Overview

Legacy finance systems classify technical writing as overhead because they were designed to measure tangible assets like machines and inventory rather than internal knowledge work. This outdated model creates a significant valuation gap for modern companies, where product, API, and compliance knowledge carry daily operational value. Because businesses track writing as a cost center rather than an asset, information workflows suffer from weak ownership, poor metadata, and fractured lifecycles, despite creating real value across customer adoption and engineering continuity.

Modern information architecture directly bridges this valuation gap by treating knowledge as a strategic asset. This structured optimization ensures that file repositories are prepared for both human teams and advanced AI systems.

Obsolete model: technical writers as overhead

A cost center tells leadership what a function spends. It does not show how much operating leverage that function creates across the company. That distinction matters in technical publications because the writing expense is visible in payroll, while the operational value is dispersed across engineering, support, customer onboarding, implementation, compliance, release readiness, and institutional continuity.

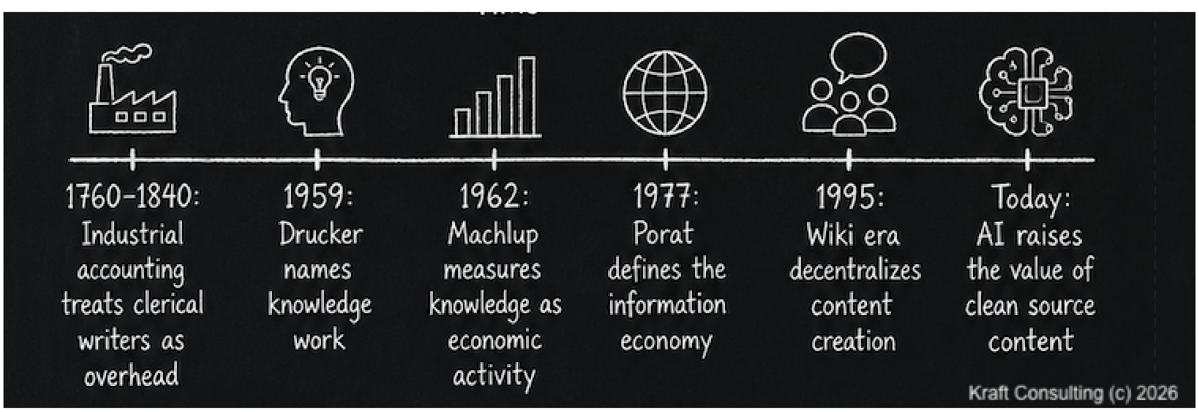

Figure 1: Timeline of accounting categories of business information sectors from the dawn of the industrial revolution through the information and AI revolution.

In older accounting models, this made technical writing easy to place under overhead. The company could see the labor cost. It had a weaker way to see the compounding value of well-governed technical knowledge. That classification became familiar, but familiarity is not the same as accuracy.

A technical publications group does more than produce documents. It creates reusable operating knowledge. That knowledge reduces support load, improves customer adoption, protects implementation quality, preserves release context, and gives new employees a path into complex systems. In fintech, where product behavior, transaction flow, API behavior, reconciliation logic, security controls, and compliance responsibilities are tightly connected, that knowledge becomes part of the operating system of the company.

The valuation gap begins when a company funds the creation of this knowledge, then manages the resulting files as a loose byproduct of labor instead of as a controlled business asset.

Payroll leakage is the CFO signal

Payroll leakage is the first measurable signal that the valuation gap has become material. In technology companies and other knowledge-worker organizations, payroll is one of the largest controllable operating expenses, and the lost time rarely sits inside one department. Engineers, product managers, technical writers, support teams, implementation teams, customer success, compliance, sales engineers, operations, and leadership all lose time when technical information is scattered, stale, duplicated, poorly named, or trapped in tools without a clear source of truth. The original draft identifies this loss directly: when information architecture is weak, staff can lose roughly 20 percent of their time to search and recovery work. In AI-assisted environments, that loss can climb higher because employees now spend paid time checking what AI agents retrieved, summarized, or drafted.

