

Article

The anatomy of collaborative open-source software development: A practice-based view on how open-source communities work

<http://dx.doi.org/10.11645/20.1.813>

Renesa Tamannum

Doctoral candidate, Åbo Akademi University, Finland. Email: renesa.tamannum076@gmail.com.
ORCID: [0009-0000-5719-561X](https://orcid.org/0009-0000-5719-561X)

José Apolinário Teixeira

Lecturer, Department of Computer Science, Aalto University, Finland. Email:
jose.apolinarioteixeira@aalto.fi. ORCID: [0000-0002-7195-4157](https://orcid.org/0000-0002-7195-4157).

Gunilla Widén

Rector, Åbo Akademi University, Finland. Email: gunilla.widen@abo.fi. ORCID: [0000-0002-4242-0455](https://orcid.org/0000-0002-4242-0455).

Abstract

Knowledge work is comprised of specialists who collaborate by exchanging expertise and skills to produce products and services; this is exactly what happens in open-source software development communities, albeit in a virtual and globally distributed manner. In this study, we examine two large and well-known open-source communities, specifically OpenStack and Automotive Grade Linux. More specifically, we take the theoretical notion of information literacy (IL) practice to analyse the day-to-day information-intensive activities of those communities. We collect and analyse naturally occurring publicly available digital trace data derived from the projects from an IL perspective. We report a set of ten information-intensive activities that are presented to an audience interested in IL and digital work, but not necessarily in software development. These ten activities are branching, committing, fetching, pushing, merging, reviewing, continuously integrating, gating, release management, and announcing — that collectively constitute the developers' information practice. While most of the reported information activities cannot be carried out in non-digital environments, the increasing trend towards the digitalisation of work reiterates the importance of studying the production and knowledge management practices in digital work.

This [Open Access](#) work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](#), allowing others to share and adapt this content, even commercially, so long as the work is properly cited and applies the same license. Copyright for the article content resides with the authors, and copyright for the publication layout resides with the Chartered Institute of Library and Information Professionals, Information Literacy Group.

Keywords

digital work; information landscape; information literacy; open-source software; virtual communities

1. Introduction

Knowledge work, as argued by Davenport & Prusak (2000) and (Drucker, 2017), and others, is comprised of specialists who collaborate by exchanging expertise and skills to develop products and services where the practice of collaboration ensures the effective share of specialised knowledge (Hetemi et al., 2022). As previously noted by Awazu & Desouza (2004), this is exactly what open-source communities do. In these communities, both professionals and hobbyists draw upon each other's skills and competencies to develop software in a virtual and globally distributed fashion (Fitzgerald, 2006; Li et al., 2025). Open-source communities are early practitioners of digital work, as open-source software developers develop digital products (i.e., software) by digital means (i.e., by relying heavily on computers connected to each other via the Internet). Open-source communities have a high societal impact, as the software developed in them can be found ubiquitously in, for example, airplanes, home appliances, mobile phones, and cars.

As recurrently pointed out across disciplines, many lessons can be learned from the open-source communities (Awazu & Desouza, 2004; Fitzgerald, 2011; Hourri, 2024; Markus & Agres, 2000; Munos, 2006; O'Reilly, 1999; von Krogh & von Hippel, 2006). These communities warrant investigation due to their increasing number and scope. Unlike communities in companies and NGOs, their work is carried out openly and transparently (Ince et al., 2012; Willinsky, 2005). Their work also leaves detailed traces of what was done, by who, when and how (Howison et al., 2006; von Hippel & von Krogh, 2003; Haese & Peukert 2025). Open-source communities and their digital traces have been explored across various disciplines and with several research methods - early work started in computer science and software engineering, but the current puzzle is now cross disciplinary (Raasch et al., 2013; von Krogh & Spaeth, 2007). As large open-source projects are often (1) globally distributed with the developer rarely meeting face to face, (2) do not follow the established project management frameworks and hierarchies seen in organisations, and (3) conducted in the commons where the participants give up intellectual property rights, the phenomenon challenges the predictions and explanations of existing theory in different fields (Teixeira et al., 2016; von Krogh & Spaeth, 2007).

In this study, we address prior calls for the study of information literacy (IL) as a situated socio-technical practice within novel contexts and domains (Lloyd, 2010a; Tuominen et al., 2005; Widén et al., 2014) by investigating two large and well-known open-source communities. The first, known as OpenStack, is a community with thousands of individuals and hundreds of organisations that jointly develop a complex open-source cloud computing infrastructure capable of handling big data. The second, known as Automotive Grade Linux, is also a community with thousands of individuals and hundreds of organisations that jointly develop a complex open-source platform for the automotive sector. While the former powers many of the cloud computing servers worldwide, the latter powers the dashboard and the media entertainment systems of many cars and trucks sold worldwide. For analysing the developers' information practice in those two large projects, we have taken the theoretical notion of IL practice (Lloyd, 2006, 2010b, 2017; Lloyd et al., 2013; Lloyd and Hicks, 2022) to analyse the

day-to-day information-intensive work in OpenStack and Automotive Grade Linux communities. Our goal is to juxtapose open-source software development with traditional closed information practices within organisations (Awazu & Desouza, 2004).

This study was carried out in a large cross-disciplinary research program addressing IL in the workplace. The focus of our work is on globally distributed virtual work. We collected and analysed publicly available and naturally occurring digital trace data derived from the OpenStack and the Automotive Grade Linux project. This data, created and maintained by developers in the public domain, allows us to trace back the individual and collective activities of different actors involved in the co-production of both projects – i) OpenStack, powering many data server centres worldwide and ii) Automotive Grade Linux, supporting in-vehicle systems in cars and trucks globally. As the key result of our research, we identified a set of ten information-intensive work activities that warrant presentation and explanation to an audience interested in IL and digital work, even if not specifically focused on software development. These ten activities - that collectively constitute the developers' information practice - are branching, committing, fetching, pushing, merging, reviewing, continuously integrating, gating, release management, and announcing.

Our main contribution is the description of ten information intensive work activities in open-source software development and the information landscape in which they are situated. Our research will further highlight the infrastructural role of the distributed version control system (and its design) as the orchestrator of the work conducted by a large body of geographically distributed open-source developers. Furthermore, we note that developers predominantly search for the information they need from the artifacts being produced (i.e., the software source-code). Surprisingly, their information-intensive practices exhibit modest dependence on rich-media and synchronous communication channels often seen as a requirement for effective production (Daft & Lengel, 1986; Kock, 2004; Workman et al., 2003). Lastly, we also note that the reported open-source information practices cannot be carried out in non-digital environments; however, the increasing trend towards the digitalisation of work practices (Harteis, 2018) reiterates the importance of studying open-source communities. After all, open-source communities have developed digital technologies by digital means already for several decades.

