. Extracting and comparing claims across papers means reading entire documents and reconciling different terminology. Evidence linkages stay implicit; readers infer which results support which claims under what assumptions. Even when artifacts are shared, results freeze at publication time. Updates to datasets or baselines require new papers, while original claims sit unchanged. As another participant observed, "*despite having a software artifact track, not many of the software artifacts are being actively maintained*" (196030130). Each paper becomes an island connected only through citations and prose, requiring manual synthesis that does not scale.

3.2 Loss of Context and Provenance

Research involves countless contextual decisions: which dataset, how to split data, which baselines, how to handle edge cases. Papers describe these choices but often omit the rationale. One participant noted frustration with "*software engineering researchers trying to do research with generative AI tools without specifying context first*" (196032564). As work gets cited and summarized, motivation and assumptions fade. Carefully qualified findings become unqualified facts in subsequent literature. Methodological decisions must be reverse-engineered, sometimes revealing that subtle choices significantly impacted results. Papers present polished final versions, hiding the path from hypothesis to result. When context and provenance are lost, later researchers either accept prior work at face value or invest substantial effort reconstructing the reasoning. Moreover, despite goals to enhance practice, research has "*minimal*" and "*almost no relevance*" to industry contexts (192101946).

3.3 Claims Evolve Without Tracking

Scientific claims get refined, qualified, contradicted, and superseded as evidence accumulates, but the publication system provides few tracking mechanisms. The literature may contain conflicting claims (technique A outperforms B in one paper, the reverse in another) without systematic flagging. Survey respondents noted that "*most of the SE research being published is neither reproducible nor replicable*" (195853386) and "*we are reluctant to accept papers that replicate previous study...see medical sciences as counter example - they publish single cases as they allow to build body of konowledge in the long run*" (196067582). When researchers discover conflicts, they must investigate causes themselves, often finding subtle methodological differences explain the divergence. Refinements stay implicit: a follow-on paper may qualify a prior claim, but the original remains unchanged. Claim relationships (does one extend, contradict, or depend on another?) appear only in natural language like "building on [12]" or "in contrast to [34]." Without structured representations, determining what is currently believed, under what conditions, and with what confidence requires extensive manual synthesis.

3.4 Incentive Structures Favor Novelty Over Accumulation

The final breakdown is embedded in incentive structures. Publication venues, hiring criteria, and funding mechanisms reward novelty [18, 21]. Conference and journal reviews prioritize new techniques; replication studies, negative results, and syntheses face higher acceptance bars even when they would advance collective understanding [3]. Researchers investing in replication or shared infrastructure face opportunity costs: these efforts may not yield top-venue publications or count in promotion reviews [7]. One participant stated "*we are not good at replication (despite better rigour in what we do)...too much effort goes into chasing 'new' rather than consolidating our knowledge*" (192758693). Building knowledge repositories and interoperability layers requires sustained effort, often led by small groups without commensurate recognition. Everyone benefits from better infrastructure, but individuals face little incentive to contribute. As long as novelty is privileged over accumulation, knowledge stays fragmented.

These four breakdowns reinforce one another. Incremental fixes (better citation practices, more replications, improved repositories) help but are not enough. Addressing these barriers requires rethinking the fundamental properties of research artifacts themselves.

4 Rethinking Research Artifacts for Cumulative Progress

The structural barriers trace back to design choices in how research artifacts are produced and shared. By *research artifacts* we mean the tangible outputs of research: papers, datasets, code repositories, benchmarks, computational notebooks, knowledge graphs, and experimental protocols. These artifacts are vessels for knowledge. They encode claims, evidence, methodological context, and the provenance of results. How well knowledge accumulates depends on their properties: whether claims can be extracted, evidence traced, artifacts reused beyond publication, and communities can build on prior work.

Addressing these barriers means reconsidering artifact properties. We propose four technology-agnostic principles. Implementations will vary, but these principles provide a shared foundation for evaluating research infrastructure.

4.1 Principle 1: Structured and Interpretable

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

The first breakdown, papers as isolated units, stems from claims and evidence being woven into narrative text. Prose works well for explaining motivation, but it is not ideal for cumulative synthesis. Comparing claims across papers means reading entire documents and reconciling terminology. This does not scale as the literature grows.

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." Evidence and context should be explicitly linked, not buried in prose. Structured representations let conflicting claims become visible and resolvable by examining evidence directly. Claims referencing updated datasets can be systematically reevaluated. Examples include semantic annotations (machine-readable metadata describing claims and results) or knowledge graphs [14, 22, 24, 25] (entities with typed relationships enabling queries like "Which papers claim improvements on dataset D?"). Structure complements prose, making knowledge directly accessible while preserving narrative's explanatory power.

4.2 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 (accessible and understandable by researchers who want to examine, verify, or build on the work).

The second breakdown, loss of context and provenance, occurs because reasoning behind decisions fades as knowledge moves through publication. Papers report final results but often omit the path taken: why this dataset, baseline, or protocol. Later researchers must reconstruct this reasoning, often discovering that subtle choices significantly impacted results.

