Screen Recordings, Screencasts and Video in Software Engineering

An Evidence Map of Bug Reporting, Documentation, Training and Collaboration

Ankur Shrestha

Vertov, United States
ORCID: 0009-0009-6036-5849

14 November 2026

Abstract

Developers report bugs, teach one another, document designs and review work with recorded screens, but the research on these recordings is spread across communities that use different words for the same artifact. This paper maps that research. It is an AI-assisted evidence map. Searching open scholarly sources on 14 November 2026 returned 1466 records, or 1382 after deduplication; 198 studies were included and coded for activity, recording type, role, outcomes and strength of comparison. Tutorial and training studies (116) and requirements studies (39) are the largest groups; bug reporting accounts for 19 and code review for 1. Tool accuracy (69 studies) and satisfaction (63) are the most common outcomes; communication time was measured in 20 studies. Of the 27 studies with text, or text and screenshots, as the comparator, 19 report a direction and the results are mixed: 8 favor the recording, 8 are mixed, 1 found no difference and 2 favor the alternative. Only 6 put a recording in place of text for people, 4 of them at the edge of software engineering. The paper proposes four families of recording roles, an evidence-gap map and an open dataset that links key findings, data bases and some effect directions to quoted passages.

Keywords: screen recording; screencast; video; software engineering; evidence map; bug reporting; documentation; requirements engineering

1 Introduction

A screen recording captures what a person sees and does, usually with a voice that explains it. Software teams use such recordings to show a defect to a colleague, to teach a tool, to walk through a design and to leave a record of how something works. A developer who cannot describe a layout glitch in words can record it; a maintainer who wants to teach a new contributor can record the procedure once. Still images already carry communication on social coding platforms [1]. The practice is old: video has been proposed as a requirements medium since at least 2006 [2].

The research on these recordings does not share a vocabulary. Studies of bug reports speak of video-based bug reports and visual issue reports [3, 4]. Studies of programming knowledge speak of screencasts and developer-created videos [5, 6]. Requirements engineering speaks of vision videos [7]. Instructional design speaks of demonstration-based video tutorials [8]. A reader who wants to know what is known about recordings in software work has to search all of these, and a practitioner who is deciding whether to ask contributors for a recording rather than a written description cannot easily see which claims rest on controlled comparisons and which rest on opinion.

This paper maps that literature. It asks five questions.

- **RQ1 Usage.** In which software engineering activities have screen recordings, screencasts, video demonstrations or related visual artifacts been studied?
- **RQ2 Representation.** How are recordings captured, structured, annotated, edited, indexed, retrieved and consumed, and by whom?
- **RQ3 Outcomes.** What outcomes have researchers measured?
- **RQ4 Evidence.** Where comparative studies exist, what do they establish about recordings versus text, screenshots and mixed communication?
- **RQ5 Gaps.** Which activities, populations, formats and outcomes are insufficiently studied in the studies identified?

The questions do not presume that a recording is better than text. Theory predicts when a richer medium should help, and studies of other fields suggest that adding video to teaching helps on average [9], but the same work also shows that effects depend on how video is used [10]. Whether that carries over to software work is an empirical question, and the map is built to show contradictory and null results with the same weight as favorable ones.

Contribution. To our knowledge, and within the open sources searched up to 14 November 2026, no prior secondary study maps this literature across bug reporting, documentation and training, and collaboration. The nearest secondary work is narrower: systematic reviews of bug-report analysis [11] and of bug reproduction and localization [12], the requirements-engineering video literature that a quality model for vision videos builds on, which its authors describe as not systematic [7], and reviews of video in education [9, 13]. Three near misses are secondary work in the requirements strand alone: a master's thesis whose literature search covers methods for assessing shared understanding and adapts them to vision videos in requirements engineering [14], a doctoral dissertation on vision videos in requirements engineering that develops and validates an approach and is not a review [15], and a mapping of educational software engineering content on YouTube, which maps the videos and not the research on them [16]. None maps recordings across bug reporting, documentation and training, and collaboration. The contributions are: (1) a corpus of 198 included studies drawn from 1382 unique records; (2) a proposed grouping of the coded roles into four families; (3) cross-tabulations of activity, artifact, outcome, method and evidence strength, including an evidence-gap map; (4) a table and discussion of the 19 studies that report a direction against a text comparator; and (5) an open dataset in which key findings, data bases and some effect directions carry quoted passages. The claim above is "none found", not "none exists": the reasons that it could be wrong are listed in Section 6.

Disclosure. The author is affiliated with a company that develops screen-recording software. The eligibility rules and the search strings were committed to the project repository before coding began (they were not registered), vendor material was excluded, and contradictory and null results are reported in full. The competing-interest statement at the end of the paper gives the formal wording.

2 Background

Why a recording might help. Communication needs mutual understanding, and a medium changes what it costs to reach it [17]. A shared view of the work lets a helper see the object, the person's actions and the state of the task without asking, and it changes the language people use [18, 19]. Media synchronicity theory adds that media differ in how well they support conveying information and converging on a shared meaning, and that the fit depends on the task [20]. A recording is a rich but one-way and asynchronous medium: it shows the screen and carries speech, but the viewer cannot ask questions. For instruction, the multimedia-learning literature adds that

narration and animation can reduce load when they are designed well and can add load when they are not [21]. These theories predict when a recording should help; they do not show that it does, and they are used here only to set up the questions.

Prior secondary studies. The secondary literature that touches this topic is narrow. A systematic review covers machine-learning methods for bug reports as text [11] and a mapping study covers bug reproduction and localization [12], which could not be read because its text was blocked; a vision paper covers bug tracking in the generative-AI era [22]; mapping or review studies cover usability-defect reporting [23], documentation in continuous development [24], onboarding solutions [25] and code review [26]. The others were read at the level the open sources allowed, and the recording is not their subject. The requirements-engineering video literature is the oldest strand, and a quality model for vision videos rests on two reviews that its authors do not claim to be systematic [7]. Education has several reviews of video and screencast feedback, whose subjects are students and not software work [9, 10, 13]. A mapping across activities therefore has a different scope from every secondary study found.

Evidence maps. An evidence map describes the extent and nature of research on a topic and shows where evidence exists and where it does not, without synthesizing effects [27]. It is related to scoping reviews [28, 29] and to systematic mapping studies in software engineering [30], which follow the wider guidelines for systematic reviews in the field [31]. This paper follows the evidence-map logic: it classifies studies on a small number of dimensions and reports counts and gaps, and where it summarizes results it does so narratively, in the manner of synthesis without meta-analysis [32].

3 Methods

Design and label. The study is an AI-assisted evidence map. It is not a systematic review, not a meta-analysis and not a census; it was not preregistered, and it reports no pooled effect, no p-value and no confidence interval. Reporting is guided by PRISMA-ScR [29] where its items fit. The eligibility rules, the research questions, the codebook and the search strings were committed to the project repository before coding began, with no external timestamp, and are included in the dataset. Searching, screening, extraction and analysis were carried out by AI agents that wrote only to their own folders, followed by a merge step that deduplicates records, validates every value against a controlled vocabulary and writes every disagreement to a log. No person screened records or independently recoded values.

3.1 Search

Searching ran on 14 November 2026 across open interfaces: OpenAlex, Crossref, arXiv, Semantic Scholar, Zenodo and web search, with citation snowballing through OpenCitations, OpenAlex and Semantic Scholar. The core string combined recording terms (screencast, screen recording, video bug report, vision video, requirements video, demo video, walkthrough video, programming video, code screencast) with software engineering contexts (software engineering, software development, requirements engineering, bug report, issue report, defect report, program comprehension, software documentation, code review). Four topic agents also ran their own variants for visual bug reporting; documentation, tutorials and training; collaboration and requirements; and theory and prior reviews. The agents logged 545 queries, each with its database, date, string and hit count. Backward snowballing scanned the reference lists of saved texts, an approach that guidelines for snowballing in software engineering describe [33] and that is often combined with database search [34]; forward snowballing was done for seed and core papers by the bug-reporting and collaboration

strands, and not for the documentation and training strand, where the OpenAlex limit was reached first. The ACM Digital Library, IEEE Xplore, Scopus and Web of Science were not searched, DBLP returned a bot challenge and was not queried, and the OpenAlex and Semantic Scholar rate limits cut some queries short; these gaps are logged and are discussed in Section 6.

3.2 Screening and eligibility

A study was eligible when it was peer-reviewed or archived research (a preprint, thesis or technical report with retrievable text; guidelines for multivocal reviews discuss when such grey literature belongs in a review [35]) from 2000 to 2026, in which a recorded screen or video is a communication artifact in a software engineering context: reported, produced, consumed or evaluated by developers, testers, designers, users reporting to developers, students learning software tasks, or tools and agents. Tool and technique papers were included when the recording is the communicated artifact. Studies were excluded for: X1 not software engineering or software-task related; X2 video is not a communication artifact; X3 no retrievable text or abstract; X4 duplicate or superseded version of an included record; X5 opinion or vendor material with no research content; X6 outside the date range; X7 a study of a different artifact (screenshots only, text only) with no video component.

The four topic agents kept 1466 records as candidates after their own keyword prefiltering of much larger result sets (the size of the pool before prefiltering was logged for one strand only; a record found in several sources counts in each in Figure 1). 84 duplicates were removed and 1382 unique records were screened. The largest exclusion, X1 (1011 records), comes mostly from broad queries; 998 records were excluded on a keyword and title scan without reading an abstract. 17 records had no retrievable abstract or text, and 51 are marked pending: most were shortlisted on title and keywords, and either no open text was secured or extraction was not finished in the sprint. These 68 relevant-looking records were not coded and are listed in the dataset; by title they include studies that compare video with paper or text tutorials, so the counts below are likely to understate that group. Where two agents disagreed on a record, the decision made at the higher reading level was kept provisionally, and ties were settled by the coordinating agent and logged. Records of one study in several versions were not always merged: 5 pairs of conference and journal versions remain in the corpus as separate records (10 records), and a further group of 3 records may overlap. The counts are counts of records, and merging the confirmed pairs would lower 198 by 5; the pairs are listed in the dataset.

Two groups sit at the edge of the eligibility rule and are kept and flagged in the dataset: 6 studies set in education outside software engineering, and 8 in which recordings are training data or input for an agent, which the rule allows because a tool is a listed consumer. Together they are 14 studies; 11 are in the tutorial group and 8 have a tool or agent as the consumer.

3.3 Extraction and coding

Appendix B lists every included study with the code used in this paper (Table 16). Every included study was coded on the fields listed in the dataset codebook: publication type and peer-review status of the source; software engineering activity; artifact type; communication role; producer and consumer of the recording; representation (how it is captured, structured, annotated, indexed or retrieved); research method; the data basis and its reported size; outcomes; comparator; effect direction (only for studies with a comparator); evidence level; key findings; limitations; and data availability. The evidence level has four values: 1 no comparison, 2 qualitative or descriptive, 3 observational or non-controlled comparison, 4 controlled comparison. Controlled vocabularies

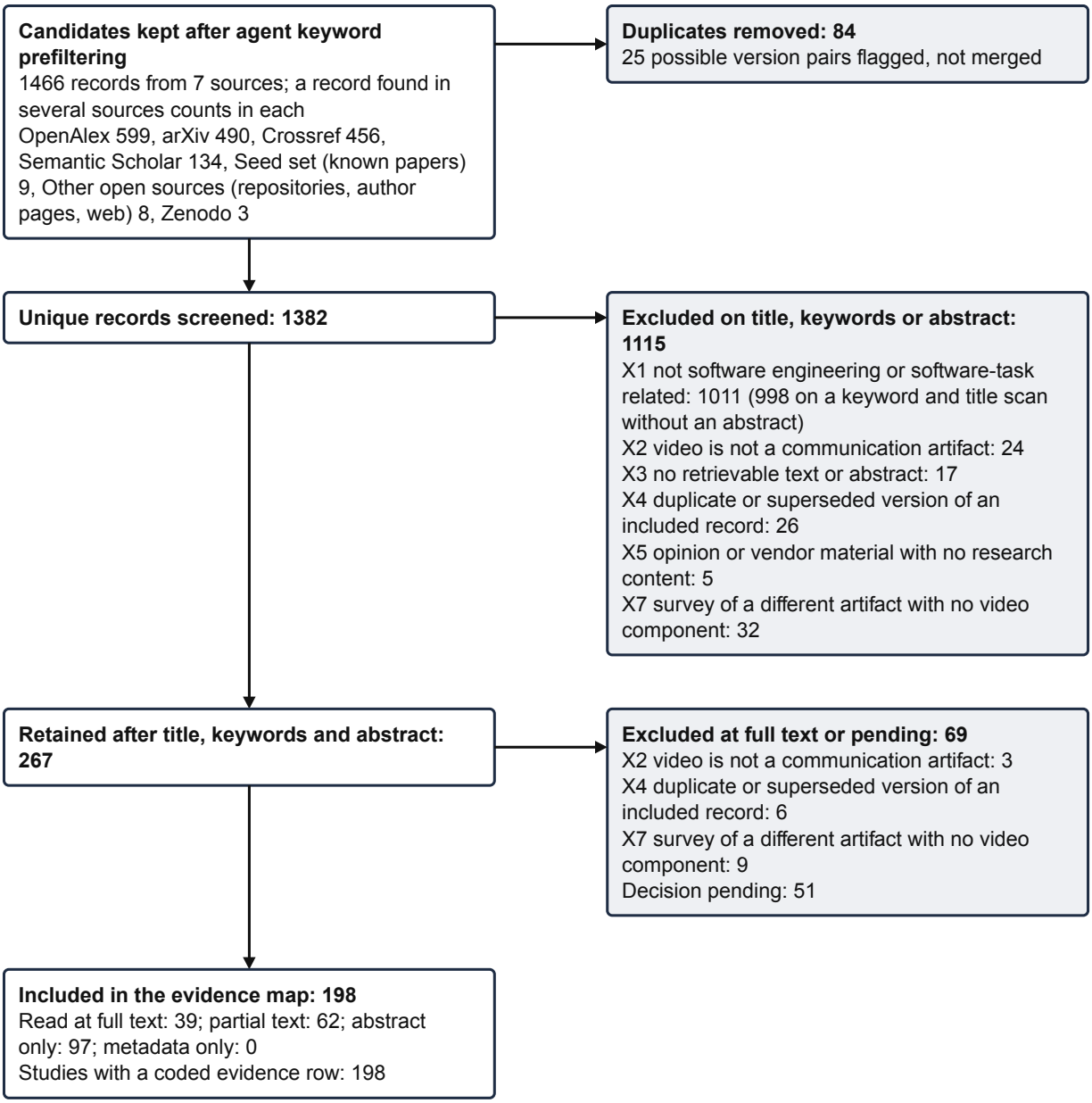


Figure 1. Study selection flow, from records identified to studies included.

were fixed in advance, and one value proposed during coding was kept and flagged.