2. Theoretical background

2.1 IL and its modalities

Our research addresses IL within new emergent forms of work. More particularly, we study IL within the context of open-source software development - a form of distributed digital work. As the conceptual foundations of IL keep evolving with recurring discussions, and even disagreements, on what IL actually means (Bawden, 2001; Behrens, 1994; Secker, 2018; Stordy, 2015; Tuominen et al., 2005; Webber & Johnston, 2017), we set our theoretical orientation by endorsing the definition originally proposed in 2000 by the Association of College & Research Libraries (ACRL):

Information literacy is a set of abilities requiring individuals to recognize when information is needed and have the ability to locate, evaluate, and use effectively the needed information (ACRL, 2000).

Even if the ACRL no longer holds up the definition as official and it transitioned to a wider framework for IL (ACRL, 2016), which draws upon the concept of metaliteracy (Mackey & Jacobson, 2011, 2014), this particular definition suited our theoretical orientation towards the understanding of how open-source software developers search, evaluate and use information in practice.

Research on IL is, so far, particularly bounded by the educational context - a context where libraries play a very important role (Lloyd, 2013, 2017; Secker, 2014). Recent research, however, pointed out the importance of exploring the concept of the IL in non-educational work contexts (Caffrey 2021; Forster, 2015; Lloyd & Williamson, 2008; Mbatha, 2015; Secker, 2014, 2018; Weiner, 2011, among others). Here we take the position of IL as dependent and inseparable of the context in which it is investigated. In this sense and given the scarcity of research bridging IL with open-source software developers before attempting to measure their different facets.

Across disciplines, there has been a practice turn to theory building (Barnes, 2001; Feldman & Orlikowski, 2011; Schatzki 2025) and research in IL is no exception, with recurrent calls for the conceptualisation of socio-technical or sociomaterial practices within IL research (Choi, 2026; Lloyd, 2010a; Pilerot, 2016; Tuominen et al., 2005). Practices are often defined as shared actions of a collective of people (Orlikowski, 2002; Schultze, 2000). The sociologist Barry Barnes (2000), in the introduction chapter of the influential *The Practice Turn in Contemporary Theory* book (Barnes, 2001), gives a broad definition of practices as: "socially recognized forms of activity, done on the basis of what members learn from others, and capable of being done well or badly, correctly or incorrectly". Our view on practices is then closely related to Lloyd's position (Lloyd et al., 2013) on information practices. She sees them not as static arrays of routinised action, but as fluid and ongoing array of activities that are enabled or constrained by social sites and collective situated actions.

Information practice is thus seen as an array of information related activities and skills, constituted, justified and organised through the arrangements of a social site, and mediated socially and materially with the aim of producing shared understanding and mutual agreement about ways of knowing, and recognizing how performance is enacted, enabled and constrained in collective situated action. (Lloyd, 2011, p. 285).

Even if practices are often associated with behaviour enacted by humans, we also endorse material agency (Pickering, 2010): both humans and non-humans (books, desks, computer screens, software, internet) perform activities in the field of practice. For example, the main non-human partner in the investigated open-source communities under investigation is the distributed version control system (aka repository) that orchestrates the different pieces of code written by humans' developers. We see then practices as routinised actions enacted by both human and non-human agents. Besides engaging with the concepts of IL and information practice, our research also engages with the concept of information modality (Lloyd, 2017). Within a view of information practices as contextual and situated, each information modality refers to explicit and tacit information and ways of knowing.

The term information modality (Lloyd, 2006) describes the broader categories of information that represent the information environment. Information modalities describe

the common spaces created and accessed by people (for example, epistemic/instrumental space, social space, and physical space). (Lloyd, 2017, p. 95).

As evidenced by the work of Lloyd, the concept of information modality (Lloyd, 2006, 2017; Lloyd et al., 2013) fits well with the study of open forms of distributed digital work as it takes information practices as socially constructed (not individualised) and as contextual (not independent from its setting). Furthermore, the concept also assumes the materiality and corporeal embodiment of information.

From this perspective, information and ways of knowing are understood to reside in social relations that in turn reside in the practices that are intertwined with, and constitute, social life. These relationships influence the modalities of information (e.g., social, epistemic, and corporeal, (Annemaree Lloyd, 2010) and constitute the information landscapes of the setting (Lloyd et al., 2013).

Prior research by Lloyd (2017) distinguishes between three types of information modalities: (1) epistemic/instrumental modalities, (2) corporeal modalities, and (3) social modalities. The epistemic or instrumental modality refers to sources of information that are objective, factual, and reproducible. The corporeal modality encompasses embodied forms of knowing, including information derived from sensory experience and situated performance, which are often contingent and context specific. The social modality, in turn, concerns the nuanced forms of information embedded in social relations, including the often-unwritten norms, conventions, and expectations that shape practice and social exchange (Lloyd, 2017). From this perspective, individuals are not understood as passive recipients of information, detached from prior experience. Rather, they are viewed as critically engaged, self-reflective, and relational actors whose understanding of information is shaped through interaction with others and with information itself.

Recent research in IL has further strengthened this perspective by explicitly foregrounding sociomateriality as a central analytical lens. Instead of conceptualising IL as a set of individual competencies or skills, current study frames it as a sociomaterial practice enacted through the entanglement of people, technologies, infrastructures, and discourses (Haider & Sundin, 2023; Lloyd, 2023; Mård & Hallin, 2023). Within this view, IL emerges through participation in practices in which information, tools, and actors are mutually constitutive, and where agency is distributed across both human and non-human elements.

This development builds on earlier practice-theoretical foundations while extending them to account for digital platforms, automated systems, and information infrastructures that increasingly shape how information is produced, accessed, evaluated, and shared in contemporary distributed work settings (Haider & Sundin, 2023; Lloyd, 2023; Mård & Hallin, 2023). Such an approach is particularly well suited to the study of distributed digital work, where IL is enacted through sociomaterial arrangements rather than possessed by individuals in isolation.