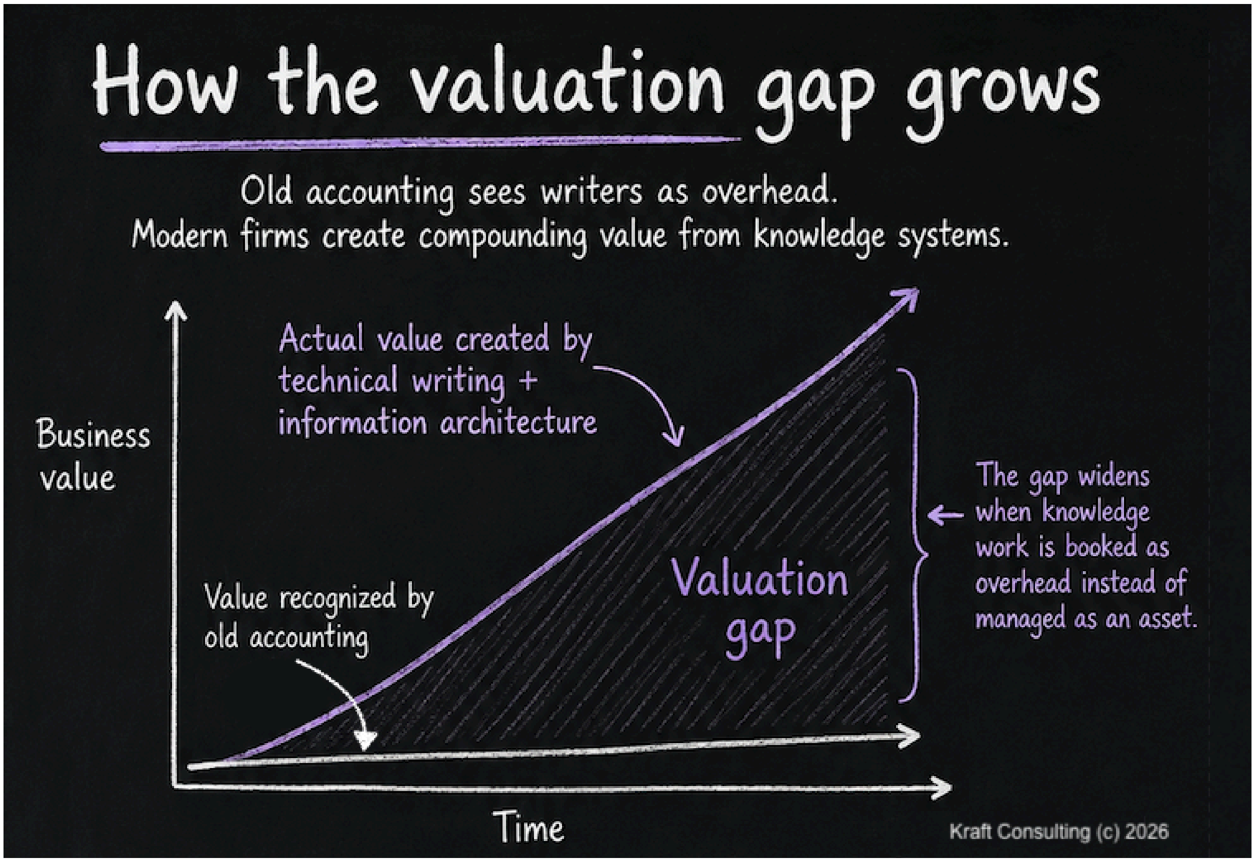

Figure 2: Describes organizations moving documentation and knowledge writers out of the past overhead model (office papers and clerical writers) into investment centers and knowledge assets.

From a CFO view, the issue is no longer whether writing is a cost. The issue is whether the company can recover full value from the technical knowledge it already paid trained people to create. Modern technical writers are not clerical writers producing office paperwork. They are highly trained technology workers who understand how information moves through software companies, platform companies, infrastructure teams, API programs, regulated systems, product organizations, and customer-facing technical ecosystems. Their work carries product truth, system behavior, release history, implementation logic, support reasoning, governance language, and customer-facing technical confidence.

That information behaves like a circulatory and interstitial system inside a technology company. It carries the nutrients of product knowledge into engineering, support, customer adoption, implementation, training, compliance, onboarding, release operations, and executive decision-making. When technical publications sit inside a cost-center model, the writers are relegated to overhead and the materials they create are also treated as overhead. The company then under-invests in the structure, ownership, metadata, lifecycle controls, and governance required to make those materials perform like business assets.