Each coded value carries a status (reported in the source, calculated from reported numbers, inferred, not reported, not applicable). Verbatim quotes of at most 40 words, with their location, were collected for key findings (165 studies), the data basis (124), method (19) and effect direction (6), and for some limitations and data-availability statements. The categorical codes (activity, artifact, role, producer, consumer, outcomes, comparator, evidence level) were assigned by the agent from the text and carry no quote. Every record carries the reading level at which it was coded: metadata only, abstract, partial text (the full text was saved, and extraction used the abstract and keyword-located passages) or full text. A script then checks that each quote occurs, in fragments of at least six words, in the saved source text after folding case, punctuation and

hyphenation: 454 quotes passed, none failed, and 5 were too short to test. The check shows that the words exist in the source, not that they support the code. The identity of an included record (title, authors, year, DOI) was checked against Crossref, DataCite, arXiv or the publisher page for 99 studies, partly for 9 and not successfully for 10; for 80 no check was recorded in the evidence table. References were taken from source records and not typed from memory. Where agents with overlapping topics coded the same study differently, the merge step logged the disagreement (102 logged rows, of which 32 needed a coordinator decision) and the values from the row read at the higher level were kept; the remaining ties were settled by the coordinating agent keeping one agent's row, and those decisions are in the correction log. That rule is a convenience and not a measure of accuracy. The conflicts logged by field are a rough indicator of reliability (activity 12, role 10, outcomes 10, consumer 8, evidence level 7). Studies of large language models in screening and extraction report accuracy that depends on the task and recommend human oversight [36], and no audit sample was drawn here.

3.4 Analysis

The analysis is descriptive. A study with several values (for example two activities) counts once in each, so row and column totals can exceed the number of studies. Cross-tabulations count distinct studies. The evidence-gap matrices report, for each cell, the number of studies and the number with a non-controlled (level 3) or controlled (level 4) comparison; a cell with no study means that none was identified in this map and is not evidence that none exists. The comparative sub-corpus is the set of studies whose comparator is not "none". The text-comparison table contains the studies whose comparator is text, or text and screenshots, and whose coded direction is not "not applicable"; each is also coded for its contrast (a recording in place of text, a video added to text, an observational comparison of issue reports, a stated preference or rating, or agent participants), a coding made by the coordinating agent after the independent review. Effect directions are counts of the direction each study reported; they are not effect sizes and are not pooled, because the studies differ in design, population and outcome, and no study was weighted by quality. Every number in the abstract, tables and results text is produced by a script from one merged table, and a consistency check fails the build if a number typed in the abstract differs from the data.

4 Results

4.1 The corpus

198 studies were included (Figure 1). 154 are in sources coded as peer reviewed, 27 are preprints and 6 are theses. 149 were published in 2018 or later, and the publication count has grown from 7 studies in 2006 to 2008 to 54 in 2024 to 2026 (Figure 2). The coded evidence rests on different amounts of reading: 39 studies were read as full text, 62 in part and 97 at the abstract level only. Statements below about a study's result therefore carry the level at which it was read, and the tables give it.

The studies are mostly empirical or technical. 73 evaluate a tool, 51 report an experiment, 20 use mixed methods, 13 are case studies, 9 mine repositories, and the rest are surveys, interviews, observational studies, quasi-experiments and 4 literature reviews (Table 1).

Table 7 in Appendix A pairs the methods with the activities.

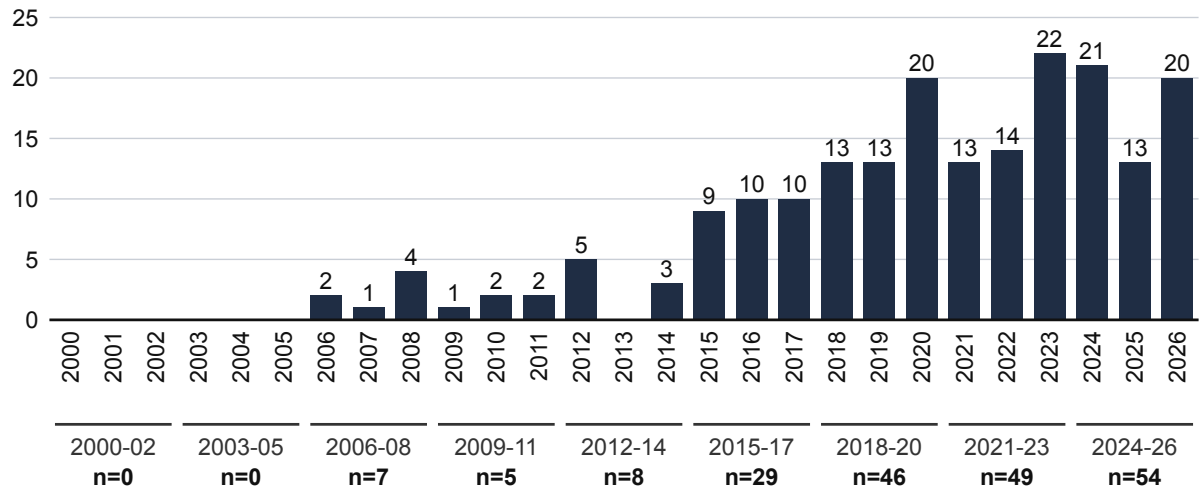


Figure 2. Included studies by publication year, with three-year band totals.

Table 1. Included studies by research method.

Method	Studies	Share of included studies (%)
Experiment	51	25.8
Quasi experiment	11	5.6
Survey	7	3.5
Interview	3	1.5
Case study	13	6.6
Mining	9	4.5
Tool evaluation	73	36.9
Literature review	4	2.0
Observational	7	3.5
Mixed	20	10.1
Other*	11	5.6

A study with several values counts once in each; 198 studies in total. An asterisk marks a value proposed during coding, not in the initial codebook. Other covers position papers, design cases, experience and idea papers and records with no stated method.

4.2 Where recordings have been studied (RQ1)

Table 2 shows the activities. Tutorial learning has the most studies (116), followed by documentation (46), requirements (39), onboarding and training (30), knowledge transfer (22) and bug reporting (19). Bug reproduction (10), program comprehension (10), testing (17) and remote collaboration (13) follow; UI design feedback (4) and code review (1) are the thinnest. 128 studies belong to more than one activity, which is why the counts add to more than 198.

The recorded artifacts divide along the same lines (Table 3). Tutorial videos (79 studies) and screencasts (60) dominate; vision videos (25) are almost all in requirements engineering; screen-recording bug reports (23) are almost all in bug reporting, reproduction and testing; live-stream recordings (11) and asynchronous video messages (3) are small groups. Figure 3 shows the pairings, and a dash in the table means no study was identified for that pair.

Table 2. Included studies by software engineering activity.

Activity	Studies	Share of included studies (%)
Bug reporting	19	9.6
Bug reproduction	10	5.1
UI design feedback	4	2.0
Code review	1	0.5
Documentation	46	23.2
Tutorial learning	116	58.6
Onboarding training	30	15.2
Requirements	39	19.7
Remote collaboration	13	6.6
Knowledge transfer	22	11.1
Program comprehension	10	5.1
Testing	17	8.6
Other	5	2.5

A study with several values counts once in each; 198 studies in total.

Table 3. Included studies by type of recorded artifact.

Artifact type	Studies	Share of included studies (%)
Screencast	60	30.3
Screen recording bug report	23	11.6
Vision video	25	12.6
Demo video	8	4.0
Tutorial video	79	39.9
Live stream recording	11	5.6
Async video message	3	1.5
Session recording	10	5.1
Other	23	11.6

A study with several values counts once in each; 198 studies in total.

4.3 The roles a recording plays (taxonomy)

The coded communication roles fall into four families, shown in Figure 4. A recording can be **evidence of a state** (a defect, a test run, a result): this family has 33 studies, mostly reporting and verification. It can be **instruction and demonstration** (138 studies): it shows how, or why, to someone who will act on it later, and it is the largest family. It can be a **record of knowledge** (57 studies): a procedure, a design or a session kept to be found again. And it can be a **shared medium** (33 studies): a common object that people discuss, review or react to, as with a vision video that stakeholders comment on. The four are not exclusive, and a study counts in every family whose role it carries. The families are a grouping of the nine coded role values, made by the coordinating agent after coding and not validated, and one family holds 138 of the 198 studies; they are a proposal for organizing the literature, not a tested taxonomy.

A second dimension is who consumes the recording. 76 studies have students as consumers and 64 developers; 41 studies, about one in five, have a tool or an agent as the consumer, as when a tool replays a bug from a video or an agent learns a task from tutorial videos. The evidence family has the largest share of tool consumption (16 of 33 studies). Table 14 in Appendix A gives the counts of producers and consumers. Producers are developers in 78 studies, end users in 47 and researchers in 51. Table 6 in Appendix A pairs the roles with the activities.

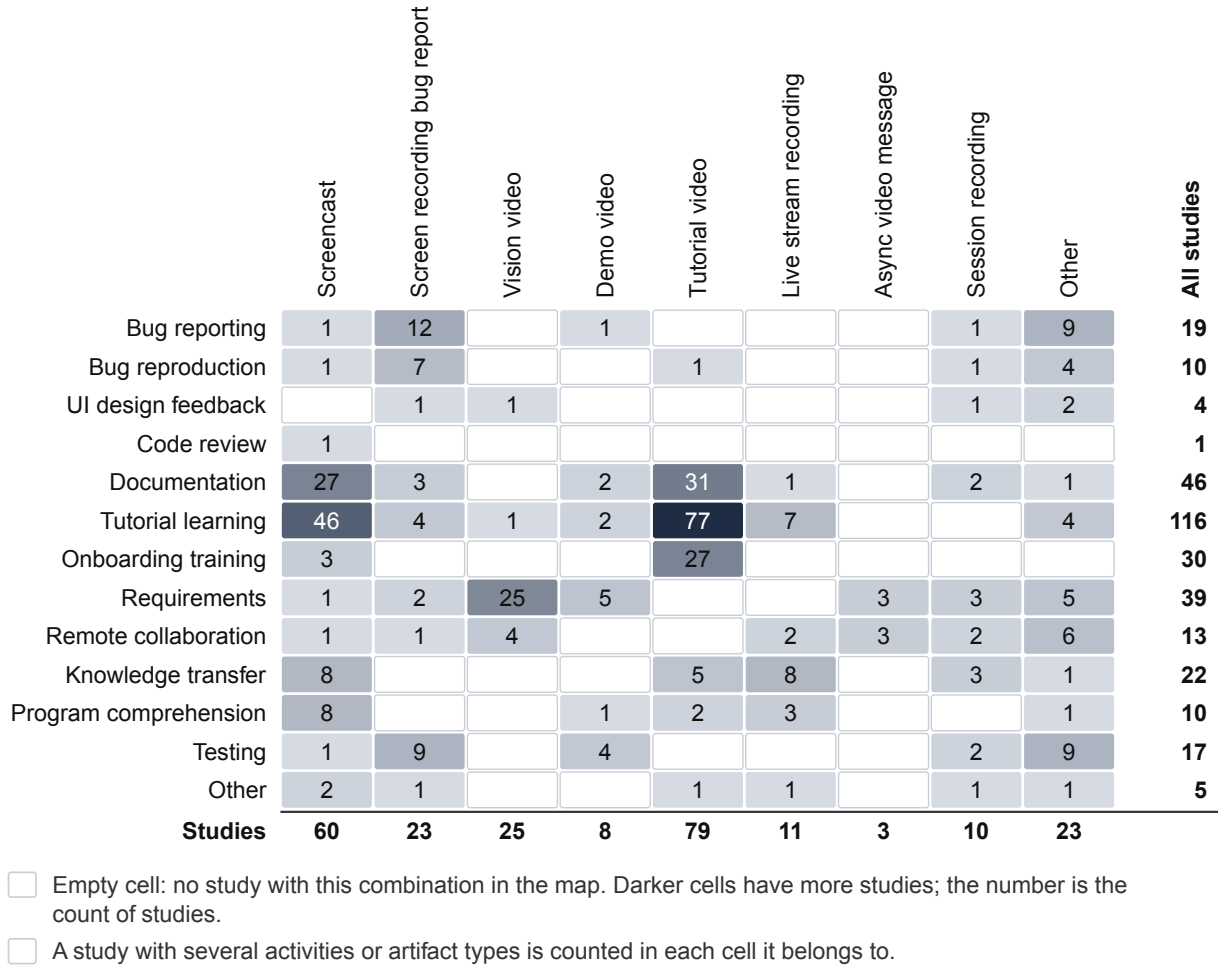


Figure 3. Included studies by software engineering activity and artifact type (number of studies per cell).