2.2 Open-source communities

Open-source communities seek to extend the intellectual commons into the realm of computer software (for example the Linux operating system), and digital repositories (such as Wikipedia for encyclopaedic entries). Open-source software is released under permissive legal licenses

that keep the software and its source code open to the common heritage of all: everybody is free to use, study, modify, and redistribute copies of the software. Most collaborative work is carried out in a decentralised fashion across the globe over the Internet. Open sourcing has gained much multidisciplinary attention since 1980 (Crowston et al., 2012; Raasch et al., 2013; Zaggi, 2025). Some early voices emphasised the freedom and other philosophical aspects in software development (Stallman, 1985), while others emphasised the practical benefits of developing software in more open and transparent ways (Raymond, 2001). Some see it as a model for organising (Lanzara & Morner, 2005; Ljungberg, 2000), as a model for producing (Feller et al., 2008), as a model for innovating (Ulhøi, 2004; Von Hippel & Von Krogh, 2003), as a model for sourcing (Ågerfalk & Fitzgerald, 2008), as a model for coordinating geographically distributed teams (Bolici et al., 2016; Lindberg et al., 2016; Zaggi, 2025), or as a cultural movement whose values mimic library values (Iglesias, 2005; Shekgola, 2025).

In contrast with the traditional proprietary model where software is often developed, controlled and sold by a single organisation, open-source software is often developed within networks of individuals and firms that are interested in co-developing software that can be freely used, studied, modified and distributed for any purpose (Stallman, 1985). The development of open-source software (e.g., the Linux operating system or R statistical package) contrasts from the more traditional open-source software counterparts (e.g., Microsoft Windows or IBM SPSS) by being more inclusive, transparent while lessening intellectual property rights (Awazu & Desouza, 2004; Markus & Agres, 2000; Teixeira, 2015). Another characteristic of open-source is that participants are geographically distributed and rarely meet physically.

In sharp contrast with the idea that intellectual property rights drive the knowledge economy, open-source communities emphasise the values of collaboration, transparency, and inclusiveness over the ownership of the software (Teixeira, 2015). Some consider that with the success of open-source software, 'openness' spread outside the confines of its original software boundaries. Nowadays, we do not just talk about open-source software, but also about open data, open hardware, open standards, open knowledge, open innovation, open access, open science, among many other manifestations of the "open phenomena" (Powell, 2012; Schlagwein et al., 2017; Sher Decusatis et al., 2012; Tkacz, 2012; West et al., 2014; West & Kuk, 2016, among others).

Many have argued before that cross-disciplinary lessons can be learned from the open-source communities (Awazu & Desouza, 2004; Fitzgerald, 2011; Hourri, 2024; Li et al., 2024; Markus & Agres, 2000; Munos, 2006; O'Reilly, 1999; von Krogh & von Hippel, 2006). Such communities produced very successful technologies such as the Linux operating system, which is widely deployed in data-centres and high-performance computers, the Apache webserver, that hosts most of the pages on the Internet; and the WebKit web-browsing technologies, that power most modern mobile devices. Like in other disciplines, the library and information science can learn from the success of open-source software and their *modus operandi* (Awazu & Desouza, 2004; Câmara & Fonseca, 2007; Shekgola, 2025; Oberg, 2003; Tidal, 2017).

2.3 Coordination in open-source software

Coordination in the production of software can be understood as the management of dependencies among the tasks of distributed contributors (Malone & Crowston, 1994). Given the networked, virtual and global distributed nature of open-source software, scholars from different disciplines investigated coordination in this specific context (e.g., Bolici et al., 2016;

Howison & Crowston, 2014; Shaikh & Henfridsson, 2017; Zaggl, 2025). In the absence of formal managerial hierarchies, this coordination is achieved through a distinctive blend of socio-material processes. As Shaikh and Henfridsson (2017) argue, governance in OSS is enacted through dynamic coordination processes rather than static structures, where the rules and norms for collaboration are continually reproduced and adjusted through everyday project interactions. This perspective aligns with the concept of "open superposition," where work is openly layered upon a shared digital artifact, allowing contributors to observe, build upon, and integrate each other's efforts without requiring extensive prior communication (Howison & Crowston, 2014). Along these lines the transparency of open-source software artifacts reduce the need for direct instruction and enables a form of stigmergic coordination, where the artefacts itself—the evolving codebase—signals necessary actions and modulates collective behaviour (Bolici et al., 2016).

A central theoretical tension in understanding this coordination lies between artifact-driven and authority-driven mechanisms. Artifact-driven coordination, as explored by Zaggl (2025), posits that the shared work product (e.g., the source code, its modular architecture, and associated tools like version control systems) serves as the primary medium for organising work. Contributors coordinate implicitly by reading, modifying, and responding to changes in the artifact, a process that minimises direct communication costs and facilitates scalable collaboration. This is contrasted with authority-driven coordination, which relies on the hierarchical decisions of designated individuals, such as module maintainers or release managers, to resolve conflicts, set direction, and integrate contributions. Zaggl (2025) finds that while artifact-based coordination generally lowers the costs of propagating changes through the system, effective projects often strategically blend both forms; authority-based mechanisms become crucial for managing complex interdependencies and ensuring architectural integrity where purely artifact-driven signals are insufficient.

Thus, the coordination landscape of successful large-scale open-source projects is a hybrid one. It leverages the efficiencies of stigmergic, artifact-based interaction for daily development, enabled by tools like Git that make the work of all contributors visible and composable. Simultaneously, it incorporates lightweight but critical authority-based governance to manage decisions that cannot be resolved through the artifact alone, such as architectural pivots or contentious proposals. This synthesis allows open-source communities to maintain both openness and coherence, enabling thousands of geographically dispersed individuals to collaboratively build complex, mission-critical software systems without traditional corporate oversight (Bolici et al., 2016; Shaikh & Henfridsson, 2017; Zaggl, 2025).

3. Methodological Design

This research is embedded within a large multi-year research program dealing with IL at the workplace. Our sub-project pays special attention to global and geographically distributed work. In this research, we scrutinise the software development practices of the two open-source communities developing a very complex infrastructure capable of handling big data in heterogeneous computing environments. Both OpenStack and Automotive Grade Linux, communities with hundreds of firms and thousands of individuals jointly developing software in the open-source arena, were our fields of practice and social site under investigation (Schatzki et al., 2001).

As our goal encompassed investigating open-source software development from an information practice perspective to an audience not necessarily interested in software development, the initial research question “What are the information-intensive activities of open-source software development that allow them to jointly develop and maintain large-scale, complex software systems?” framed our investigation - this in the context of open, large, and global, computer-supported online communities.