AI agents make this leakage more visible because they retrieve from the information environment as it exists. If the environment is fractured across Confluence, GitHub, SharePoint, Google Drive, Jira, Slack, PDFs, research folders, product notes, and individual working repositories, the AI agent may still produce a fluent draft. The organization then has to validate source authority, product truth, version scope, approved terminology, and release context. The time saved in drafting can be consumed downstream by review, correction, escalation, and rework. Payroll leakage becomes measurable because the company can see how much paid time is spent compensating for weak information architecture.

Why AI makes the gap larger

AI increases the value of clean technical knowledge because it can retrieve, summarize, compare, draft, classify, and reuse approved material quickly. It also increases the cost of weak information architecture because it scales retrieval from whatever structure the company gives it. A controlled documentation system gives AI a governed source layer with clear ownership, review status, version boundaries, approved terminology, access rules, and retrieval boundaries. In that kind of system, AI agents can help produce first drafts, summarize technical changes, build release notes, extract API details, compare decisions, identify gaps, and support onboarding with less downstream correction.

The business outcome changes when the source layer is fractured. In many technology companies, proposals live in Confluence, API references live in docs-as-code repositories, research papers live in individual GitHub accounts, decisions live in tickets, release facts live in pull requests and chats, customer guidance lives in knowledge bases, and legacy information remains in old drives or PDFs. AI retrieves from that operating reality. The output may look polished, but the company still needs paid experts to determine whether the answer is current, complete, approved, and safe to use.

This is how AI expands the valuation gap. It exposes the cost of weak source control and increases the visible burden of validation. The company expected productivity leverage, but it inherits another review queue layered on top of an already inefficient knowledge system. The core issue is retrieval confidence. AI can accelerate drafting, but drafting speed has limited value when the organization cannot trust the source material being retrieved.

For technology companies, this problem reaches across the full operating model. Software behavior, API logic, platform dependencies, implementation steps, configuration rules, release changes, security guidance, customer workflows, audit evidence, and support procedures all depend on current technical knowledge. When those sources are weakly governed, AI output becomes another item requiring expert review. The company pays for the original knowledge work, the AI tooling, the AI-generated draft, and the human validation layer required to make the draft safe.

Technical publications as operational infrastructure

Technical publications should be managed as operational infrastructure because technical knowledge is part of how technology companies run. It supports product adoption, platform stability, customer implementation, API use, support quality, release readiness, training, compliance, employee continuity, and customer trust. This does not require treating every document as a formal balance sheet asset. It requires managing technical knowledge with asset discipline, meaning the company knows what a file is, who owns it, which system it belongs to, which version it covers, whether it is current, which audience it serves, which source has authority, and whether it is eligible for AI retrieval.

That distinction separates a document collection from a knowledge system. A document collection stores files. A knowledge system returns trusted answers. A technical publications function with information architecture discipline becomes a control layer for operational knowledge. It protects the value the company already paid to create and makes that value easier to recover through engineering execution, customer support, onboarding, implementation, compliance review, sales support, partner enablement, and AI-assisted workflows.

In this sense, technical publications are not side materials attached to “real” product work. They are part of the operating body of the company. Product knowledge flows through them. When that flow is clear, the organization can move faster and make better decisions. When the flow is blocked by weak structure, stale ownership, scattered files, and conflicting versions, every downstream function compensates through search, meetings, rework, escalation, and interruption.

Use case: technical knowledge split across tools

Consider a technology company using AI agents as technical writers in a docs-as-code environment. The official API documentation lives in GitHub. Product proposals and implementation notes live in Confluence. Research papers and technical experiments live in individual repositories. Release details sit across pull requests, Jira tickets, Slack threads, and team notes. Customer-facing articles live in a help center. Internal troubleshooting steps live in support tools. Technical writers use AI agents to draft documentation, summarize changes, and retrieve background material.

At first, this looks efficient because AI reduces drafting time. Then the cost moves downstream. Engineers check whether the AI used current system behavior. Product managers verify whether a proposal became an approved decision. Technical writers compare research notes against production behavior. Support teams ask which guide customers should follow. Implementation teams rebuild instructions for customer-specific cases. Compliance or risk teams ask which file represents approved control language. The company sees faster drafts, but confidence in the source system remains slow.