4.4 Representation and use (RQ2)

Studies describe representation in the free-text field of the codebook, and most of what they describe concerns extraction and indexing: detecting duplicate video-based bug reports from visual and textual features [4, 37], translating a recording into replayable steps [38], generating captions for the steps in a recording [39], and extracting code, text and actions from programming screencasts with optical character recognition and action recognition [40]. A smaller group studies how people produce recordings: what makes a vision video good [41], how developers document knowledge with YouTube screencasts [6, 5], and how streamers share their work [42]. The studies that report on the format of what is consumed (length, narration, segmentation, cueing, interactivity) are mostly instructional-design experiments, and their within-video comparisons tend to find small or null effects, such as cueing and segmentation [43, 44, 45]. The coded data support a count of these studies but not a comparison of representations across the whole map, because the representation field is free text and partly recorded at the abstract level.

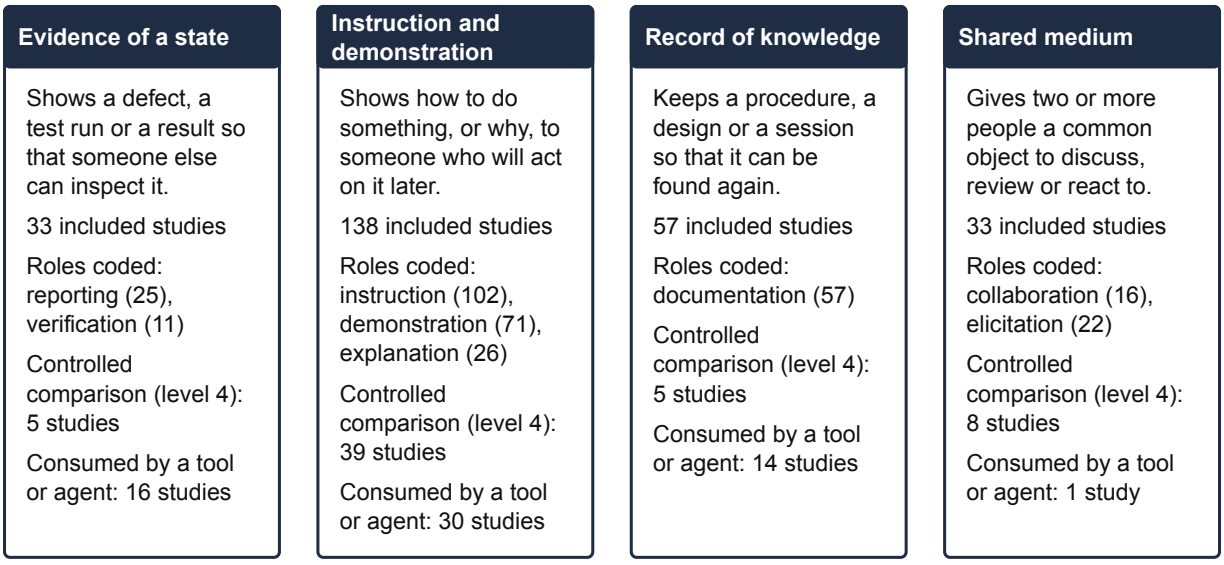


Figure 4. Roles of a recording in the included studies. A study counts in every role it carries, so the four counts overlap.

4.5 What has been measured (RQ3)

Table 4 lists the outcomes. The most common is tool accuracy (69 studies), reflecting the weight of tool papers, followed by satisfaction (63) and learning performance (45). Communication time (20), adoption or use (20), comprehension (18) and task completion (16) are measured in a similar number of studies, and cognitive load in 12. Reproduction accuracy (5 studies), clarification effort (4) and error rate (1), the outcomes that matter most to the person who receives a report, are rarely measured. Table 8 in Appendix A shows the pairing with activities, and Table 15 shows the size of the data bases the studies report. Bug reporting is measured mostly through tool accuracy and communication time; tutorial learning and onboarding through learning performance and satisfaction; requirements through comprehension, satisfaction and a large group of outcomes coded as other.

Table 4. Outcomes measured in the included studies.

Outcome	Studies	Share of included studies (%)
Comprehension	18	9.1
Task completion	16	8.1
Reproduction accuracy	5	2.5
Communication time	20	10.1
Clarification effort	4	2.0
Error rate	1	0.5
Learning performance	45	22.7
Cognitive load	12	6.1
Satisfaction	63	31.8
Tool accuracy	69	34.8
Adoption use	20	10.1
Other	55	27.8

A study with several values counts once in each; 198 studies in total.

4.6 What the comparative studies show (RQ4)

102 of the 198 studies have a comparator (Table 10). Most of these compare a tool with a baseline tool (49 studies, none with a direction because the baseline is not a medium), a recording with a human baseline such as classroom teaching (16), or one form of video with another (10). 27 studies have text, or text and screenshots, as the comparator. 8 of them state no direction (tool papers, experience reports and descriptive studies) and are not tabulated; 19 report a direction (Table 5). Of those, 9 are controlled comparisons (level 4) and 7 are observational (level 3).

The 19 studies are not all contrasts of a recording with text. 4 test a video added to a text specification, 6 put a recording in place of text for people, 3 observe issue reports with and without visual content, 5 measure a stated preference or rating, and 1 has agents as the participants. The direction of the reported results is mixed: 8 favor the recording, 8 are mixed, 1 found no difference and 2 favor the alternative. Among the controlled comparisons, 5 favor the recording, 3 are mixed and 1 found no difference. Of the 6 studies that put a recording in place of text, 2 favor it and 2 of those are set at the edge of software engineering; 4 of the 6 are. Table 13 in Appendix A gives the directions by evidence level for all comparative studies. 12 of the 19 studies were read as full text or in part and 7 from the abstract only; the abstract-only directions are the authors' summary and not a checked result. A quoted passage supports the direction for 6 of the 19.

Table 5. Studies with text, or text and screenshots, as the comparator and a reported direction.

Study	Activity	Contrast	Direction	n	Level	Read	Finding
Liu et al. 2025 [46]	Tutorial learning	Agent participants	Favors video	361	4	Partial text	On OSWorld the video-demonstration condition scored 50.3 versus 46.8 for the baseline agent and 44.0 with a text tutorial on the 211-task subset; adding transcripts alone did not help.
Shi et al. 2024 [47]	Requirements, Testing	Added to text	Favors video	22	4	Partial text	Participants who saw test-execution videos together with Gherkin specifications identified more requirement inconsistencies than those with Gherkin alone (significant difference on the primary metric, p=0.024).
Shi et al. 2023 [48]	Requirements, Testing	Added to text	Favors video	22	4	Abstract	With 22 participants, GUI test videos with Gherkin specifications tended to lead to more identified inconsistencies than Gherkin specifications alone.

Study	Activity	Contrast	Direction	n	Level	Read	Finding
Karras et al. 2018 [49]	Requirements	Added to text	Favors video	16	4	Full text	Specification plus vision video cut time to extract information by 24.67% (775.8 s to 584.4 s) and raised correct answers by 20.59% (6.8 to 8.5) versus specification alone.
Lloyd and Robertson 2011 [50]	Tutorial learning	Recording in place of text (edge setting)	Favors video	n/r	4	Abstract	Screencast tutorials were reported as effective and efficient, especially for higher-order conceptual statistical knowledge, compared with a text tutorial.
Schneider et al. 2019 [51]	Requirements	Added to text	Mixed	20	4	Partial text	Participants who first read text and then watched video (or reverse) contributed significantly more new requirements-related ideas; no overall preference between camera perspectives.
Käfer et al. 2017 [52]	Tutorial learning, Knowledge transfer	Recording in place of text	Mixed	n/r	4	Full text	Text-only learners finished the tutorial faster; video-only learners applied the content faster; final performance was about equal; learners prefer video for learning and text for look-up.
Meij and Meij 2014 [53]	Tutorial learning, Onboarding training	Recording in place of text (edge setting)	Mixed	n/r	4	Abstract	Both tutorial types raised task relevance, self-efficacy, mood and flow with about 30 percent learning gain; a task showed a significant interaction with tutorial type favoring the video.
Shi et al. 2023 [54]	Testing, Bug reporting, Program comprehension	Recording in place of text	No difference	11	4	Partial text	Videos and screenshots differed visibly in effectiveness and efficiency of defect analysis but the differences were not statistically significant (Mann-Whitney U, p between .22 and .79).

Study	Activity	Contrast	Direction	n	Level	Read	Finding
Worlitz et al. 2018 [55]	Tutorial learning, Onboarding training	Recording in place of text (edge setting)	Favors video	56	3	Full text	73 percent of students used the video tutorial; in the second test video tutorials shortened task duration, while suitability ratings and result quality did not differ between aids.
Kuramoto et al. 2024 [3]	Bug reporting	Observational repository	Mixed	481603	3	Partial text	Visual reports were more often bug reports (44.6% images, 71.1% animations/videos) and had shorter descriptions.
Paikari 2023 [56]	Bug reporting	Observational repository	Mixed	2814599	3	Partial text	Including videos did not reliably help: resolution time was barely affected, a lower share of reports with videos was resolved with a patch, and back-and-forth was higher.
Kuramoto et al. 2022 [57]	Requirements, Remote collaboration	Observational repository	Mixed	1230	3	Full text	Issue reports with images are described in fewer words than text-only reports; videos had a median description length of 54.5 words versus 60.0 without media; discussions concentrate on reproduction conditions and GUI.
Nørgaard and Hornbæk 2009 [58]	UI design feedback	Preference or rating	Mixed	3	3	Abstract	Initially, developers rated redesign proposals, multimedia reports and annotated screen dumps above problem lists and scenarios; after working with the feedback, ratings did not differ significantly.
Høegh 2008 [59]	UI design feedback	Preference or rating	Mixed	n/r	3	Abstract	Written usability reports alone were criticized for overload and missing information; oral and multimedia feedback helped developers understand problems better but were less cost-effective.

Study	Activity	Contrast	Direction	n	Level	Read	Finding
Mestre 2012 [60]	Tutorial learning	Recording in place of text (edge setting)	Favors alternative	21	3	Abstract	Across learning preferences, students recreated tasks much better after the static screenshot tutorial than after the screencast tutorial.
Tayeb et al. 2024 [61]	Tutorial learning, Onboarding training	Preference or rating	Favors video	n/r	2	Full text	Video tutorials remained highly preferred for comprehensive coverage; many respondents viewed AI chatbots as potential replacements for written tutorials and as supplements to video tutorials.
Ghilay 2015 [62]	Tutorial learning, Onboarding training	Preference or rating (edge setting)	Favors video	30	2	Abstract	Learners perceived the learning process as excellent and video clips as better than texts; value depends on high-quality, curriculum-wide coverage.
Obaidi et al. 2025 [63]	Tutorial learning, Documentation	Preference or rating	Favors alternative	58	2	Full text	Users preferred moderately detailed explanations in short text formats; app-specific knowledge was not related to the preferred level of detail.

Contrast = what the study compares; edge setting = a setting at the edge of software engineering (Section 3.2). n is the sample size as parsed: participants, or reports or tasks for repository and agent studies; n/r = not reported or not parsed. Level 3 = observational or non-controlled comparison, level 4 = controlled comparison. Read = how much of the study was read. Findings are the first sentence of the extracted key findings (AI first-pass extraction, not human-verified); a direction is the authors' summary where the study was read from the abstract.

Requirements and testing. The strongest favorable results are in requirements engineering. In a controlled experiment with 16 participants, a specification supplemented by a vision video cut the time to extract information by 24.67% and raised the number of correct answers by 20.59% against the specification alone [49]. Experiments with test-execution videos found that participants who saw the video together with Gherkin scenarios tended to identify more inconsistencies than those who saw Gherkin alone, in one case read in part and in the other from the abstract only [47, 48]. These designs add a video to a specification, so they test the supplement and not a replacement, and a second medium added in either order also raised the ideas contributed in one study [51]. The research group around Kurt Schneider at Leibniz University Hannover co-authored 21 of the 198 studies, 20 of the 39 requirements studies and 9 of the 12 controlled requirements studies, and it produced 3 of the 8 favorable results in the text-comparison table; Oliver Karras is a co-author of 17. The controlled evidence in requirements engineering is largely the work of one group.

Learning and training. Results depend on the task and the outcome. In one controlled comparison of a text and a video tutorial, text-only learners finished the tutorial faster, video-only learners applied the content faster and final performance was about equal; learners preferred video for learning and text for look-up [52]. Other work found the same gain from both tutorial types apart from one retention task that favored video [53], reported screencast tutorials as effective for statistical software [50], found no difference between video tutorials and teacher guidance [64], and a result in the opposite direction: students recreated tasks better after a static screenshot tutorial than after a screencast [60]. A survey found that users preferred moderately detailed explanations in short text formats [63], while other studies in the table measure only preferences or perceptions [61, 62]. Of 16 studies with a human baseline, 4 favor the video, 3 found no difference and 9 state no direction.