We take a cross-disciplinary approach by design to leverage the distinct disciplinary orientation of the authors (from ‘Software Engineering’ to ‘Information Systems’ and ‘Library and Information Science’). Therefore, we attempted to cross the different branches, disciplines, sub-disciplines, and specialties of scientific knowledge (Morillo et al., 2003). Prior years of engagement with the OpenStack and the Automotive Grade Linux communities led to related research (Teixeira, 2014, 2017, 2023; Teixeira et al., 2015, 2016; Teixeira & Hyrynsalmi, 2017; Teixeira, J., & Karsten 2019) and expedited our research efforts.

We collected and analysed qualitative publicly available and naturally occurring archival data derived from the OpenStack and the Automotive Grade Linux projects. This kind of data is not a consequence of our own actions as researchers, but they are created and maintained by the two open-source communities in their own pursuits of producing large and complex open-source cloud computing and automotive infrastructures. We have considered key guidelines on how to conduct qualitative empirical research online (Kozinets, 2009, 2012). From the initial sources of online data, we were forced to follow many links to collect further information related to knowledge management. First for the OpenStack project and then later for the Automotive Grade Linux project to further extend the initial research. The collected data captures not only the OpenStack and Automotive Grade Linux software ecosystems but also their evolution. Several official Internet pages, wikis and blogs created by the community to guide the joint development of both projects were analysed (see for example OpenStack¹ or AGL²). The software development documentation guiding the day-to-day work of those contributing to the open-source project also guided our research efforts. Overall, the collected data, left by the developers under the public domain, allows us, to trace back and understand certain elements of the individual and collective digital behaviour of developers from their interaction with digital systems (Choi, 2020; Pentland et al., 2020).

As Lindberg et al. (2016) and Howison & Crowston (2014) who relied on similar data but from other open-source projects, we crossed observation (direct and retrospective) with some computational approaches to make sense of the day-to-day activities of open-source software developers. In this sense we align very much with the ideas of Venturini and Latour (Venturini & Latour, 2009) that interactions that pass through digital media leave traces that can be easily recorded, massively stored and inexpensively retrieved.

The construction of social phenomena implies tracking each of the actors involved and each of the interactions between them. This was an impossible goal just a dozen years ago, but it starts to become more and more realistic as digital technologies spreads. (Venturini & Latour, 2009).

¹ https://wiki.openstack.org/wiki/How_To_Contribute

² <https://www.automotivelinux.org/>

Leveraging prior research efforts (Teixeira, 2014, 2017, 2023; Teixeira et al., 2015, 2016; Teixeira & Hyrynsalmi, 2017) allowed us to get access to software developers, and get familiar with their ways of working. We turned to theory on IL to unveil and analyse recurrent work activities. Our results are grounded from observable, regularly occurring user activities and routines. Our awareness on this recurrent work activities emerged from the collected trace data, where repetition patterns emerged. Furthermore, documentation developed by the software developers themselves as well as documentation of tools used by the same software developers, called our attention to recurrent work activities established in both communities. Studying the social sites for a long period of time was crucial, as such working practices follow cycles of three to six months. One cycle often starts with practices related to actual functional development, and cycles often end with distinct activities dealing more with quality assurance.

Two software developers, each with a track record of contributing to one of the investigated projects, were invited to comment to enhance the validity of our interpretations. They were also invited to reflect on their own work activities and to comment a preliminary version of our results section.

4. Results

As the key output of our research efforts, we identified ten recurrent work activities that we think are particularly important. These activities are valuable both to people interested in information practices in general and information practices in geographically distributed digital environments more particularly. These key ten recurrent activities are branching, committing, fetching, pushing, merging, reviewing, continuously integrating, gating, release management and announcing. While many of the other identified activities are very contextual, dependent on the actor's role (e.g., coder, tester or user-interface designer), on the technologies in use (e.g., programming language, hardware), the artifacts' functionality (e.g., a new software feature, a new user-interface design or a new robot testing other digital artifacts), and the level of engagement with the project (e.g., full-time, part-time, occasional contributor), these ten activities here reported were found to be recurrent across the two projects overall. In other words, we found that independently of the type of work being done (e.g., software, testing, design or documentation) and its context (e.g., situated goals, technology, affiliated company), it is very likely that all actors involved in the two communities experience these ten activities on a regular basis.

In our view, the following ten activities are worthy being reported and explained to an audience who might not be interested in software development. Therefore, we here attempted to understand, analyse, report and explain the following activities recurrently enacted in our investigated open-source communities.

branching an activity in which developers isolate their own work (in their own computer) from the commonly shared work base (distributed across developers). An activity much influenced by the technological artifacts.

committing an activity in which developers communicate to other developers what they changed (in their own computer) and why. An activity much influenced by the education, the workplace, the community, and the technological artifacts.

fetching an activity in which developers receive all the changes performed by the other developers (distributed across the globe) to their own computer. An activity much influenced by the technological artifacts.

pushing an activity in which developers share their own changes (in their own computer) to the other developers (distributed across the globe). An activity much influenced by the technological artifacts.

merging an activity in which developers integrate their most recent work (e.g., a new feature implemented in their own computer) with the commonly shared work base (distributed across developers). An activity much influenced by the community, the workplace, and the technological artifacts.

reviewing an activity in which new contributions (i.e., proposed changes to the commonly shared work base) are both reviewed and evaluated by other developers and several robots. An activity much influenced by the education, the community and the technological artifacts.

continuously integrating an activity in which humans and robots continuously integrate and test new contributions at a very fast pace. An activity much influenced by the community and the technological artifacts.

gating an activity in which humans and robots keep low-quality contributions away from the officially shared work base. An activity much influenced by the education, the workplace, the community, and the technological artifacts.

release management an activity in which lead-developers decide what contributions are good enough to be included in a forthcoming official product version (a stable commonly shared work base). An activity much influenced by the workplace, the community, and the technological artifacts.

announcing after contributions are recurrently tested, reviewed and evaluated by humans and robots and finally integrated with a stable commonly shared work base. The activity of announcing, communicating and marketing new contributions (new landings in the official master version of the project repository). An activity much influenced by the workplace, the community, and the technological artifacts.

While the ten identified activities are presented here as distinct analytical categories, within the living context of the open-source project they do not occur as isolated events. Instead, they form a deeply processual and temporally structured workflow that governs the entire lifecycle of a contribution. The journey of a single code change typically begins with branching and committing, moves through the synchronous gatekeeping of fetching, pushing, and merging, enters the critical and often iterative phase of reviewing and gating, and only after successfully navigating continuous integration tests does it culminate in release management and eventual announcing. This sequence is not merely linear but cyclical and recursive; a contribution that fails gating may be sent back for further committing, while the announcing of one release simultaneously marks the starting point for the next development cycle. Furthermore, these activities are tightly coupled to the project's cadence of three-to-six-month release cycles,

creating predictable peaks in reviewing and integration labour as milestone deadlines approach. Thus, the information practice of these developers is not a static set of skills, but a temporally orchestrated performance where the meaning and urgency of each activity shift according to its position in the collective production timeline.