From a CFO view, the issue is conversion. The company needs to convert technical knowledge into trusted operating leverage. When source authority is unclear, AI-generated drafts increase the need for review. When naming conventions are weak, search becomes expensive. When ownership is unclear, employees interrupt higher-cost experts. When lifecycle controls are missing, stale files remain active in the retrieval path. Payroll leakage becomes material because the cost is company-wide time spent compensating for an unmanaged knowledge system. The loss is not one flawed document. The loss is the recurring labor pattern created by unclear source authority.

Building a business case for leadership

Leadership needs the business case stated in financial terms: current leakage, recoverable value, and reduced operating risk. Start with a payroll leakage model that estimates the number of employees affected, their average fully loaded hourly cost, and the percentage of time spent searching, validating, recreating, or correcting technical information. A conservative case may use 20 percent, consistent with the original working estimate in the draft. In an AI-assisted but weakly governed environment, the current-state case may need to test 30 to 40 percent for the teams most affected, especially where engineers, technical writers, product owners, and support experts are repeatedly pulled into validation work.

The next step is operational evidence. Search analytics, ticket escalation patterns, onboarding time, review cycle time, duplicate file counts, release correction history, AI answer validation time, and unresolved ownership questions all help show where the loss occurs. This evidence should come from the functions that depend on technical knowledge every day: engineering, technical publications, product, implementation, support, compliance, customer success, sales engineering, operations, and leadership. The value of the business case grows when leadership can see that the company has already paid to create the knowledge and is now paying again to find it, verify it, rebuild it, and correct mistakes around it.

This shifts the discussion away from documentation cleanup and toward operating leverage. The stronger case is that technical knowledge is already a paid-for business asset, and poor structure prevents the organization from recovering full value from it. Information architecture becomes the control model that lets the company reclaim expert time, reduce support escalation, shorten review cycles, improve onboarding, reduce release correction work, lower audit preparation time, and make AI-assisted work more dependable.

Implementation model

The implementation should start with a documentation audit focused on business value and operational risk. The audit should identify which files are active, evergreen, duplicate, deprecated, archival, unowned, high use, high risk, and currently used by AI agents. This audit should also determine which files carry source authority and which files are research, proposal, draft, or historical context. That distinction is important because an old proposal, a research note, and a current API reference can all look useful to an AI system while carrying very different levels of operational authority.

After the audit, classify documents by business role. A source-of-truth API reference, a research note, an approved architecture decision, a release record, a customer instruction, a support guide, and a compliance evidence file should be managed as different kinds of business objects. Each has a different purpose, review requirement, lifecycle, access rule, and retrieval value. This is where information architecture begins to create financial value because it reduces the cost of ambiguity. Employees and AI systems need to know which material is approved, which material is provisional, which material is archival, and which material is suitable for customer, operational, or AI-assisted reuse.

Once the business role is clear, establish source authority. A technology company may decide that API behavior lives in docs-as-code, approved decisions live in structured decision records, proposals live in Confluence until approval, research lives in a controlled research archive, customer-facing operating procedures live in a governed documentation site, and support knowledge lives in a structured knowledge base with ownership and review cycles. The exact structure can vary by company, but each content category needs a defined home, owner, review path, and lifecycle. Without that control, the company keeps paying people to interpret the structure manually.

Metadata turns that structure into a working system. Each technical asset should carry the information needed to manage it, including owner, system, audience, lifecycle status, version, last reviewed date, review cycle, source authority, access level, and AI retrieval eligibility. This metadata is the bridge between human governance and machine retrieval. It gives employees a way to understand the status of a file and gives AI systems a cleaner way to retrieve material inside approved boundaries.

Naming conventions and folder rules make the structure usable in daily work. The system should make it clear where documents live, how files are named, how approved source material differs from research or proposals, and how employees or AI agents find the current answer. In a docs-as-code environment, this is where information architecture meets repository design. In a Confluence or knowledge base environment, this is where IA defines space structure, page hierarchy, labels, ownership rules, and lifecycle status. Across tools, the goal is the same: current knowledge should be easy to find, stale knowledge should be clearly marked, and approved source material should carry authority.

The RAG source layer should be built from approved material. Draft material, research material, old proposals, and archival files may remain available to humans with the right context, while approved material becomes the reliable source layer for AI retrieval. That distinction protects the business from giving AI the wrong authority. It also gives technical writers, engineers, support teams, and product teams a cleaner working system because they can separate exploratory knowledge from approved operating knowledge.