Bug reporting. Studies that look at what happens when a bug report contains a video do not agree. A mining study of 34 large repositories reports that issues with images closed slightly faster and issues with animation or video took longer to close, with small effects [3]. A dissertation covering about 2.8 million bug reports reports that including a video did not reliably speed resolution, that a lower share of reports with videos was resolved with a patch, that back-and-forth was higher, and that effects varied by project [56]. Another study found no significant difference in response or resolution time between visual and non-visual reports [57]. In a small experiment, videos and screenshots from GUI tests differed visibly in effectiveness and efficiency of defect analysis, but not significantly [54]. These designs are observational or very small; in this map none is a controlled comparison of a recording against a written report for the same bug, and none controls for report quality. The tool papers in this activity report that visual and textual features together find duplicate video reports better than either alone [4], and that generated replay traces or captions shorten reproduction in lab tasks [65, 39], but these compare tools with baselines and do not address whether a report with a video is easier to act on than one without.

Collaboration. Studies of asynchronous discussion of a vision video found higher measured shared understanding than watching the video without discussion [66, 67], and a pilot found that asynchronous viewers raised more arguments while synchronous sessions resolved ambiguities faster [68]. Recorded explanations in code review, asynchronous developer video messages and recorded demos in agile ceremonies have almost no empirical study in this map.

What the direction counts do and do not show. The counts of favorable, mixed, null and unfavorable results are counts of studies. They are not effect sizes, they treat a study with 8 participants and a study of 2.8 million reports alike, and they come from different tasks, populations and outcomes. They also depend on one coding step, made by an AI agent, from sources that were in many cases read in part. They are best read as a picture of where results disagree: within the same activity (bug reporting, learning) the direction differs from study to study, which is itself the main result for RQ4.

4.7 Gaps (RQ5)

Table 9 and the gap matrices (Table 11, Table 12) combine activity with evidence strength (Figure 5 draws the outcome matrix). Four patterns stand out for this set of studies.

First, the activities closest to everyday practice have the least comparative evidence. Bug reporting has 19 studies, of which 3 are controlled comparisons; bug reproduction has 10 and 2; knowledge transfer has 22 and 1; code review has 1 study in total. By contrast, onboarding and training has 19 controlled comparisons among its 30 studies.

Second, the outcomes that bear on the receiver's effort are rare. Clarification effort (4 studies) and reproduction accuracy (5) appear mostly in bug reporting, bug reproduction and testing, and

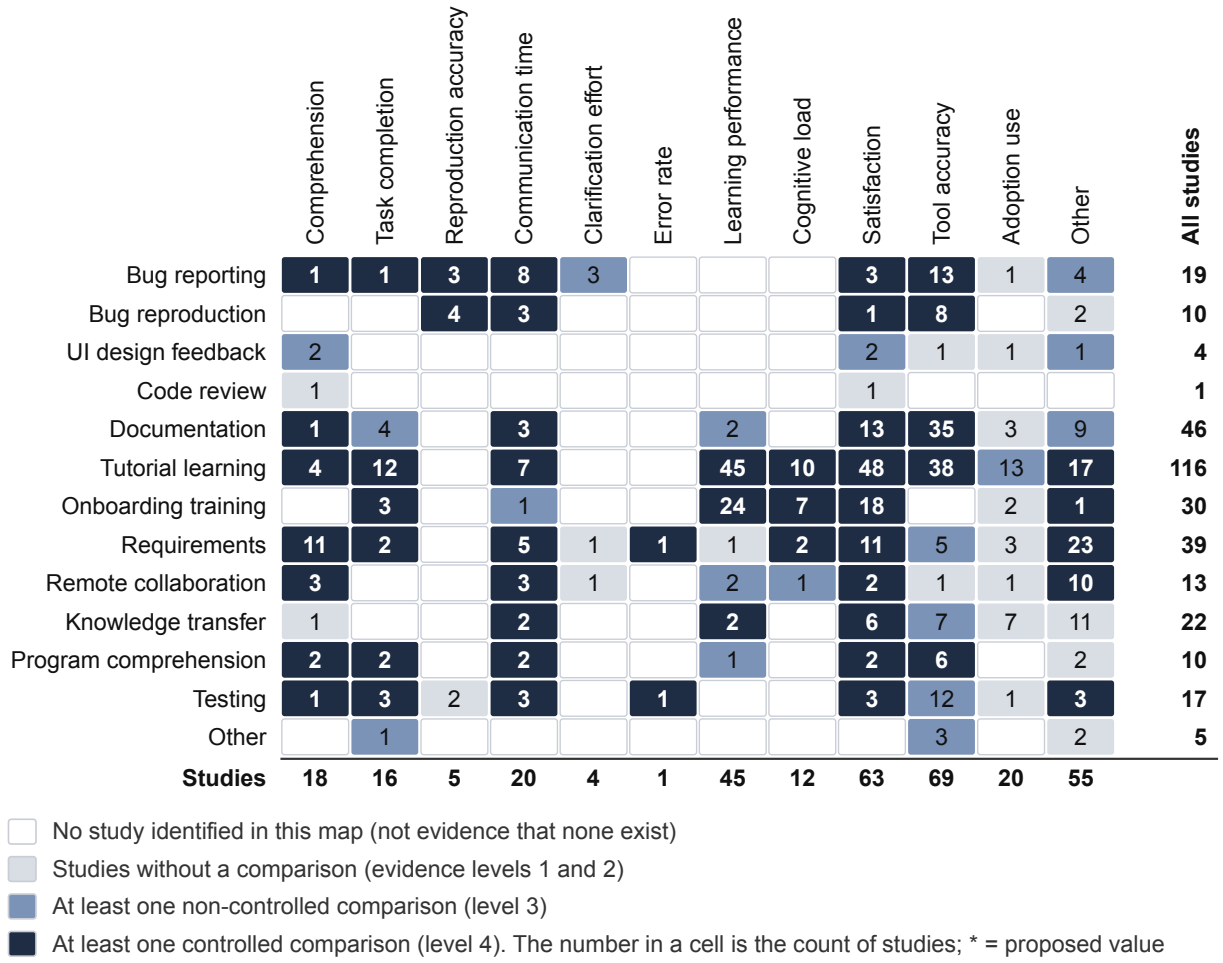


Figure 5. Evidence-gap map: software engineering activity by outcome, shaded by the strongest evidence level in each cell.

error rate has 1 study.

Third, some artifact types are almost unstudied outside one activity. Asynchronous video messages have 3 studies, all in requirements or remote collaboration.

Fourth, the consumers coded are narrow. No included study codes a tester or a designer as the consumer of a recording. The codebook had no value for reviewers, maintainers or support staff, and some consumers may have been coded as developers or end users, so this reflects the coding as much as the literature. 41 studies treat the recording as an input to a tool or agent (8 of them as training data), and whether that output helps a person is a separate question that few of them address.

These statements describe the studies identified in this map. They are not evidence that no study exists, and the reasons recall may be incomplete are in Section 6.

5 Discussion

The same artifact is studied in several separate literatures. The studies identified in this map use at least five names for what is, mechanically, a recorded screen with a voice, and

the strands measure different things: tool accuracy in one, comprehension in another, learning performance in a third. The gaps between the strands are partly gaps of vocabulary, and a search limited to one term would miss most of the others. A shared vocabulary would help the next map.

Evidence that recordings help people is thin and mixed. The largest group of outcomes is tool accuracy, which says whether a technique works and not whether a person gains from it. Among the 19 studies that report a direction against a text comparator, 8 are favorable, and most of those add a video to a text specification or come from settings at the edge of software engineering; one research line produced much of the controlled evidence. In the bug-reporting studies with the largest samples, the findings point in different directions, and observational designs of this kind cannot separate the effect of a video from the effect of the person who chose to attach one. A reader who wants a rule of thumb will not find one here. What the map supports is narrower: the outcome depends on the task and on what is measured, as with the tutorial result in which text was faster to complete and video was faster to apply [52].

What the map implies for research. Studies would be more useful if they (1) measured the receiver's effort, such as clarification rounds, time to reproduce and rework, and not only tool accuracy or satisfaction; (2) compared a recording with a written report of matched quality for the same defect or task; (3) reported the producer's effort, because a medium that is cheaper to produce can change which reports exist at all; (4) included testers, designers, reviewers and maintainers as consumers; and (5) reported replication packages, so that the next map can code design rather than guess it. For the growing group of studies in which a tool or agent consumes the recording, the open question is whether the output helps the person who asked for it. How such agents are given their tasks, including by video demonstration, is mapped separately for 85 GUI and computer-use agent benchmark families [69].

What the map implies for practice. The map does not support a general instruction to record rather than write, or the reverse. It does show where the evidence is thin enough that local trials are reasonable: in bug triage, for example, a team can compare the time to reproduce and the number of clarifying questions for reports with and without a recording. The author has a commercial interest in the subject, and the conclusions are deliberately limited to what the included studies show.

6 Limitations

Literature coverage. This study draws on accessible scholarly sources rather than an exhaustive search of all major academic databases. Some potentially relevant publications could not be fully assessed.

Evidence quality. Studies were reviewed at different levels of detail, including abstracts and selected full-text passages. Consequently, some classifications and reported findings may be incomplete. The analysis provides a structured map of the available literature rather than definitive estimates of effectiveness.

Classification and scope. The author reviewed the AI-assisted coding. Some publications have overlapping versions or borderline eligibility, and classification decisions may affect the reported distributions.

Interpretation. The findings describe patterns in the reviewed literature. They do not establish causal relationships, and comparisons between communication formats should not be interpreted as pooled estimates of effectiveness.

7 Conclusion

Research on screen recordings in software engineering is large, scattered across separate vocabularies and weighted toward tutorials, requirements videos and tool papers. 198 studies were identified. Comparisons of a recording with text are few (19 studies) and point in different directions, and the outcomes that matter to the person receiving a report are rarely measured. The evidence is not yet enough to say when a recording is better than text, a screenshot or a call. It is enough to say what a better test would measure, and the open dataset and the gap map give the next study a place to start.

Statements

Data and materials availability. The merged dataset, the evidence register (verbatim quotes and locations for key findings, data bases, methods and some effect directions), the codebook, the search strategy, the search log, the screening decisions, the correction log and the analysis scripts are released on Zenodo (dataset DOI: https://doi.org/10.5281/zenodo.24222760; data CC BY 4.0, code MIT). The full texts of the included studies are third-party works and are not redistributed; document hashes and URLs allow verification.

Competing interests. The author is affiliated with Vertov (`https://vertov.app`), a company developing screen-recording and AI-agent collaboration software. Vertov has a commercial interest in the subject of this research.

Funding. This research was funded by Vertov. No external funding was received.

Use of AI. Claude (Anthropic) was used to assist with literature research, data extraction and classification, analysis, verification checks and manuscript preparation. The methods and limitations of AI-assisted coding are described in Sections 3 and 6.

Ethics. The study analyzed published research papers. No human participants and no personal data were involved.

Author contributions. Ankur Shrestha: Conceptualization, methodology, investigation, analysis, writing and project supervision.

References

[1] M. Nayebi and B. Adams, "Image-based communication on social coding platforms," Journal of Software: Evolution and Process, 2023. [Online]. Available: https://doi.org/10.1002/smr.2609

[2] O. Creighton, M. Ott, and B. Bruegge, "Software Cinema-Video-based Requirements Engineering," in *14th IEEE International Requirements Engineering Conference (RE'06)*, 2006. [Online]. Available: https://doi.org/10.1109/re.2006.59

[3] H. Kuramoto, D. Wang, M. Kondo, Y. Kashiwa, Y. Kamei, and N. Ubayashi, "Understanding the characteristics and the role of visual issue reports," *Empirical Software Engineering*, 2024. [Online]. Available: https://doi.org/10.1007/s10664-024-10459-3

[4] N. Cooper, C. Bernal-Cárdenas, O. Chaparro, K. Moran, and D. Poshyvanyk, "It Takes Two to Tango: Combining Visual and Textual Information for Detecting Duplicate Video-Based Bug Reports," in *ACM/IEEE International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*, 2021. [Online]. Available: https://doi.org/10.1109/icse43902.2021.00091

[5] L. MacLeod, A. Bergen, and M.-A. Storey, "Documenting and sharing software knowledge using screencasts," *Empirical Software Engineering*, 2017. [Online]. Available: https://doi.org/10.1007/s10664-017-9501-9

[6] L. MacLeod, M. Storey, and A. Bergen, "Code, Camera, Action: How Software Developers Document and Share Program Knowledge Using YouTube," in *2015 IEEE 23rd International Conference on Program Comprehension (ICPC)*, 2015. [Online]. Available: https://doi.org/10.1109/icpc.2015.19

[7] O. Karras, K. Schneider, and S. A. Fricker, "Representing software project vision by means of video: A quality model for vision videos," *Journal of Systems and Software*, 2020. [Online]. Available: https://doi.org/10.1016/j.jss.2019.110479

[8] H. v. d. Meij and J. v. d. Meij, "Demonstration-based training (DBT) in the design of a video tutorial for software training," *Instructional Science*, 2016. [Online]. Available: https://doi.org/10.1007/s11251-016-9394-9

[9] M. Noetel, S. Griffith, O. Delaney, T. Sanders, P. Parker, B. del Pozo Cruz, and C. Lonsdale, "Video Improves Learning in Higher Education: A Systematic Review," Review of Educational Research, 2021. [Online]. Available: https://doi.org/10.3102/0034654321990713