Most of the reported activities can be easily mapped to Git (i.e., the version control and distributed revision control system created by Linus Torvalds to support the development of Linux). In the Git reference manual³, we can find information about the git branch, git commit, git fetch, git push, and git merge commands that developers use daily. We note, however, that the practices of continuously integrating, gating, release management, and announcing relied mostly on other systems, often integrated with Git. Furthermore, there is a great degree of cross-project customisation, and the actual activities can vary from project to project. For example, the OpenStack community started relying more on the Git flow project⁴ over the standard Git for merges and reviews. In another example, some of the basic functionalities for peer review on the Automotive Grade Linux project were heavily customised and handled in part by other tools such as Gerrit⁵. To sum up, being able to work with Git is fundamental for open-source software developers to co-produce their work in a global collaborative way. However, from project to project, there are customisations and workflows integrate Git with other systems.

As mentioned earlier in the methodology, we asked two software developers from the projects under investigation to comment on our recurrent activities. One of the developers called our attention to the gating activity (a process in which non-conforming or low-quality developments are turned down) - many contributions submitted to the project do not open the number of gates required for its integration within the code base. Much of the performed work is “turned down”, and, therefore, not included in the official OpenStack releases (i.e., an official software version). Such “sent back” work needs to be improved by the developers so that it can open all the required gates to be included in a later release, becoming then visible to other developers, users, customers and researchers such as us. The same activity of ‘gating’ did not emerge from the analysis of the Automotive Grade Linux project; however, both communities do cover the activity of ‘reviewing’ which has a more positive connotation attached to it. Both informants also pointed out that the larger the pool of contributors, the higher the ‘reviewing’ load as all contributions to the project need to be reviewed by more experienced developers. In this sense, open-source code reviews, and academic peer review share some parallels.

5. Discussion

Our main contribution is the empirical grounded description of ten work activities, extending prior work on IL and modalities of information (Lloyd, 2006, 2010b, 2017; Lloyd et al., 2013; Špiranec & Kos, 2013). As noted in recent studies on artifact-based coordination in open-source software highlight that developers strongly rely on shared digital artifacts to organise their work, rather than on direct communication or hierarchies (Howison & Crowston, 2014; Zaggi, 2025). Therefore, technological artifacts should not be merely considered as an output of work activities. Instead, they hold active inscriptions of practical knowledge that support collective production efforts. Technological artifacts are intertwined with all the observed work activities

³ <https://git-scm.com/docs>

⁴ <https://git-flow.sh/docs/about/>

⁵ <https://www.gerritcodereview.com/about.html>

and act as a source of information by themselves that influences the developer's work and coordination. When analysing the practice of opensource software development at the empirical level, researchers should not consider technological artefacts merely as the output of work activities. Instead, they should consider technological artifacts as holding information that shapes the behaviour of software developers.

Informed by the selected theory background and by following the digital traces left by software developers doing their work (i.e. by retrospectively analysing collective joint development efforts of the developers), we could recognise four modalities of information. In addition to the three early recognised modalities of information broadly proposed by Lloyd (2017) - the epistemic/instrumental, the corporeal, and the social modalities - we identified a fourth information modality that we here label as 'technological artifacts'. By focusing on the literacies of information at the empirical level, and while grasping the elements of the practice under investigation, we could not fully explain the work activities of the developers by solely looking at their formal education, the workplace, and community aspects, or the corporeal dimension. Most of the investigated work activities deal directly with the technological artifacts being developed. Even if many say that in open-source communities 'code is king', the commonly shared work base includes not only the software source code (aka codebase) but also other technological artifacts such as documentation, automated tests, and infrastructure (e.g., software robots that continuously integrate, build and test what was developed so far).

Furthermore, our findings also reinforce the early view on work artifacts as 'dynamic vehicles' that make knowledge creation, accumulation, and dissemination possible throughout the Internet (Lanzara & Morner, 2003, 2005). Furthermore, we also reinforce views on artifacts as enacting stigmergic mechanisms that enable coordination without discussion or direct communication (Bolici et al., 2016; Crowston et al., 2007; Robles et al., 2005). In the cases of OpenStack and Automotive Grade Linux, where independent and organisational affiliated developers work together, such technological artifacts are available to any user or potential contributor independently of any age, group, organisational affiliation, and field of endeavour. They belong in an intellectual property regime that allows the incremental innovation that characterises open-source communities (Awazu & Desouza, 2004). Under the traditional software development (i.e., the proprietary way), such work artifacts are not available. Instead, they are controlled and protected with intellectual property mechanisms. In this sense, proprietary software limits the information modalities to potential improvers to the software.

Almost all the recurrent activities found in both projects are orchestrated by Git, the version control and distributed revision control system supporting distributed, non-linear workflows in the projects. In each of these activities, different actors within the studied projects interact directly with artefacts orchestrated by Git. Being able to work with Git is, therefore, a required skill to contribute directly to the two projects. As researchers, we needed to learn how Git works before making sense of how developers co-produce their work in a global collaborative way.

The investigated open-source developers interact with the Git system as part of their day-to-day work. This is consistent with prior research highlighting the role of version control and distributed revision control systems as orchestrating the work activities of open-source software developers (Lanzara & Morner, 2003, 2005; Teixeira & Karsten, 2019). While in some open-source projects the version control and distributed revision control system are two different systems, in our two projects, the Git system integrated both systems in one. As reported by Davis (2015) 'learning

version control is ... a key skill for the libraries and archives of tomorrow” and “Git makes collaborative work much easier” (Davis, 2015).

Even if developers work mostly alone in front of their own computers in a geographically distributed setting with little face-to-face interaction, they are still doing a vast collective effort. Surprisingly, such collective efforts exhibit modest dependence on rich-media and synchronous communication channels traditionally seen as a requirement for effective production (Daft & Lengel, 1986; Kock, 2004; Müller et al., 2026; Workman et al., 2003). It is also important to notice that the practice here described require artifacts to move freely from worker to work with low latency. The work base is jointly developed, as well as recent changes to it, move to and from the developers’ computers very frequently, often several times a day. As in open-source communities, the work base is fully digital and can move around at a very fast pace from computer to computer over the Internet. Such practice and its activities would not work with physical work bases (e.g., a building, an airplane, or a ship). Even if most of the here reported work activities cannot be carried out in non-digital environments, the increasing trend towards the digitalisation of work activities (Harteis, 2018) reiterates the importance of studying the information-intensive information practices of open-source communities. After all, their work activities are nearly fully digitalised already.