Artifacts should document not just what was found but how and why. Every claim should trace to its sources: data, code, configuration, assumptions. Methodological decisions should be explicitly justified. When results are updated, the provenance chain should track changes [11]. Provenance enables trust: researchers can inspect a claim's lineage, verify evidence, and understand conditions. When claims conflict, provenance helps diagnose divergence. Examples include versioned computational artifacts (code and data in version control with clear lineage) or provenance graphs (explicit representations linking claims to experiments to data).

4.3 Principle 3: Long-Lived and Reusable

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

Papers are snapshots frozen in time. Once published, they rarely update when new evidence emerges, datasets are revised, or methods improve. Follow-on work cites and describes differences in prose, but original claims remain unchanged. Research artifacts are also often hard to reuse. Prior efforts to replicate research systems from ICSE and FSE tool demonstration tracks succeeded in running only about half (76/131), despite thousands of hours of effort [16].

Artifacts should be living substrates that can be updated, extended, and reused. When datasets are corrected, dependent claims should be re-evaluable. When techniques are refined, prior results should be comparable. This does not mean rewriting papers. Instead, underlying structured representations decouple from narrative documents. Narratives remain stable as historical records while structured substrates evolve. Examples include living knowledge bases (repositories where claims and datasets are versioned and updated) or executable benchmarks (evaluation frameworks rerun with updated data). Knowledge should outlive individual papers, accumulating in shared substrates.

4.4 Principle 4: Governed with Human Oversight

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

Even with perfect technical infrastructure, cumulative progress requires coordination: quality standards, conflict resolution, and recognition for consolidation work. Artifacts cannot govern themselves. Structured representations and provenance tracking are valuable only if the community trusts, maintains, and uses them responsibly. This requires human oversight: peer review for knowledge contributions, curation, dispute resolution, and credit systems valuing infrastructure work alongside novel research. The human-artifact model suggests human expertise is necessary to understand artifacts within broader ecosystems and motivate future designs [4].

Governance also addresses ethical concerns: ownership, attribution, bias, and access. Who controls shared knowledge bases? How is credit assigned for incremental contributions? How do we prevent infrastructures from perpetuating biases? These questions require community deliberation and ongoing stewardship. Examples include community curation processes (peer-reviewed contributions with clear criteria) or credit systems recognizing infrastructure, replication, and consolidation work. Cumulative progress is a collective endeavor; technical solutions must be embedded in social practices that align individual incentives with collective goals.

5 From Principles to Practice: Implications for the Future of Software Engineering

These four principles provide a framework for evaluating incremental steps: does a proposed tool or practice make artifacts more structured, inspectable, reusable, or better governed? The principles are technology-agnostic. Knowledge graphs, computational notebooks, and repositories are possible instantiations, but the principles describe *properties*, leaving room for diverse implementations. The challenge is designing systems and reforming incentives so cumulative knowledge building becomes both possible and rewarded.

5.1 Research Practice and Publication

Research practice would shift toward documenting process alongside outcomes: recording methodological decisions and their rationales using computational notebooks, version control, and workflow platforms. Researchers would supplement narrative papers with structured representations (semantic annotations or knowledge graph entries) making contributions directly accessible for synthesis. Designing for reuse would become standard: datasets documented with schemas, code modular and documented, protocols reproducible.

Publication and review would need to value consolidation alongside novelty. Papers that resolve contradictions, curate benchmarks, or provide infrastructure deserve recognition. These replications are critical to avoid contributing to the "vast graveyard of undead theories" in software research [9]. Conferences should treat structured artifacts as first-class contributions, not optional supplements [17]. As one participant noted, replication studies are considered "*thankless volunteer work*" and "*it is not sustainable for people to be doing these things well and frequently without proper incentives*" (196051220). Replication efforts in other domains found only about one-third of published experiments in psychology (39/100) [6] and economics (22/67) [5] could reproduce statistically significant results. If artifacts evolve post-publication, papers become snapshots while living artifacts accumulate evidence, requiring new norms for citing evolving work.

5.2 Community Infrastructure

The community needs platforms for storing, querying, and updating structured artifacts (knowledge graphs, benchmark repositories, collaborative platforms) maintained through community governance [2, 15]. One survey participant proposed "*a shared repository for research ideas, datasets, and research materials that lead to better distribution, collaboration, and replication*" (196181126). Shared standards (ontologies, schemas, reporting protocols) enable reuse across studies [26]. Infrastructure contributions and replications need recognition through alternative metrics or contribution-tracking platforms [1]. Governance structures manage stewardship: maintaining repositories, resolving disputes, addressing ethical concerns. Without such infrastructure, the principles stay aspirational.

5.3 FOSE as a Venue for Experimentation

The Future of Software Engineering track could support experimentation with alternative artifact types: submissions experimenting with structured artifacts or living documents, judged on potential to demonstrate new knowledge-organizing approaches rather than solely novelty.

Consider a study finding automated code review reduces defects by 15%. Under current norms, this paper gets cited and superseded as its dataset ages. Under our principles, the group registers structured claims in a knowledge graph, links them to versioned data with provenance, and designs for reuse. When contradictions emerge, structured representations enable diagnosing whether differences stem from datasets or methods, turning disputes into refinement. FOSE could welcome such experiments, building evidence for broader adoption.