[10] M. Noetel, S. Griffith, O. Delaney, N. R. Harris, T. Sanders, P. Parker, B. del Pozo Cruz, and C. Lonsdale, "Multimedia Design for Learning: An Overview of Reviews With Meta-Meta-Analysis," Review of Educational Research, 2021. [Online]. Available: https://doi.org/10.3102/00346543211052329

[11] G. Long, J. Gong, H. Fang, and T. Chen, "Learning Software Bug Reports: A Systematic Literature Review," ACM Transactions on Software Engineering and Methodology, 2026. [Online]. Available: https://doi.org/10.1145/3750040

[12] D. Wang, M. Galster, and M. Morales-Trujillo, "A systematic mapping study of bug reproduction and localization," Information and Software Technology, 2024. [Online]. Available: https://doi.org/10.1016/j.infsof.2023.107338

[13] S. Penn and N. Brown, "Is Screencast Feedback Better Than Text Feedback for Student Learning in Higher Education? : A Systematic Review," Ubiquitous Learning: An International Journal, 2022. [Online]. Available: https://doi.org/10.18848/1835-9795/cgp/v15i02/1-18

[14] R. Völkner, "Entwicklung eines konzepts zur beurteilung von gemeinsamen verständnis mittels vision videos in requirements engineering basierend auf einer systematischen literatursuche," Master's thesis, Gottfried Wilhelm Leibniz Universität Hannover, Institut für Praktische Informatik, Fachgebiet Software Engineering, Hannover, Germany, Nov. 2020, in German. English title: Development of a Concept for Assessing Shared Understanding by Means of Vision Videos in Requirements Engineering Based on a Systematic Literature Review. Master's thesis. [Online]. Available: https://doi.org/10.15488/11961

[15] O. Karras, "Supporting requirements communication for shared understanding by applying vision videos in requirements engineering," Ph.D. dissertation, Gottfried Wilhelm Leibniz Universität Hannover, Berlin, Germany, 2021, doctoral dissertation; no DOI. [Online]. Available: https://www.logos-verlag.de/cgi-bin/engbuchmid?isbn=5297&lng=deu&id=

[16] M. Pejović, N. Vasović, A. Cicchetti, and R. Jongeling, "Mapping educational software engineering content on YouTube," in *Proceedings of the IEEE/ACM 48th International Conference on Software Engineering (ICSE-SEET '26), pp. 174-184*, 2026. [Online]. Available: https://doi.org/10.1145/3786580.3786974

[17] H. H. Clark and S. E. Brennan, "Grounding in communication," Perspectives on socially shared cognition, 1991. [Online]. Available: https://doi.org/10.1037/10096-006

[18] R. E. Kraut, S. R. Fussell, and J. Siegel, "Visual Information as a Conversational Resource in Collaborative Physical Tasks," Human–Computer Interaction, 2003. [Online]. Available: https://doi.org/10.1207/s15327051hci1812_2

[19] D. Gergle, R. E. Kraut, and S. R. Fussell, "Using Visual Information for Grounding and Awareness in Collaborative Tasks," Human-Computer Interaction, 2013. [Online]. Available: https://collablab.northwestern.edu/pubs/HCI2013-GergleKrautFussell-UsingVisualInformationforGroundingandAwarenessinCollaborativeTasks.pdf

[20] A. R. Dennis, R. M. Fuller, and J. S. Valacich, "Media, Tasks, and Communication Processes: A Theory of Media Synchronicity1," MIS Quarterly, 2008. [Online]. Available: https://doi.org/10.2307/25148857

[21] R. E. Mayer and R. Moreno, "Nine Ways to Reduce Cognitive Load in Multimedia Learning," Educational Psychologist, 2003. [Online]. Available: https://doi.org/10.1207/s15326985ep3801_6

[22] U. B. Torun, M. T. Demircan, M. F. Gön, and E. Tüzün, "Past, Present, and Future of Bug Tracking in the Generative AI Era," ACM Transactions on Software Engineering and Methodology, 2026. [Online]. Available: https://doi.org/10.1145/3806655

[23] N. S. M. Yusop, J. Grundy, and R. Vasa, "Reporting Usability Defects: A Systematic Literature Review," IEEE Transactions on Software Engineering, 2017. [Online]. Available: https://doi.org/10.1109/tse.2016.2638427

[24] T. Theunissen, U. van Heesch, and P. Avgeriou, "A mapping study on documentation in Continuous Software Development," Information and Software Technology, 2022. [Online]. Available: https://doi.org/10.1016/j.infsof.2021.106733

[25] I. Santos, K. R. Felizardo, I. Steinmacher, and M. A. Gerosa, "Software solutions for newcomers' onboarding in software projects: A systematic literature review," Information and Software Technology, 2025. [Online]. Available: https://doi.org/10.1016/j.infsof.2024.107568

[26] N. Davila and I. Nunes, "A systematic literature review and taxonomy of modern code review," Journal of Systems and Software, 2021. [Online]. Available: https://doi.org/10.1016/j.jss.2021.110951

[27] I. M. Miake-Lye, S. Hempel, R. Shanman, and P. G. Shekelle, "What is an evidence map? A systematic review of published evidence maps and their definitions, methods, and products," Systematic Reviews, 2016. [Online]. Available: https://doi.org/10.1186/s13643-016-0204-x

[28] H. Arksey and L. O'Malley, "Scoping studies: towards a methodological framework," International Journal of Social Research Methodology, 2005. [Online]. Available: https://doi.org/10.1080/1364557032000119616

[29] A. C. Tricco, E. Lillie, W. Zarin, K. K. O'Brien, H. Colquhoun, D. Levac, D. Moher, M. D. Peters, T. Horsley, L. Weeks, S. Hempel, E. A. Akl, C. Chang, J. McGowan, L. Stewart, L. Hartling, A. Aldcroft, M. G. Wilson, C. Garritty, S. Lewin, C. M. Godfrey, M. T. Macdonald, E. V. Langlois, K. Soares-Weiser, J. Moriarty, T. Clifford, Ö. Tunçalp, and S. E. Straus, "PRISMA Extension for Scoping Reviews (PRISMA-ScR): Checklist and Explanation," Annals of Internal Medicine, 2018. [Online]. Available: https://doi.org/10.7326/m18-0850

[30] K. Petersen, S. Vakkalanka, and L. Kuzniarz, "Guidelines for conducting systematic mapping studies in software engineering: An update," Information and Software Technology, 2015. [Online]. Available: https://doi.org/10.1016/j.infsof.2015.03.007

[31] B. Kitchenham and S. Charters, "Guidelines for performing Systematic Literature Reviews in Software Engineering (Version 2.3, EBSE Technical Report EBSE-2007-01)," Keele University and University of Durham technical report, 2007. [Online]. Available: https://www.elsevier.com/___data/promis_misc/525444systematicreviewsguide.pdf

[32] M. Campbell, J. E. McKenzie, A. Sowden, S. V. Katikireddi, S. E. Brennan, S. Ellis, J. Hartmann-Boyce, R. Ryan, S. Shepperd, J. Thomas, V. Welch, and H. Thomson, "Synthesis without meta-analysis (SWiM) in systematic reviews: reporting guideline," BMJ, 2020. [Online]. Available: https://doi.org/10.1136/bmj.l6890

[33] C. Wohlin, "Guidelines for snowballing in systematic literature studies and a replication in software engineering," Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering, 2014. [Online]. Available: https://doi.org/10.1145/2601248.2601268

[34] C. Wohlin, M. Kalinowski, K. Romero Felizardo, and E. Mendes, "Successful combination of database search and snowballing for identification of primary studies in systematic literature studies," Information and Software Technology, 2022. [Online]. Available: https://doi.org/10.1016/j.infsof.2022.106908

[35] V. Garousi, M. Felderer, and M. V. Mäntylä, "Guidelines for including grey literature and conducting multivocal literature reviews in software engineering," Information and Software Technology, 2019. [Online]. Available: https://doi.org/10.1016/j.infsof.2018.09.006

[36] Q. Khraisha, S. Put, J. Kappenberg, A. Warraitch, and K. Hadfield, "Can large language models replace humans in systematic reviews? Evaluating GPT -4's efficacy in screening and extracting data from peer-reviewed and grey literature in multiple languages," Research Synthesis Methods, 2024. [Online]. Available: https://doi.org/10.1002/jrsm.1715

[37] Y. Yan, N. Cooper, O. Chaparro, K. Moran, and D. Poshyvanyk, "Semantic GUI Scene Learning and Video Alignment for Detecting Duplicate Video-based Bug Reports," in *ACM/IEEE International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*, 2024. [Online]. Available: https://doi.org/10.1145/3597503.3639163

[38] C. Bernal-Cárdenas, N. Cooper, K. Moran, O. Chaparro, A. Marcus, and D. Poshyvanyk, "Translating video recordings of mobile app usages into replayable scenarios," in *ACM/IEEE International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*, 2020. [Online]. Available: https://doi.org/10.1145/3377811.3380328

[39] S. Feng, M. Xie, Y. Xue, and C. Chen, "Read It, Don't Watch It: Captioning Bug Recordings Automatically," in *ACM/IEEE International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*, 2023. [Online]. Available: https://doi.org/10.1109/icse48619.2023.00197

[40] L. Bao, Z. Xing, X. Xia, and D. Lo, "VT-Revolution: Interactive Programming Video Tutorial Authoring and Watching System," *IEEE Transactions on Software Engineering*, 2019. [Online]. Available: https://doi.org/10.1109/tse.2018.2802916

[41] O. Karras and K. Schneider, "Software Professionals are Not Directors: What Constitutes a Good Video?" in *2018 1st International Workshop on Learning from other Disciplines for Requirements Engineering (D4RE)*, 2018. [Online]. Available: https://doi.org/10.1109/d4re.2018.00011

[42] A. Alaboudi and T. D. LaToza, "An Exploratory Study of Live-Streamed Programming," in *2019 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, 2019. [Online]. Available: https://doi.org/10.1109/vlhcc.2019.8818832

[43] V. Ragazou and I. Karasavvidis, "Optimizing Video Tutorials for Software Training through Cueing," in *IADIS International Conferences Mobile Learning and Educational Technologies (ML and ICEduTech)*, 2021. [Online]. Available: https://doi.org/10.33965/ml_icedutech2021_202102l016

[44] ——, "Considering visual cueing and practice type for software training's optimisation: a study of domain experts," *Interactive Technology and Smart Education*, 2021. [Online]. Available: https://doi.org/10.1108/itse-05-2021-0091

[45] N. Garrett, "Segmentation's failure to improve software video tutorials," *British Journal of Educational Technology*, 2020. [Online]. Available: https://doi.org/10.1111/bjet.13000

[46] Y. Liu, Z. Wang, H. Chen, X. Sun, X. Yu, J. Wu, J. Liu, E. Barsoum, Z. Liu, and S. Chang, "Learning from Online Videos at Inference Time for Computer-Use Agents," arXiv, 2025. [Online]. Available: http://arxiv.org/abs/2511.04137v1

[47] J. Shi, J. Mönnich, J. Klünder, and K. Schneider, "Organizing Graphical User Interface tests from behavior-driven development as videos to obtain stakeholders' feedback," *Journal of Software: Evolution and Process*, 2024. [Online]. Available: https://doi.org/10.1002/smr.2721

[48] ——, "Using GUI Test Videos to Obtain Stakeholders' Feedback," in *2023 IEEE/ACM International Conference on Software and System Processes (ICSSP)*, 2023. [Online]. Available: https://doi.org/10.1109/icssp59042.2023.00014

[49] O. Karras, A. Hamadeh, and K. Schneider, "Enriching Requirements Specifications with Videos - The Use of Videos to Support Requirements Communication," Softwaretechnik-Trends, 2018. [Online]. Available: https://fb-swt.gi.de/fileadmin/FB/SWT/Softwaretechnik-Trends/Verzeichnis/Band_38_Heft_1/10_paper_3.pdf

[50] S. A. Lloyd and C. L. Robertson, "Screencast Tutorials Enhance Student Learning of Statistics," *Teaching of Psychology*, 2011. [Online]. Available: https://doi.org/10.1177/0098628311430640

[51] K. Schneider, M. Busch, O. Karras, M. Schrapel, and M. Rohs, "Refining Vision Videos," in *Lecture Notes in Computer Science*, 2019. [Online]. Available: https://doi.org/10.1007/978-3-030-15538-4_10

[52] V. Käfer, D. Kulesz, and S. Wagner, "What Is the Best Way For Developers to Learn New Software Tools? An Empirical Comparison Between a Text and a Video Tutorial," *The Art, Science, and Engineering of Programming*, 2017. [Online]. Available: https://doi.org/10.22152/programming-journal.org/2017/1/17

[53] J. v. d. Meij and H. v. d. Meij, "A test of the design of a video tutorial for software training," *Journal of Computer Assisted Learning*, 2014. [Online]. Available: https://doi.org/10.1111/jcal.12082

[54] J. Shi, O. Karras, M. Obaidi, and M. Tandun, "Can Videos as a By-Product of GUI Testing Help Developers Understand GUI Tests?" in *IEEE International Requirements Engineering Conference Workshops (REW)*, 2023. [Online]. Available: https://doi.org/10.1109/rew57809.2023.00031

[55] J. Worlitz, A. Stabler, and R. Woll, "The Usage of Video Tutorials, Personal Support and Written Instructions for Knowledge Acquisition and Refreshment," *Quality Innovation Prosperity*, 2018. [Online]. Available: https://doi.org/10.12776/qip.v22i2.1065