6. Conclusions

Many have argued that many cross-disciplinary lessons can be learned from the open-source communities (Awazu & Desouza, 2004; Fitzgerald, 2011; Hourri, 2024; Markus & Agres, 2000; Munos, 2006; O’Reilly, 1999; von Krogh & von Hippel, 2006 ; Zaggl, 2025). In our view, open-source communities are pioneers of digitalisation as since their inception they develop digital artifacts primarily by digital means. In this study, we investigated the information-intensive information practices of two large open-source communities. The first known as OpenStack as a community of individuals and organisations that jointly develop a complex open-source cloud computing infrastructure; and the second known as Automotive Grade Linux as jointly developing a complex open-source platform that is integrated in cars.

Our research efforts led to the recognition of ten key information-intensive work activities that merit attention from audiences not specifically focused on software development. The reported information activities embedded in the day to day of the development efforts are branching, committing, fetching, pushing, merging, reviewing, continuously integrating, gating, release management, and announcing. While most of these activities constituting the practice of open-source software development cannot be carried out in non-digital environments, the increasing trend towards the digitalisation of work practices reiterates the importance of studying information-intensive information practices in digital work.

When analysing the reported activities from the perspective of IL and information modalities (Lloyd, 2006, 2010b, 2017; Lloyd et al., 2013;) we argue that technological artifacts should not be merely considered as an output of work activities. Instead, they hold active inscriptions of practical knowledge that support the collective production efforts. In the observed software development context, technological artifacts are intertwined within the day-to-day digital work activities. As technological artifacts as holding information that shapes the behaviour of software developers, they should be considered an information modality by itself. When investigating the development of complex technology, scholars should not consider technological artifacts merely

as the output of work activities nor as a medium solely for holding information - instead, they should also consider the potential of technological artifacts as an information modality that shape social practice.

Declarations

Ethics approval

Ethical approval was not considered necessary in alignment with Åbo Akademi University or Aalto University's guidance on the conduct of ethical research.

Funding

This work was partially supported by the Research Council of Finland (formally known as Academy of Finland), project The Impact of Information Literacy in the Digital Workplace [grant number 295743].

AI-generated content

AI tools were used for proofreading.

References

- ACRL. (2000). [ACRL standards: Information literacy competency standards for higher education](#). *College & Research Libraries News*, 61(3), 207–215.
- ACRL. (2016). [Framework for information literacy for higher education](#). Association of College and Research Libraries.
- Ågerfalk, P. J., & Fitzgerald, B. (2008). [Outsourcing to an unknown workforce: Exploring opensourcing as a global sourcing strategy](#). *MIS Quarterly: Management Information Systems*, 32(2), 385–409.
- Awazu, Y., & Desouza, K. C. (2004). [Open knowledge management: Lessons from the open source revolution](#). *Journal of the American Society for Information Science and Technology*, 55(11), 1016–1019.
- Barnes, B. (2001). [Practice as collective action](#). In T. R. Schatzki, K. K. Cetina, & E. von Savigny (Eds.), *The practice turn in contemporary theory*. Routledge.
- Bawden, D. (2001). [Information and digital literacies: A review of concepts](#). *Journal of Documentation*, 57(2), 218–259.
- Behrens, S. J. (1994). [A conceptual analysis and historical overview of information literacy](#). *College & Research Libraries*, 55(4), 309–322.
- Bolici, F., Howison, J., & Crowston, K. (2016). [Stigmergic coordination in FLOSS development teams: Integrating explicit and implicit mechanisms](#). *Cognitive Systems Research*, 38, 14–22.

- Câmara, G., & Fonseca, F. (2007). [Information policies and open source software in developing countries](#). *Journal of the American Society for Information Science and Technology*, 58(1), 121–132.
- Crowston, K., Li, Q., Wei, K., Eseryel, U. Y., & Howison, J. (2007). [Self-organization of teams for free/libre open source software development](#). *Information and Software Technology*, 49(6), 564–575.
- Crowston, K., Wei, K., Howison, J., & Wiggins, A. (2012). [Free/libre open-source software development: What we know and what we do not know](#). *ACM Computing Surveys*, 44(2), 7.
- Daft, R. L., & Lengel, R. H. (1986). [Organizational information requirements, media richness and structural design](#). *Management Science*, 32(5), 554–571.
- Davenport, T. H., & Prusak, L. (2000). *Working knowledge: How organizations manage what they know*. Harvard Business Press.
- Drucker, P. (2017). *The age of discontinuity: Guidelines to our changing society* (2nd ed.). Routledge.
- Feldman, M. S., & Orlikowski, W. J. (2011). [Theorizing practice and practicing theory](#). *Organization Science*, 22(5), 1240–1253.
- Feller, J., Finnegan, P., Fitzgerald, B., & Hayes, J. (2008). [From peer production to productization: A study of socially enabled business exchanges in open source service networks](#). *Information Systems Research*, 19(4), 397–521.
- Fitzgerald, B. (2006). [The transformation of open source software](#). *MIS Quarterly*, 30(3) 587–598.
- Fitzgerald, B. (2011). [Open source software: Lessons from and for software engineering](#). *Computer*, 44(10), 25–30.
- Forster, M. (2015). [Refining the definition of information literacy: The experience of contextual knowledge creation](#). *Journal of Information Literacy*, 9(1), 62–73.
- Gherardi, S. (2006). *Organizational knowledge: The texture of workplace learning*. Blackwell Publishing.
- Haider, J., & Sundin, O. (2023). [Sociomateriality](#). In A. Hicks, A. Lloyd, & O. Pilerot (Eds.), *Information literacy through theory* (pp. 149-164). Facet Publishing.
- Harteis, C. (Ed.) (2018). [The impact of digitalization in the workplace: An educational view](#). Springer International Publishing.
- Hetemi, E., Pushkina, O., & Zerjav, V. (2022). [Collaborative practices of knowledge work in IT projects](#). *International Journal of Project Management*, 40(8), 906–920.