Governance keeps the system from decaying after the first cleanup. It defines who can create, approve, update, retire, archive, and verify technical content. It also defines review cycles, archival triggers, metadata requirements, and escalation paths when source conflicts appear. Measurement completes the model by tracking search time, document access rates, support ticket trends, review cycle time, onboarding time, duplicate file reduction, release correction rates, audit preparation time, and AI answer validation time. These metrics give leadership a way to see whether the company is recovering value from knowledge it already paid to create.

What leadership should fund

Leadership should fund the part of the system that converts technical knowledge into measurable operating leverage. That means funding the audit, repository cleanup, source authority model, metadata model, naming conventions, lifecycle governance, docs-as-code structure, Confluence boundary rules, research archive rules, RAG source layer, review workflows, AI retrieval controls, and reporting dashboard. These items should be treated as one operating control system, not a series of disconnected documentation chores.

This is an operational control project for technical knowledge. In technology companies and related knowledge-worker fields, that distinction matters because the knowledge system touches product behavior, customer trust, support quality, implementation speed, compliance readiness, employee productivity, investor confidence, and the practical value of AI investment. Leadership should fund information architecture because weak knowledge systems create payroll leakage, increase validation cost, slow execution, and reduce the return on AI tools.

The investment case is strongest when framed as value recovery. The company has already paid to create years of technical knowledge. Information architecture increases the usable return on that paid knowledge by making it governed, searchable, reusable, and safe for AI retrieval. In that sense, IA does not add a new layer of overhead. It recovers value from existing payroll, reduces waste in future payroll, and gives the business a cleaner operating base for AI-enabled work.

Summary

Technical publications create operational knowledge that companies use to build, run, scale, support customers, manage risk, and prepare for AI-enabled work. The valuation gap exists because companies pay to create that knowledge, then manage it through cost-center habits that weaken ownership, retrieval, reuse, lifecycle control, and measurement. The gap becomes visible through payroll leakage: paid employees spend recurring time searching, validating, recreating, and correcting information the company already funded.

AI makes the gap larger by increasing the value of clean source material and increasing the cost of fractured source systems. When AI agents are used while source material remains split across Confluence, GitHub, research folders, tickets, chats, personal repositories, knowledge bases, and informal files, drafting may speed up while payroll leakage grows through validation, correction, repeated search, and source-authority disputes.

Information architecture closes the gap by turning scattered technical files into managed operational infrastructure. It gives the company source authority, metadata, naming conventions, lifecycle governance, RAG-ready source material, and measurable recovery of paid knowledge value. Technical writing defines critical path processes and infrastructure. It should be managed as an operational knowledge asset because it reduces support costs, accelerates onboarding, protects product quality, and preserves institutional knowledge.

For leadership, this is the business case: fund the system that reduces payroll leakage, protects operating knowledge, improves retrieval confidence, and raises the return on AI investment.

References

RAG authority list

Primary publication sources ranked by notabile sorting criteria. Discipline-specific indexes rank above institutional mirrors. Downstream citations rank above general discovery pages only when they add substantive use.

Table valuation reference legend
• **Association for Computational Linguistics (ACL) Anthology** is the most authoritative source. • **NoDaLiDa/Baltic-HLT proceedings**, **DBLP**, **University of Helsinki**, and **University of Tartu DSpace** add credibility support. • **A. Nieminen's thesis** is the s useful downstream reference. • **Google Scholar** and **ResearchGate** explore discovery signals.

Top ranked source, first to last	Source background and reach	What it contributes	Notability judgment
ACL Anthology	Primary publication record for the paper. ACL Anthology is a major library for computational linguistics, speech, and NLP research, hosting more than 122,000 papers from official ACL and related venues. (ACL Anthology)	Provides the official citation, paper metadata, PDF, venue, pages, publisher, and citation count.	Most important source. Use this as the authoritative citation anchor.
NoDaLiDa/Baltic-HLT 2025 proceedings	Official conference proceedings for the Joint 25th Nordic Conference on Computational Linguistics and 11th Baltic Conference on Human Language Technologies. The proceedings are published by University of Tartu Library and indexed in ACL Anthology. (ACL Anthology)	Establishes the peer conference venue and formal proceedings context.	High authority. Use to show the paper is part of a recognized NLP and language technology research venue.