[56] E. Paikari, "Bug Report Quality Prediction and the Impact of Including Videos on the Bug Reporting Process," UC Irvine Electronic Theses and Dissertations (eScholarship, qt9tn4t5gs), 2023. [Online]. Available: https://escholarship.org/uc/item/9tn4t5gs

[57] H. Kuramoto, M. Kondo, Y. Kashiwa, Y. Ishimoto, K. Shindo, Y. Kamei, and N. Ubayashi, "Do visual issue reports help developers fix bugs?" in *Proceedings of the 30th IEEE/ACM International Conference on Program Comprehension*, 2022. [Online]. Available: https://doi.org/10.1145/3524610.3527882

[58] M. Nørgaard and K. Hornbæk, "Exploring the Value of Usability Feedback Formats," *International Journal of Human-Computer Interaction*, 2009. [Online]. Available: https://doi.org/10.1080/10447310802546708

[59] R. T. Høegh, "Software Development and Feedback from Usability Evaluations," in *Interdisciplinary Aspects of Information Systems Studies*, 2008. [Online]. Available: https://doi.org/10.1007/978-3-7908-2010-2_7

[60] L. S. Mestre, "Student preference for tutorial design: a usability study," *Reference Services Review*, 2012. [Online]. Available: https://doi.org/10.1108/00907321211228318

[61] A. J. Tayeb, M. D. Alahmadi, E. Tajik, and S. Haiduc, "Investigating Developers' Preferences for Learning and Issue Resolution Resources in the ChatGPT Era," IEEE International Conference on Software Maintenance (ICSM), 2024. [Online]. Available: https://doi.org/10.1109/icsme58944.2024.00045

[62] Y. Ghilay, "Computer Courses In Higher-Education: Improving Learning By Screencast Technology," *i-manager's Journal of Educational Technology*, 2015. [Online]. Available: https://doi.org/10.26634/jet.11.4.3148

[63] M. Obaidi, J. Fischbach, M. Herrmann, H. Deters, J. Droste, J. Klünder, and K. Schneider, "How Does Users' App Knowledge Influence the Preferred Level of Detail and Format of Software Explanations?" Requirements Engineering: Foundation for Software Quality (Lecture Notes in Computer Science), Springer Nature Switzerland, pp. 106-122, 2025. [Online]. Available: https://doi.org/10.1007/978-3-031-88531-0_8

[64] J. Worlitz, A. Stabler, S. Peplowsky, and R. Woll, "Video Tutorials: An Appropriate Way of Teaching QM Tools Applied with Software," *Quality Innovation Prosperity*, 2016. [Online]. Available: https://doi.org/10.12776/qip.v20i2.754

[65] S. Feng and C. Chen, "GIFdroid: Automated Replay of Visual Bug Reports for Android Apps," in *Proceedings of the 44th International Conference on Software Engineering (ICSE 2022)*, 2022. [Online]. Available: https://doi.org/10.1145/3510003.3510048

[66] L. Nagel, O. Karras, S. M. Amiri, and K. Schneider, "Supporting Shared Understanding in Asynchronous Communication Contexts," in *Lecture Notes in Computer Science*, 2023. [Online]. Available: https://doi.org/10.1007/978-3-031-29786-1_3

[67] ——, "Turning asynchronicity into an opportunity: asynchronous communication for shared understanding with vision videos," *Requirements Engineering*, 2024. [Online]. Available: https://doi.org/10.1007/s00766-024-00414-5

[68] L. Nagel, J. Shi, and M. Busch, "Viewing Vision Videos Online: Opportunities for Distributed Stakeholders," in *2021 IEEE 29th International Requirements Engineering Conference Workshops (REW)*, 2021. [Online]. Available: https://doi.org/10.1109/rew53955.2021.00054

[69] A. Shrestha, "How GUI agents receive instructions: An exploratory mapping of 85 benchmark families," Nov. 2026. [Online]. Available: https://doi.org/10.5281/zenodo.24217329

[70] V. Agrawal, Y.-H. Lin, and J. Cheng, "Understanding the Characteristics of Visual Contents in Open Source Issue Discussions: A Case Study of Jupyter Notebook," in *International Conference on Evaluation and Assessment in Software Engineering (EASE)*, 2022. [Online]. Available: https://doi.org/10.1145/3530019.3534082

[71] A. Alaboudi and T. D. LaToza, "Supporting Software Engineering Research and Education by Annotating Public Videos of Developers Programming," in *International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE)*, 2019. [Online]. Available: https://doi.org/10.1109/chase.2019.00035

[72] ——, "An Exploratory Study of Debugging Episodes," in *arXiv*, 2021. [Online]. Available: http://arxiv.org/abs/2105.02162v1

[73] M. D. Alahmadi, "VID2META: Complementing Android Programming Screencasts with Code Elements and GUIs," *Mathematics*, 2022. [Online]. Available: https://doi.org/10.3390/math10173175

[74] ——, "VID2XML: Automatic Extraction of a Complete XML Data From Mobile Programming Screencasts," *IEEE Transactions on Software Engineering*, 2023. [Online]. Available: https://doi.org/10.1109/tse.2022.3188898

[75] ——, "Evaluating Large Language Models for Generating YouTube Programming Tutorial Metadata From Video Transcripts," *IEEE Access*, 2026. [Online]. Available: https://doi.org/10.1109/access.2026.3696617

[76] M. D. Alahmadi and M. Alshangiti, "Optimizing OCR Performance for Programming Videos: The Role of Image Super-Resolution and Large Language Models," *Mathematics*, 2024. [Online]. Available: https://doi.org/10.3390/math12071036

[77] M. D. Alahmadi, A. J. Tayeb, A. Malkadi, E. Parra, and S. Haiduc, "UIScreens: extracting user interface screens from mobile programming video tutorials," in *ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE)*, 2020. [Online]. Available: https://doi.org/10.1145/3368089.3417935

[78] M. D. Alahmadi, A. Khormi, B. Parajuli, J. Hassel, S. Haiduc, and P. Kumar, "Code Localization in Programming Screencasts," *Empirical Software Engineering*, 2020. [Online]. Available: https://doi.org/10.1007/s10664-019-09759-w

[79] M. D. Alahmadi, K. T. Mursi, M. A. Alqarni, A. J. Tayeb, and F. S. Alsubaei, "Analyzing and Categorization Developer Intent on Twitch Live Chat," *Programming and Computer Software*, 2024. [Online]. Available: https://doi.org/10.1134/s0361768824700191

[80] O. AlGhamdi, A. AlGhamdi, F. Alorabi, M. Alahmadi, A. Alzahrani, and A. Almazroui, "VidSumEval: A Web Platform and Benchmark for Evaluating AI-Generated Programming Video Summaries," in *Proceedings of the 41st IEEE/ACM International Conference on Automated Software Engineering*, 2026. [Online]. Available: https://doi.org/10.1145/3832783.3834594

[81] V. C. O. Aureliano, "Self-explaining from video recordings as a methodology for learning programming," in *ACM Conference on International Computing Education Research (ICER)*, 2014. [Online]. Available: https://doi.org/10.1145/2632320.2632321

[82] Z. Bai, Z. Chen, Y. Lin, K. Q. Lin, D. Gao, X. Guo, X. Wang, and M. Z. Shou, "Demo2Tutorial: From Human Experience to Multimodal Software Tutorials," arXiv, 2026. [Online]. Available: http://arxiv.org/abs/2606.03951v1

[83] L. Bao, J. Li, Z. Xing, X. Wang, X. Xia, and B. Zhou, "Extracting and analyzing time-series HCI data from screen-captured task videos," *Empirical Software Engineering*, 2016. [Online]. Available: https://doi.org/10.1007/s10664-015-9417-1

[84] L. Bao, Z. Xing, X. Xia, D. Lo, M. Wu, and X. Yang, "psc2code," *ACM Transactions on Software Engineering and Methodology*, 2020. [Online]. Available: https://doi.org/10.1145/3392093

[85] Z. J. Beasley and A. R. Johnson, "The Impact of Remote Pair Programming in an Upper-Level CS Course," in *Proceedings of the 27th ACM Conference on on Innovation and Technology in Computer Science Education Vol. 1*, 2022. [Online]. Available: https://doi.org/10.1145/3502718.3524772

[86] C. Bernal-Cárdenas, N. Cooper, M. Havranek, K. Moran, O. Chaparro, D. Poshyvanyk, and A. Marcus, "Translating Video Recordings of Complex Mobile App UI Gestures into Replayable Scenarios," *IEEE Transactions on Software Engineering*, 2023. [Online]. Available: https://doi.org/10.1109/tse.2022.3192279

[87] G. Broll, H. Hußmann, E. Rukzio, and R. Wimmer, "Using Video Clips to Support Requirements Elicitation in Focus Groups - An Experience Report." in *SE 2007 Workshop on Multimedia Requirements Engineering, Hamburg, 27 March 2007*, 2007. [Online]. Available: https://eprints.lancs.ac.uk/id/eprint/41903/

[88] B. Bruegge, H. Stangl, and M. Reiss, "An experiment in teaching innovation in software engineering," Companion to the 23rd ACM SIGPLAN conference on Object-oriented programming systems languages and applications, 2008. [Online]. Available: https://doi.org/10.1145/1449814.1449868

[89] K. Buffardi and R. Wang, "Integrating Videos with Programming Practice," in *Annual Conference on Innovation and Technology in Computer Science Education*, 2022. [Online]. Available: https://doi.org/10.1145/3502718.3524778

[90] C. S. Cheah, "The moderating effect of logic in the learning of C++ computer programming using screencasting," *Malaysian Online Journal of Educational Technology*, 2019. [Online]. Available: https://doi.org/10.17220/mojet.2019.02.006

[91] C.-S. Cheah, "Does modality principle occur in the learning of C++ computer programming using screencasting?" *Journal of Educational Sciences & Psychology*, 2021. [Online]. Available: https://doi.org/10.51865/jesp.2021.2.07

[92] C. S. Cheah and L.-M. Leong, "Investigating the Redundancy Effect in the Learning of C++ Computer Programming Using Screencasting," *International Journal of Modern Education and Computer Science*, 2019. [Online]. Available: https://doi.org/10.5815/ijmecs.2019.06.03

[93] C.-Y. Chen, "The Influence of Representational Formats and Learner Modality Preferences on Instructional Efficiency Using Interactive Video Tutorials," *Journal of Education and Training*, 2020. [Online]. Available: https://doi.org/10.5296/jet.v7i2.17415

[94] Y. Chen, W. S. Lasecki, and T. Dong, "Towards Supporting Programming Education at Scale via Live Streaming," *Proceedings of the ACM on Human-Computer Interaction*, 2021. [Online]. Available: https://doi.org/10.1145/3434168

[95] J. Chen, A. Swearngin, J. H. Y. Wu, T. Barik, J. W. Nichols, and X. Zhang, "Extracting Replayable Interactions from Videos of Mobile App Usage," arXiv (Cornell University), 2022. [Online]. Available: https://doi.org/10.48550/arxiv.2207.04165

[96] O. Cherkashyn, "Enhancing Web Test Automation Through Screenshot and Video-Based Execution Reporting," Digitalen Hochschulbibliothek Sachsen-Anhalt (Universitäts- und Landesbibliothek Sachsen-Anhalt), 2026. [Online]. Available: https://doi.org/10.25673/123848

[97] P.-Y. Chi, "Designing Video-Based Interactive Instructions," PhD dissertation, EECS Department, University of California, Berkeley; Technical Report UCB/EECS-2016-150, 15 Aug 2016, 2016. [Online]. Available: https://www2.eecs.berkeley.edu/Pubs/TechRpts/2016/EECS-2016-150.html

[98] D. Costa, G. Matos, A. Lins, L. Barroso, C. Aguiar, and E. Bezerra, "Web Application Interface Data Collector for Issue Reporting," *Advances in Science Technology and Engineering Systems Journal*, 2024. [Online]. Available: https://doi.org/10.25046/aj090501

[99] I. Drosos and P. J. Guo, "Streamers Teaching Programming, Art, and Gaming: Cognitive Apprenticeship, Serendipitous Teachable Moments, and Tacit Expert Knowledge," in *2021 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, 2021. [Online]. Available: https://doi.org/10.1109/vl/hcc51201.2021.9576481

[100] I. Drosos, A. Sarkar, and A. D. Gordon, ""My toxic trait is thinking I'll remember this": gaps in the learner experience of video tutorials for feature-rich software," arXiv (Cornell University), 2024. [Online]. Available: https://doi.org/10.48550/arxiv.2404.07114

[101] B. J. Dybvik, "An examination of student perceptions of lynda.com software tutorial training," University of Wisconsin-Stout (Master's thesis, Plan B), 2008. [Online]. Available: https://minds.wisc.edu/items/f4426439-9b77-4834-bd31-44a9f112b407

[102] J. C. Désiron, T. Endres, and S. Schneider, "Is it not too redundant? When signaling overlap reduces extraneous load and enhances retention in a software video tutorial," *Frontiers in Psychology*, 2026. [Online]. Available: https://doi.org/10.3389/fpsyg.2026.1795142

[103] M. Ellmann, A. Oeser, D. Fucci, and W. Maalej, "Find, understand, and extend development screencasts on YouTube," in *Proceedings of the 3rd ACM SIGSOFT International Workshop on Software Analytics*, 2017. [Online]. Available: https://doi.org/10.1145/3121257.3121260