- Hourri, Y. (2024, November 21-22). [Towards understanding open-source software communities](#). The 23rd Belgium-Netherlands Software Evolution Workshop, Belgium.
- Howison, J., Conklin, M., & Crowston, K. (2006). [FLOSSmole: A collaborative repository for FLOSS research data and analyses](#). *International Journal of Information Technology and Web Engineering (IJITWE)*, 1(3), 17–26.
- Howison, J., & Crowston, K. (2014). [Collaboration through open superposition](#). *MIS Quarterly*, 38(1), 29–50.
- Iglesias, E. (2005, April 7-10). *Open source culture*. ACRL 12th National Conference, Minnesota, United States.
- Ince, D. C., Hatton, L., & Graham-Cumming, J. (2012). [The case for open computer programs](#). *Nature*, 482(7386), 485–488.
- Kock, N. (2004). [The psychobiological model: Towards a new theory of computermediated communication based on Darwinian evolution](#). *Organization Science*, 15(3).
- Kozinets, R. V. (2009). *Netnography: Doing ethnographic research online*. SAGE Publications Ltd.
- Kozinets, R. V. (2012). [Marketing netnography: Prom/ot\(Ulqat\)ing a new research method](#). *Methodological Innovations Online*, 7(1), 37–45.
- Lanzara, G. F., & Morner, M. (2003, July 3-5). [The knowledge ecology of open-source software projects](#). 19th EGOS Colloquium, Copenhagen, Denmark.
- Lanzara, G. F., & Morner, M. (2005). Artifacts rule! How organizing happens in open source software projects. In B, Czarniawska, T. Hernes,(Eds.), *Actor-network theory and organizing*. Liber; Oxford University Press.
- Lindberg, A., Berente, N., Gaskin, J., & Lyytinen, K. (2016). [Coordinating interdependencies in online communities: A study of an open source software project](#). *Information Systems Research*, 27(4), 751–772.
- Ljungberg, J. (2000). [Open source movements as a model for organising](#). *European Journal of Information Systems*, 9(4), 208–216.
- Lloyd, A. (2005). [Information literacy: Different contexts, different concepts, different truths?](#) *Journal of Librarianship and Information Science*, 37(2), 82–88.
- Lloyd, A. (2006). [Information literacy landscapes: An emerging picture](#). *Journal of Documentation*, 62(5), 570–583.
- Lloyd, A. (2010). [Framing information literacy as information practice: Site ontology and practice theory](#). *Journal of Documentation*, 66(2), 245–258.

- Lloyd, A. (2010). Information literacy landscapes: Information literacy in education, workplace and everyday contexts. Chandos Publishing.
- Lloyd, A. (2011). [Trapped between a rock and a hard place: What counts as information literacy in the workplace and how is it conceptualized?](#) *Library Trends*, 60(2), 277–296.
- Lloyd, A. (2013). [Building information resilient workers: The critical ground of workplace information literacy. What have we learnt?](#) In S. Kurbanoglu, E. Grassian, D. Mizrachi, R. Catts, & S. Špiranec (Eds.), *Worldwide commonalities and challenges in information literacy research and practice* (pp. 219–228). Springer International Publishing.
- Lloyd, A. (2017). [Information literacy and literacies of information: A mid-range theory and model.](#) *Journal of Information Literacy*, 11(1), 91–105.
- Lloyd, A. (2023). [Information literacy and the social: Applying a practice theory view to information literacy.](#) In A. Hicks, A. Lloyd, & O. Pilerot (Eds.), *Information literacy through theory*. (pp. 27-28) Facet Publishing.
- Lloyd, A., Anne Kennan, M., Thompson, K. M., & Qayyum, A. (2013). [Connecting with new information landscapes: Information literacy practices of refugees.](#) *Journal of Documentation*, 69(1), 121–144.
- Lloyd, A., & Williamson, K. (2008). [Towards an understanding of information literacy in context: Implications for research.](#) *Journal of Librarianship and Information Science*, 40(1), 3–12.
- Mackey, T. P., & Jacobson, T. E. (2011). [Reframing information literacy as a metaliteracy.](#) *College & Research Libraries*, 72(1), 62–78.
- Mackey, T. P., & Jacobson, T. E. (2014). *Metaliteracy: Reinventing information literacy to empower learners*. American Library Association.
- Mård, M., & Hallin, A. (2023). [Digital literacy in a post-digital era: Rethinking 'literacy' as sociomaterial practice.](#) In G. Widén & J. Teixeira (Eds.), *Information literacy and the digitalisation of the workplace* (pp. 15–28). Facet Publishing.
- Markus, M. L., & Agres, B. M. C. E. (2000). [What makes a virtual organization work?](#) *MIT Sloan Management Review*, 42(1), 13.
- Martin, J. (2013). [Refreshing information literacy: Learning from recent british information literacy models.](#) *Communications in Information Literacy*, 7(2), 6.
- Mbatha, B. (2015). [Mapping and auditing digital literacy of civil servants in selected South African government departments.](#) *Journal for Communication Studies in Africa*, 34(1), 49–64.

- Morillo, F., Bordons, M., & Gómez, I. (2003). [Interdisciplinarity in science: A tentative typology of disciplines and research areas](#). *Journal of the American Society for Information Science and Technology*, 54(13), 1237–1249.
- Müller, K., Harutyunyan, N., Riehle, D., & Koch, C. (2026). [Best practices for work from home: A qualitative survey in open source and distributed software development](#). *Information and Software Technology*, 190, 107952.
- Munos, B. (2006). [Can open-source R&D reinvigorate drug research?](#) *Nature Reviews Drug Discovery*, 5(9), 723–729.
- Nicolini, D. (2012). *Practice theory, work, and organization*. Oxford University Press.
- Oberg, S. (2003). [Open source software](#). *Serials Review*, 29(1), 36–39.
- O'Reilly, T. (1999). [Lessons from opensource software development](#). *Communications of the ACM*, 42(4), 32–37.
- Orlikowski, W. J. (2002). [Knowing in practice: Enacting a collective capability in distributed organizing](#). *Organization Science*, 13(3), 249–273.
- Pickering, A. (2010). *The mangle of practice: Time, agency, and science*. The University of Chicago Press.
- Pilerot, O. (2016). [A practice-based exploration of the enactment of information literacy among PhD students in an interdisciplinary research field](#). *Journal of Documentation*, 72(3), 414–434.
- Powell, A. (2012). [Democratizing production through open source knowledge: From open software to open hardware](#). *Media, Culture & Society*, 34(6), 691–708.
- Raasch, C., Lee, V., Spaeth, S., & Herstatt, C. (2013). [The rise and fall of interdisciplinary research: The case of open source innovation](#). *Research Policy*, 42(5), 1138–1151.
- Raymond, E. S. (2001). [The cathedral & the bazaar: Musings on linux and open source by an accidental revolutionary](#). O'Reilly Media.
- Reckwitz, A. (2002). [Toward a theory of social practices: A development in cultural theorizing](#). *European Journal of Social Theory*, 5(2), 243–263.
- Robles, G., Merelo Guervós, J., & Gonzalez-Barahona, J. M. (2005, May). [Self-organized development in libre software: a model based on the stigmergy concept](#). 6th International Workshop on Software Process Simulation and Modeling, Missouri, United States.
- Schatzki, T. R. (2002). *The site of the social: A philosophical account of the constitution of social life and change*. Pennsylvania State University Press.