Top ranked source, first to last	Source background and reach	What it contributes	Notability judgment
DBLP computer science bibliography	DBLP is described by Schloss Dagstuhl as a premier open bibliographic database, search engine, and knowledge graph for computer science publications. (Home)	Indexes the paper within the computer science research ecosystem.	High discovery authority. Strong support that the paper is visible in computer science bibliographic infrastructure.
University of Helsinki research portal	Institutional research record. The University of Helsinki states it ranks among the top one percent of universities globally, and its portal records the paper as a peer-reviewed scientific conference contribution. (University of Helsinki)	Confirms peer-reviewed conference output, metadata, pages, series, and proceedings citation.	Strong institutional validation. Useful because it explicitly labels the work as peer-reviewed scientific conference output.
University of Tartu DSpace / University of Tartu Library	Official repository and publisher record for the proceedings item. The DSpace page lists the date, authors, publisher, file, handle URI, and collection. (DSpace)	Provides a stable institutional repository record and confirms the University of Tartu Library publisher.	Strong archival source. Use as secondary confirmation, after ACL.
University of Eastern Finland research systems / UEFConnect	Institutional author-affiliation record. UEFConnect lists Alexandre Nikolaev and the paper in his publication record. (UEFConnect)	Confirms author affiliation and institutional publication tracking.	Useful for author background. Less important than ACL, proceedings, DBLP, and publisher records.
A. Nieminen, *LLM Assistance for Technical Documentation Writing*	Downstream academic or thesis reference. The GUPEA record shows Nieminen citing de Souza, Nikolaev, and Koponen 2025. (Gupea)	Shows at least one later technical-document ation study is using the paper.	Important because it is an actual referring source. Reach appears narrower than ACL or DBLP,

Top ranked source, first to last	Source background and reach	What it contributes	Notability judgment
			but it is substantively relevant.
Google Scholar author profile / citation discovery	Google Scholar surfaces the paper under Karen de Souza's author profile and shows citation discovery signals. (Google Scholar)	Helps locate citation trails and related work.	Useful for discovery, but weaker as a publication authority. Do not rely on it as a primary source.
ResearchGate	ResearchGate surfaces the paper and author profile pages. It claims broad platform scale, including 25 million members, 160 million publication pages, and 2.3 billion citations. (ResearchGate)	Gives visibility and informal discovery.	High public reach, lower research authority. Use only as a discovery aid.

Top ranked source, first to last	Source background and reach	What it contributes	Notability judgment
ResearchGate topic pages for technical writing and software documentation	Topic pages list the paper under technical writing and software documentation publication feeds. (ResearchGate)	Shows topical surfacing within ResearchGate categories.	Lowest value. Useful only as a weak signal that the paper appears in relevant topic feeds.

ROI authority list

1. The ROI of Modern Document Management: Reducing costs and risks, understanding the real value and investment calculation for your document management system)
2. Information Architecture 101 for Technical Writers
3. Technical Documentation: Cost or Investment?
4. Best Practices for Technical Architecture Documentation
5. What is Knowledge Base Architecture?
6. How to Stop Wasting 20% of Your Day Searching for Information.
7. Does Your Workforce Spend Too Much Time Searching for Information?
8. What is Information Architecture and how can we benefit from it?
9. The importance of information architecture.
10. What Is Information Architecture & Why Does It Matter?
11. Here's how you can evaluate information architecture comprehensively
12. How can you measure the effectiveness of your information architecture?
13. How can you evaluate and compare different information architecture options?
14. How can you measure the success of your information architecture?
15. The Complexities of Measuring AI ROI
16. How can companies effectively measure the ROI of their business intelligence initiatives?
17. Top 5 Returns on Investment (ROI) from Document Management
18. "The High Cost of Not Finding Information", 2001 IDC Report, Susan Feldman and Chris Sherman, sponsored by Convera
19. "The Hidden Costs of Information Work", 2005 IDC Report, Susan Feldman et al., sponsored by Content Technologies.
20. "Information Is Useless If It Can't Be Found and Retrieved", 2018, Medium, Aaron Ganek
21. "Bridging the Information Worker Productivity Gap", 2012 IDC white paper, Melissa Webster, Sponsored by Adobe.

22. "Neglecting Small Data Risks Big Problems", 2014, Genetic Engineering and Biotechnology News, Ryan Sasaki
23. "Loses a document", 2017, LuitBiz DMS