[104] J. Escobar-Avila, E. Parra, and S. Haiduc, "Text Retrieval-Based Tagging of Software Engineering Video Tutorials," in *2017 IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C)*, 2017. [Online]. Available: https://doi.org/10.1109/icse-c.2017.121

[105] N. Etemadi, "Toward A Novel Tool for Incorporating Video Into Effective Bug Reporting," UC Irvine Electronic Theses and Dissertations (eScholarship, qt0h25c29d), 2023. [Online]. Available: https://escholarship.org/uc/item/0h25c29d

[106] T. Faas, L. Dombrowski, A. Young, and A. D. Miller, "Watch me code: programming mentorship communities on Twitch.tv," *Proceedings of the ACM on Human-Computer Interaction*, 2018. [Online]. Available: https://doi.org/10.1145/3274319

[107] Y. Fan, Z. Di, Z. Wen, Y. Yang, M. Cheng, Q. Dai, B. Liu, K. Qiu, Y. Dong, J. Li, and C. Luo, "RESOURCE2SKILL: Distilling Executable Agent Skills from Human-Created Multimodal Resources," arXiv, 2026. [Online]. Available: https://doi.org/10.48550/arxiv.2606.29538

[108] P. Feijóo-García and C. Gardner-Mccune, "Using Student-created Instructional Videos in CS Upper-level Courses: A Successful Strategy in a Functional Programming Course," in *International Conference on Computer Supported Education*, 2020. [Online]. Available: https://doi.org/10.5220/0009416404120419

[109] S. Feng, C. Chen, and Z. Xing, "Video2Action: Reducing Human Interactions in Action Annotation of App Tutorial Videos," in *ACM Symposium on User Interface Software and Technology (UIST)*, 2023. [Online]. Available: https://doi.org/10.1145/3586183.3606778

[110] S. Feng, D. Wang, N. Tomic, T. Yu, A. Aleti, and C. Chen, "ViBR: Automated Bug Replay from Video-Based Reports using Vision-Language Models," in *Proceedings of the ACM on software engineering.*, 2026. [Online]. Available: https://doi.org/10.1145/3808152

[111] S. A. Fricker, K. Schneider, F. Fotrousi, and C. Thuemmler, "Workshop videos for requirements communication," *Requirements Engineering*, 2015. [Online]. Available: https://doi.org/10.1007/s00766-015-0231-5

[112] M. Galster, A. Mitrovic, and M. Gordon, "Toward enhancing the training of software engineering students and professionals using active video watching," in *Proceedings of the 40th International Conference on Software Engineering: Software Engineering Education and Training*, 2018. [Online]. Available: https://doi.org/10.1145/3183377.3183384

[113] A. Granowski, "The Aesthetic Components of GIFs in Online Instruction Analyzing GIFs in Help Center Instruction Based Articles," *Advances in Literary Study*, 2025. [Online]. Available: https://doi.org/10.4236/als.2025.132010

[114] K. R. Green, T. Pinder–Grover, and J. M. Millunchick, "Impact of Screencast Technology: Connecting the Perception of Usefulness and the Reality of Performance," *Journal of Engineering Education*, 2012. [Online]. Available: https://doi.org/10.1002/j.2168-9830.2012.tb01126.x

[115] X. Hu, Y. Huang, B. Liu, R. Wu, Y. O. Hu, A. J. Quigley, M. Fan, C. Yu, and Y. Shi, "SmartRecorder: An IMU-based Video Tutorial Creation by Demonstration System for Smartphone Interaction Tasks," in *ACM International Conference on Intelligent User Interfaces (IUI)*, 2023. [Online]. Available: https://doi.org/10.1145/3581641.3584069

[116] Y. Huang, H. Li, Y. Cao, Z. Cai, T. Xie, Y. Gao, P. Guo, C. Zhang, B. Li, R. Zheng, H. Yan, and N. Liao, "Work2Skill: Capturing Human GUI Experience for Reusable Agent Skill Induction," arXiv, 2026. [Online]. Available: http://arxiv.org/abs/2610.12984v1

[117] W. Jang, H. Gao, T. Michaeli, and E. Kasneci, "Understanding Visual Attention and Communication in Distributed Pair Programming for Programming Education," in *2026 7th International Conference on Information Technology and Education Technology (ITET)*, 2026. [Online]. Available: https://doi.org/10.1109/itet70052.2026.11650705

[118] T. Jie, "The effectiveness of video tutorial and preview on self-efficacy, task performance and learning : an experimental study conducted at a middle school in Shanghai, China," University of Twente (MSc thesis, Educational Science and Technology), 2015. [Online]. Available: https://purl.utwente.nl/essays/69309

[119] M. Jirotka and P. Luff, "Supporting requirements with video-based analysis," *IEEE Software*, 2006. [Online]. Available: https://doi.org/10.1109/ms.2006.84

[120] A. Kadriu, L. A. Bexheti, H. A. Alili, and V. Ramadani, "Investigating trends in learning programming using YouTube tutorials," *International Journal of Learning and Change*, 2020. [Online]. Available: https://doi.org/10.1504/ijlc.2020.106721

[121] O. Karras, "Software Professionals' Attitudes Towards Video as a Medium in Requirements Engineering," in *Lecture Notes in Computer Science*, 2018. [Online]. Available: https://doi.org/10.1007/978-3-030-03673-7_11

[122] O. Karras and K. Schneider, "An Interdisciplinary Guideline for the Production of Videos and Vision Videos by Software Professionals," arXiv preprint, Tech. Rep., 2020. [Online]. Available: https://doi.org/10.48550/arxiv.2001.06675

[123] O. Karras, S. Kiesling, and K. Schneider, "Supporting Requirements Elicitation by Tool-Supported Video Analysis," in *2016 IEEE 24th International Requirements Engineering Conference (RE)*, 2016. [Online]. Available: https://doi.org/10.1109/re.2016.10

[124] O. Karras, J. Klünder, and K. Schneider, "Enrichment of Requirements Specifications with Videos - Enhancing the Comprehensibility of Textual Requirements," in *Zenodo (TPDL 2016 author copy)*, 2016. [Online]. Available: https://doi.org/10.5281/zenodo.59011

[125] O. Karras, C. Unger-Windeler, L. Glauer, and K. Schneider, "Video as a By-Product of Digital Prototyping: Capturing the Dynamic Aspect of Interaction," in *2017 IEEE 25th International Requirements Engineering Conference Workshops (REW)*, 2017. [Online]. Available: https://doi.org/10.1109/rew.2017.16

[126] O. Karras, J. Klünder, and K. Schneider, "Tool-Supported Experiments for Continuously Collecting Data of Subjective Video Quality Assessments During Video Playback," arXiv preprint, 2019. [Online]. Available: https://doi.org/10.48550/arxiv.1911.09091

[127] O. Karras, S. Polst, and K. Späth, "Using Vision Videos in a Virtual Focus Group: Experiences and Recommendations," arXiv preprint, 2020. [Online]. Available: https://doi.org/10.48550/arxiv.2011.00965

[128] O. Karras, E. Kristo, and J. Klunder, "The Potential of Using Vision Videos for CrowdRE: Video Comments as a Source of Feedback," in *2021 IEEE 29th International Requirements Engineering Conference Workshops (REW)*, 2021. [Online]. Available: https://doi.org/10.1109/rew53955.2021.00053

[129] P. Kefalas and I. Stamatopoulou, "Using screencasts to enhance coding skills: The case of logic programming?" *Computer Science and Information Systems*, 2018. [Online]. Available: https://doi.org/10.2298/csis180312028k

[130] K. Khandwala and P. J. Guo, "Codemotion: expanding the design space of learner interactions with computer programming tutorial videos," in *ACM Conference on Learning @ Scale (L@S)*, 2018. [Online]. Available: https://doi.org/10.1145/3231644.3231652

[131] V. Kharisma, "Design a Screencast Video for Software Learning in Higher Education (Case Study: Tutorial Video for Digital Illustration Course)," *JISA(Jurnal Informatika dan Sains)*, 2020. [Online]. Available: https://doi.org/10.31326/jisa.v3i1.638

[132] S. Kiesling, O. Karras, and K. A. Schneider, "ReqVidA - Requirements Video Analyzer." Softwaretechnik-Trends 36(3), 2016. [Online]. Available: https://dl.gi.de/handle/20.500.12116/40695

[133] E. Kokinda and D. M. Boyer, "The Other Side of the Screen: Motivations to Watch and Engage in Software Development Live Streams," in *arXiv*, 2025. [Online]. Available: https://doi.org/10.48550/arxiv.2511.02588

[134] E. Kokinda and P. Rodeghero, "Streaming software development: Accountability, community, and learning," Journal of Systems and Software, 2023. [Online]. Available: https://doi.org/10.1016/j.jss.2023.111630

[135] S. Kopczyńska and J. Shi, "Can Videos be Used to Communicate Non-functional Requirements? An Early Empirical Investigation," in *2023 IEEE 31st International*

Requirements Engineering Conference (RE), 2023. [Online]. Available: https://doi.org/10.1109/re57278.2023.00040

[136] D. Kueker and J. Moore, "Learning from screencast software tutorials: A comparison of cognitive load in dual and single-monitor learning environments," *Journal of Computer Assisted Learning*, 2023. [Online]. Available: https://doi.org/10.1111/jcal.12875

[137] B. T. Kumaravel, C. Nguyen, S. DiVerdi, and B. Hartmann, "TutoriVR: A Video-Based Tutorial System for Design Applications in Virtual Reality," in *CHI Conference on Human Factors in Computing Systems*, 2019. [Online]. Available: https://doi.org/10.1145/3290605.3300514

[138] C. Lamontagne, S. Sénécal, M. Frédette, É. Labonté-LeMoyne, and P. Léger, "The effect of the segmentation of video tutorials on User's training experience and performance," *Computers in Human Behavior Reports*, 2021. [Online]. Available: https://doi.org/10.1016/j.chbr.2021.100071

[139] B. Law, "Screencasts: enhancing coursework feedback for game programming students," in *ICCGI 2017: The Twelfth International Multi-Conference on Computing in the Global Information Technology, pp. 17-21 (ISBN 978-1-61208-571-5)*, 2017. [Online]. Available: https://www.thinkmind.org/index.php?view=article&articleid=iccgi_2017_2_30_10045

[140] M. J. Lee, S. Pradhan, and B. Dalgarno, "The Effectiveness of Screencasts and Cognitive Tools as Scaffolding for Novice Object-Oriented Programmers," *Journal of Information Technology Education Research*, 2008. [Online]. Available: https://doi.org/10.28945/179

[141] J. Li, L. Bao, Z. Xing, X. Wang, and B. Zhou, "BPMiner: mining developers' behavior patterns from screen-captured task videos," in *ACM Symposium on Applied Computing*, 2016. [Online]. Available: https://doi.org/10.1145/2851613.2851771

[142] K. Li, F. Chen, Z. Wang, S. Kim, H. Jin, and Y. Fu, "Screencast Tutorial Video Understanding," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition/Proceedings - IEEE Computer Society Conference on Computer Vision and Pattern Recognition/Proceedings*, 2020. [Online]. Available: https://doi.org/10.1109/cvpr42600.2020.01254

[143] W. G. Li, R. D. Pea, N. Haber, and H. Subramonyam, "Tutorly: Turning Programming Videos Into Apprenticeship Learning Environments with LLMs," arXiv (Cornell University), 2024. [Online]. Available: https://doi.org/10.48550/arxiv.2405.12946

[144] W. Li, R. Pea, N. Haber, and H. Subramonyam, "CogGen: A Learner-Centered Generative AI Architecture for Intelligent Tutoring with Programming Video," arXiv, 2025. [Online]. Available: http://arxiv.org/abs/2506.20600v1

[145] D. Lin, C. Bezemer, and A. E. Hassan, "Identifying gameplay videos that exhibit bugs in computer games," *Empirical Software Engineering*, 2019. [Online]. Available: https://doi.org/10.1007/s10664-019-09733-6

[146] Y. Lin, M. K. Yeh, and S.-R. Tan, "Teaching Programming by Revealing Thinking Process: Watching Experts' Live Coding Videos With Reflection Annotations," *IEEE Transactions on Education*, 2022. [Online]. Available: https://doi.org/10.1109/te.2022.3155884

[147] K. Q. Lin, L. Li, D. Gao, Q. WU, M. Yan, Z. Yang, L. Wang, and M. Z. Shou, "VideoGUI: A Benchmark for GUI Automation from Instructional Videos," arXiv, 2024. [Online]. Available: http://arxiv.org/abs/2406.10227v1

[148] S. Liu, S. Wang, and K. Sun, "Having Difficulty Understanding Manuals? Automatically Converting User Manuals into Instructional Videos," arXiv, 2023. [Online]. Available: http://arxiv.org/abs/2311.11031v2

[149] W. Liu, L. Guo, Y. W. Heng, C. Li, Tse-Hsun, Chen, and A. E. Hassan, "Screencast-Based Analysis of User-Perceived GUI Responsiveness," in *arXiv (Cornell University)*, 2025. [Online]. Available: https://doi.org/10.48550/arxiv.2508.01337

[150] W. Liu, F. Lin, L. Guo, T.-H. P. Chen, and A. E. Hassan, "GUIWatcher: Automatically Detecting GUI Lags by Analyzing Mobile Application Screencasts," in *IEEE/ACM International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*, 2025. [Online]. Available: https://doi.org/10.1109/icse-seip66354.2025.00010