- Schatzki, T. R., Knorr Cetina, K., & von Savigny, E. (2001). *The practice turn in contemporary theory*. Routledge.
- Schlagwein, D., Conboy, K., Feller, J., Leimeister, J. M., & Morgan, L. (2017). [“Openness” with and without information technology: A framework and a brief history](#). *Journal of Information Technology*, 32(4), 297–305.
- Schultze, U. (2000). [A confessional account of an ethnography about knowledge work](#). *MIS Quarterly: Management Information Systems*, 24(1), 3–40.
- Secker, J. (2014). [Broadening the scope of information literacy in 2015: Beyond libraries](#). *Journal of Information Literacy*, 8(2), 1–2.
- Secker, J. (2018). [The revised CILIP definition of information literacy](#). *Journal of Information Literacy*, 12(1), 156–158.
- Sher Decusatis, C. J., Carranza, A., & Decusatis, C. M. (2012). [Communication within clouds: Open standards and proprietary protocols for data center networking](#). *IEEE Communications Magazine*, 50(9), 26–33.
- Shove, E., Pantzar, M., & Watson, M. (2012). [The dynamics of social practice: Everyday life and how it changes](#). SAGE Publications Inc.
- Špiranec, S., & Kos, D. (2013). [Information literacy practices and student protests: Mapping community information landscapes](#). *Information Research: An International Electronic Journal*, 18(3).
- Stallman, R. (1985). [The GNU manifesto](#). GNU Operating System.
- Stordy, P. H. (2015). [Taxonomy of literacies](#). *Journal of Documentation*, 71(3), 456–476.
- Teixeira, J. (2014, October 27-30). [Developing a cloud computing platform for Big Data: The OpenStack Nova case](#). IEEE International Conference on Big Data, Washington DC, United States.
- Teixeira, J. (2015, August 19-21). [On the openness of digital platforms/ecosystems](#). 11th International Symposium on Open Collaboration, OPENSYM 2015, San Francisco, United States.
- Teixeira, J. (2017). [Release early, release often and release on time. An empirical case study of release management](#). In F. Balaguer, R. Di Cosmo, A. Garrido, F. Kon, G. Robles, & S. Zacchiroli (Eds.), *Open source systems: Towards robust practices* (pp. 167–181). Springer International Publishing.
- Teixeira, J., & Hyrnsalmi, S. (2017, June 12-13). [How do software ecosystems co-evolve? A view from OpenStack and beyond](#). 8th International Conference on Software Business, Essen, Germany.

- Teixeira, J., & Karsten, H. (2019). [Managing to release early, often and on time in the OpenStack software ecosystem](#). *Journal of Internet Services and Applications*, 10(1).
- Teixeira, J., Mian, S., & Hytti, U. (2016, December 11-14). [Cooperation among competitors in the open source arena: The case of OpenStack](#). Proceedings of the International Conference on Information Systems (ICIS 2016), Dublin, Ireland.
- Teixeira, J., Robles, G., & González-Barahona, J. M. (2015). [Lessons learned from applying social network analysis on an industrial Free/Libre/Open Source Software ecosystem](#). *Journal of Internet Services and Applications*, 6(1).
- Tidal, J. (2017). The promise and perils of open source. In C. Antonucci & S. Clapp (Eds.), *The LITA leadership guide : The librarian as entrepreneur, leader, and technologist* (1st ed., pp. 79–91). Rowman & Littlefield Publishers, Incorporated
- Tkacz, N. (2012). [From open source to open government: A critique of open politics](#). *Ephemera: Theory & Politics in Organization*, 12(4), 386–405.
- Tuominen, K., Savolainen, R., & Talja, S. (2005). [Information literacy as a sociotechnical practice](#). *Library Quarterly*, 75(3), 329–345.
- Ulhøi, J. P. (2004). [Open source development: A hybrid in innovation and management theory](#). *Management Decision*, 42(9), 1095–1114.
- Venturini, T., & Latour, B. (2009). *The social fabric: Digital traces and quali-quantitative methods*. In Proceedings of Future En Seine 2009. Future En Seine.
- von Hippel, E., & von Krogh, G. (2003). [Open source software and the “private-collective” innovation model: Issues for organization science](#). *Organization Science*, 14(2), 209–223.
- von Krogh, G., & Spaeth, S. (2007). [The open source software phenomenon: Characteristics that promote research](#). *The Journal of Strategic Information Systems*, 16(3), 236–253.
- von Krogh, G., & von Hippel, E. (2006). [The promise of research on open source software](#). *Management Science*, 52(7), 975–983.
- Webber, S. A., & Johnston, B. (2017). [Information literacy: Conceptions, context and the formation of a discipline](#). *Journal of Information Literacy*, 11(1), 156–183.
- Weiner, S. A. (2011). [Information literacy and the workforce: A review](#). *Education Libraries* 34(2), 7–14.
- West, J., & Kuk, G. (2016). [The complementarity of openness: How MakerBot leveraged Thingiverse in 3D printing](#). *Technological Forecasting and Social Change*, 102, 169–181.
- West, J., Salter, A., Vanhaverbeke, W., & Chesbrough, H. (2014). [Open innovation: The next decade](#). *Research Policy*, 43(5), 805–811.

- Widén, G., Steinerová, J., & Voisey, P. (2014, September 2-5). [Conceptual modelling of workplace information practices: A literature review](#). *Information Research an International Electronic Journal*, 19(4).
- Willinsky, J. (2005). [The unacknowledged convergence of open source, open access, and open science](#). *First Monday*, 10(8).
- Workman, M., Kahnweiler, B., & Bommer, W. (2003). [The effects of cognitive style and media on commitment to telework and virtual teams](#). *Journal of Vocational Behavior*, 63(2), 199–219.
- Zaggl, M. A. (2025). [How artifact-based and authority-based coordination affect propagation costs in open source software development](#). *Management Information Systems Quarterly*, 49(2), 805–822.