6 Limitations

This paper diagnoses structural barriers and proposes principles, but does not provide complete solutions.

Survey limitations. Our analysis draws on 280 FOSE pre-survey responses, a self-selected fraction of the global community. The data are descriptive: they establish that the community is experienced and productive, supporting our argument that barriers are structural, but they do not prove specific breakdowns. The survey reflects perspectives at one moment, and our principles may not generalize to other domains.

Feasibility challenges. Producing structured, provenance-aware artifacts takes effort that researchers may lack without support. The needed infrastructure (shared repositories, standards, governance) does not yet exist in many areas [12]. Adoption depends on changing incentives at multiple levels, which is difficult and slow.

Ethical concerns. Structured knowledge infrastructures raise questions about ownership, bias, access, and privacy. These are not purely technical issues; they require community deliberation and ongoing stewardship.

7 Conclusion

Software engineering research is productive and globally distributed, yet knowledge accumulation lags behind knowledge production. Drawing on survey responses from the community, this paper argues the barriers are structural: papers function as isolated documents, provenance is lost, claims evolve without tracking, and incentives favor novelty over consolidation. We propose four principles to guide future work: structured and interpretable, inspectable and provenance-aware, long-lived and reusable, and governed with human oversight.

These principles are aspirational and technology-agnostic. Realizing them requires changes in practice, publication norms, infrastructure, and incentives. The challenges are significant, but continuing with practices optimized for dissemination rather than accumulation will perpetuate fragmentation as the field grows.

The Future of Software Engineering track provides space for this conversation. Next steps include experimenting with alternative artifact designs, developing aligned infrastructure, and revising recognition systems to value consolidation alongside novelty. The community must engage in dialogue about what cumulative progress requires and what trade-offs are acceptable.

The future of software engineering depends not only on what we discover but on how we organize and preserve what we know. Building infrastructures that support cumulative knowledge building, making it rewarded and not just possible, is essential to ensuring the next generation 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] 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

[4] Susanne Bødker and Clemens Nylandsted Klokmose. 2011. The human–artifact model: An activity theoretical approach to artifact ecologies. *Human–Computer Interaction* 26, 4 (2011), 315–371.

[5] Andrew C Chang and Phillip Li. 2015. Is economics research replicable? Sixty published papers from thirteen journals say'usually not'. (2015).

[6] Open Science Collaboration. 2015. Estimating the reproducibility of psychological science. *Science* 349, 6251 (2015), aac4716.

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

[8] Daniela S Cruzes and Tore Dybå. 2010. Synthesizing evidence in software engineering research. In *Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement.* 1–10.

[9] Christopher J Ferguson and Moritz Heene. 2012. A vast graveyard of undead theories: Publication bias and psychological science's aversion to the null. *Perspectives on Psychological Science* 7, 6 (2012), 555–561.

[10] 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

[11] 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

[12] 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

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

[14] 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

[15] Iman Keivanloo, Christophe Forbes, Jamal Ali, and Ying Zou. 2022. Defining a Knowledge Graph Development Process Through a Systematic Review. *ACM Transactions on Software Engineering and Methodology* 32, 1, Article 6 (2022). doi:10.1145/3522586

[16] Emerson Murphy-Hill et al. 2016. *Reviving 76 of 131 of Tools from ICSE and FSE*. Technical Report. GitHub (unpublished). https://github.com/SoftwareEngineeringToolDemos/Paper.

[17] Elisa Yumi Nakagawa, Pooja Rani, Sergei Ostroumov, et al. 2024. Research artifacts in software engineering publications: Status and trends. *Information and Software Technology* 171 (2024), 107457. doi:10.1016/j.infsof.2024.107457 Empirical study of research artifacts in SE publications from 2017-2022, finding artifact adoption increased from 60.1% to 81.1% in top-tier conferences.

[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] 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

[20] 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

[21] 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

[22] Markus Stocker, Allard Oelen, Muhammad Haris, Olga Kravchenko, Giuseppe Abramo, and Sören Auer. 2024. Organizing Scientific Knowledge from Engineering Sciences Using the Open Research Knowledge Graph: The Tailored Forming Process Chain Use Case. *Data Science Journal* 23, 1 (2024), 52. doi:10.5334/dsj-2024-052 Demonstrates structured knowledge organization using ORKG for engineering research.

[23] Margaret Storey and Andre van der Hoek. 2025. *Community Survey for ICSE 2026 Future of Software Engineering: Toward a Healthy Software Engineering Community*. doi:10.5281/zenodo.18217799

[24] Chuanyi Wang, Hao Hu, Fangcheng Yang, Yuwei Zeng, and Qingshan Zheng. 2023. Application of knowledge graph in software engineering field: A systematic literature review. *Information and Software Technology* 163 (2023), 107304. doi:10.1016/j.infsof.2023.107304

[25] 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

[26] Claes Wohlin, Per Runeson, Martin Höst, Magnus C. Ohlsson, Björn Regnell, and Anders Wesslén. 2012. Experimentation in Software Engineering. (2012). doi:10.1007/978-3-642-29044-2