[151] W. Lu, A. Senchenko, A. Hindle, and C.-P. Bezemer, "Automated Bug Frame Retrieval from Gameplay Videos Using Vision-Language Models," arXiv, 2025. [Online]. Available: https://doi.org/10.48550/arxiv.2508.04895

[152] D. Lu, Y. Xu, J. Wang, H. Wu, X. Wang, Z. Wang, J. Yang, H. Su, J. Chen, J. Chen, Y. Mao, J. Zhou, J. Lin, B. Hui, and T. Yu, "VideoAgentTrek: Computer Use Pretraining from Unlabeled Videos," arXiv, 2025. [Online]. Available: http://arxiv.org/abs/2510.19488v1

[153] S. Lubos, A. Felfernig, D. Garber, V.-M. Le, and M. Henrich, "Recommending Usability Improvements with Multimodal Large Language Models," *Proceedings of the ACM on Software Engineering 3(FSE), pp. 568-589*, 2026. [Online]. Available: https://doi.org/10.1145/3797121

[154] S. Luo, S. Luo, X. Guo, F. Zhang, J. Liang, and Y. Chen, "AutoCue: Multimodal LLM-Assisted Externalization of Implicit Inputs as Instructional Visual Cues in Screencast Tutorials," in *Graphics Interface*, 2026. [Online]. Available: https://doi.org/10.1145/3831423.3831452

[155] G. Majgaard, "Teaching Game Programming using Video Tutorials," in *2nd Exploring Teaching for Active Learning in Engineering Education (ETALEE), Odense, 23-24 May 2017 (poster, peer reviewed)*, 2017. [Online]. Available: https://portal.findresearcher.sdu.dk/da/publications/teaching-game-programming-using-video-tutorials-teaching-game-pro/

[156] A. Malkadi, M. D. Alahmadi, and S. Haiduc, "A Study on the Accuracy of OCR Engines for Source Code Transcription from Programming Screencasts," in *International Conference on Mining Software Repositories (MSR)*, 2020. [Online]. Available: https://doi.org/10.1145/3379597.3387468

[157] R. May, J. Daher, J. Krüger, and T. Leich, "An Experience Report on Using Video-Creation Tasks in Requirements-Engineering Education," in *Proceedings of the 46th International Conference on Software Engineering: Software Engineering Education and Training*, 2024. [Online]. Available: https://doi.org/10.1145/3639474.3640057

[158] A. McGowan, N. Anderson, G. Trombino, P. Sage, J. Adhikari, and L. Galway, "Learning to Code – Investigating the Adoption of Video-Based Lab Solutions for

University-Level Novice Programmers," in *ICERI proceedings*, 2023. [Online]. Available: https://doi.org/10.21125/iceri.2023.0380

[159] H. v. d. Meij, "Advance organizers in videos for software training of Chinese students," *British Journal of Educational Technology*, 2018. [Online]. Available: https://doi.org/10.1111/bjet.12619

[160] H. v. d. Meij and J. Maseland, "Practice schedules in a video-based software training arrangement," *Social Sciences & Humanities Open*, 2021. [Online]. Available: https://doi.org/10.1016/j.ssaho.2021.100133

[161] H. v. d. Meij and J. v. d. Meij, "The effects of reviews in video tutorials," *Journal of Computer Assisted Learning*, 2016. [Online]. Available: https://doi.org/10.1111/jcal.12136

[162] H. v. d. Meij and K. Nuketayeva, "Effects of practice schedules in video tutorials for software training," *Computers & Education*, 2023. [Online]. Available: https://doi.org/10.1016/j.compedu.2023.104786

[163] H. v. d. Meij and T. Voerman, "Effects of Instruction–Problem Arrangements in Video Tutorials for Software Training," *IEEE Transactions on Professional Communication*, 2024. [Online]. Available: https://doi.org/10.1109/tpc.2024.3449508

[164] H. v. d. Meij, J. v. d. Meij, T. Voerman, and E. F. Duipmans, "Supporting motivation, task performance and retention in video tutorials for software training," *Educational Technology Research and Development*, 2017. [Online]. Available: https://doi.org/10.1007/s11423-017-9560-z

[165] S. Mohorovicic and E. Tijan, "Using Screencasts in Computer Programming Courses," 2011. [Online]. Available: https://openalex.org/W2745314778

[166] P. Moslehi, B. Adams, and J. Rilling, "Feature location using crowd-based screencasts," in *International Conference on Mining Software Repositories (MSR)*, 2018. [Online]. Available: https://doi.org/10.1145/3196398.3196439

[167] ——, "A feature location approach for mapping application features extracted from crowd-based screencasts to source code," *Empirical Software Engineering*, 2020. [Online]. Available: https://doi.org/10.1007/s10664-020-09874-z

[168] E. R. Murphy-Hill, "Continuous social screencasting to facilitate software tool discovery," 2012 34th International Conference on Software Engineering (ICSE), 2012. [Online]. Available: https://doi.org/10.1109/icse.2012.6227090

[169] L. Nagel and O. Karras, "Keep Your Stakeholders Engaged: Interactive Vision Videos in Requirements Engineering," in *2021 IEEE 29th International Requirements Engineering Conference Workshops (REW)*, 2021. [Online]. Available: https://doi.org/10.1109/rew53955.2021.00014

[170] L. Nagel, A. Specht, and J. Bürst, "Encouraging Asynchronous Crowd Discussions Using an Interactive Video Player," in *2023 IEEE 31st International Requirements Engineering Conference Workshops (REW)*, 2023. [Online]. Available: https://doi.org/10.1109/rew57809.2023.00026

[171] L. Nagel, M. Schmedes, M. Ahrens, and K. Schneider, "When details are difficult to portray: enriching vision videos," *Requirements Engineering*, 2023. [Online]. Available: https://doi.org/10.1007/s00766-023-00403-0

[172] L. Nagel, J. Droste, A. Hess, and K. Schneider, "Paving the Way Towards an Effective Vision Video Usage: An Exploratory Study," in *2024 IEEE 32nd International Requirements Engineering Conference (RE)*, 2024. [Online]. Available: https://doi.org/10.1109/re59067.2024.00048

[173] A. Naumann, F. Hertlein, J. Höllig, L. Cazzonelli, and S. Thoma, "CodeSCAN: ScreenCast ANalysis for Video Programming Tutorials," arXiv (Cornell University), 2024. [Online]. Available: https://doi.org/10.48550/arxiv.2409.18556

[174] C. Nong, Q. Zhang, L. Huang, D. Cui, Q. Zheng, and T. Liu, "FVT: A Fragmented Video Tutor for "Dubbing" Software Development Tutorials," in *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET)*, 2019. [Online]. Available: https://doi.org/10.1109/icse-seet.2019.00018

[175] J. Ott, A. Atchison, P. Harnack, A. Bergh, and E. J. Linstead, "A deep learning approach to identifying source code in images and video," in *International Conference on Mining Software Repositories (MSR)*, 2018. [Online]. Available: https://doi.org/10.1145/3196398.3196402

[176] E. L. Ouh, B. Gan, and D. Lo, "ITSS: Interactive Web-Based Authoring and Playback Integrated Environment for Programming Tutorials," in *2022 IEEE/ACM 44th International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET)*, 2022. [Online]. Available: https://doi.org/10.1145/3510456.3514142

[177] Y. Pal and S. R. Iyer, "Classroom Versus Screencast for Native Language Learners: Effect of Medium of Instruction on Knowledge of Programming," in *ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE)*, 2015. [Online]. Available: https://doi.org/10.1145/2729094.2742618

[178] ——, "Effect of Medium of Instruction on Programming Ability Acquired through Screencast," in *International Conference on Learning and Teaching in Computing and Engineering (LaTICE)*, 2015. [Online]. Available: https://doi.org/10.1109/latice.2015.38

[179] G. E. Palaigeorgiou and T. Despotakis, "Known and Unknown Weaknesses in Software Animated Demonstrations (Screencasts): A Study in Self-Paced Learning Settings," *Journal of Information Technology Education Research*, 2010. [Online]. Available: https://doi.org/10.28945/1129

[180] K. Pankov, "Video tutorials – the educational media of tomorrow ? A comparison of the effectiveness of dynamic vs. static tutorial for procedural learning," Université de Genève (Master's thesis), 2018. [Online]. Available: https://archive-ouverte.unige.ch/unige:110082

[181] J. Park, Y. H. Park, and A. Oh, "Non-Linear Editor for Text-Based Screencast," in *Adjunct Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology (UIST '17 Adjunct), pp. 183-185*, 2017. [Online]. Available: https://doi.org/10.1145/3131785.3131831

[182] J. Park, Y. H. Park, J. Kim, J. Cha, S. Kim, and A. H. Oh, "Elicast: embedding interactive exercises in instructional programming screencasts," in *ACM Conference on Learning @ Scale (L@S)*, 2018. [Online]. Available: https://doi.org/10.1145/3231644.3231657

[183] E. Parra, "Developers and Modern Communication Mediums," in *IEEE International Conference on Software Maintenance (ICSM)*, 2023. [Online]. Available: https://doi.org/10.1109/icsme58846.2023.00078

[184] R. Perraud, "Adaptability of software tutorials to user interfaces," Actes joints de la Conférence Internationale Francophone sur l'Interaction Humain-Machine (IHM'24), AFIHM, 2024. [Online]. Available: https://hal.science/hal-04490515v1

[185] R. Perraud, A. Tabard, and S. Malacria, "Tutorial mismatches: investigating the frictions due to interface differences when following software video tutorials," in *Designing Interactive Systems Conference*, 2024. [Online]. Available: https://doi.org/10.1145/3643834.3661511

[186] R. Pham, S. Meyer, I. Kitzmann, and K. Schneider, "Interactive Multimedia Storyboard for Facilitating Stakeholder Interaction: Supporting Continuous Improvement in IT-ecosystems," in *2012 Eighth International Conference on the Quality of Information and Communications Technology*, 2012. [Online]. Available: https://doi.org/10.1109/quatic.2012.35

[187] H. V. Pham, T. T. T. Nguyen, P. Vu, and T. T. Nguyen, "Toward Mining Visual Log of Software," in *arXiv (Cornell University)*, 2016. [Online]. Available: https://doi.org/10.48550/arxiv.1610.08911

[188] L. Ponzanelli, G. Bavota, A. Mocci, R. Oliveto, M. D. Penta, S. Haiduc, B. Russo, and M. Lanza, "Automatic Identification and Classification of Software Development Video Tutorial Fragments," *IEEE Transactions on Software Engineering*, 2019. [Online]. Available: https://doi.org/10.1109/tse.2017.2779479

[189] L. M. Powell and H. Wimmer, "Evaluating the Effectiveness of Self-Created Student Screencasts as a Tool to Increase Student Learning Outcomes in a Hands-On Computer Programming Course," in *Information Systems Education Journal 13(5)*, 2015. [Online]. Available: https://www.isedj.org/2015-13/n5/ISEDJv13n5p106.html

[190] V. Ragazou and I. Karasavvidis, "THE CONTRIBUTION OF PRACTICE PRINCIPLE IN COMPLEX SOFTWARE TRAINING WITH VIDEO TUTORIALS: AN EXPERIMENTAL STUDY WITH NOVICES," in *INTED proceedings*, 2022. [Online]. Available: https://doi.org/10.21125/inted.2022.0977

[191] ——, "Effects of Signaling and Practice Types in Video-Based Software Training," *Education Sciences*, 2023. [Online]. Available: https://doi.org/10.3390/educsci13060602

[192] K. Rechert, D. Espenschied, and R. Gieschke, "Collecting Actionable Operational Knowledge for Software Preservation from User-generated Screencasts," in *Proceedings of the 24th ACM/IEEE Joint Conference on Digital Libraries*, 2024. [Online]. Available: https://doi.org/10.1145/3677389.3702520

[193] S. Sabab, A. A. Khan, P. K. Chilana, J. McGrenere, and A. Bunt, "An Automated Approach to Assessing an Application Tutorial's Difficulty," in *IEEE Symposium on Visual Languages / Human-Centric Computing Languages and Environments*, 2020. [Online]. Available: https://doi.org/10.1109/vl/hcc50065.2020.9127271

[194] H. Sakinah, G. Y. Saputra, and A. Prayogo, "The Effect of YouTube Learning Videos on Student Learning Outcomes in Block Programming Subjects," *Jurnal Inovasi Pendidikan dan Sains*, 2025. [Online]. Available: https://doi.org/10.51673/jips.v6i3.2690

[195] A. C. M. Santos, M. C. Júnior, and E. de Carvalho Andrade, "Multimedia resources as a support for requirements engineering and software maintenance," *Journal of Software: Evolution and Process*, 2020. [Online]. Available: https://doi.org/10.1002/smr.2327

[196] J. Schenk, L. Prechelt, and S. Salinger, "Distributed-pair programming can work well and is not just distributed pair-programming," in *Companion Proceedings of the 36th International Conference on Software Engineering*, 2014. [Online]. Available: https://doi.org/10.1145/2591062.2591188

[197] M. Schmedes, M. Ahrens, L. Nagel, and K. Schneider, "Enriching Vision Videos with Text: An Eye Tracking Study," in *2022 IEEE 30th International Requirements Engineering Conference (RE)*, 2022. [Online]. Available: https://doi.org/10.1109/re54965.2022.00014

[198] K. Schneider and L. M. Bertolli, "Video Variants for CrowdRE: How to Create Linear Videos, Vision Videos, and Interactive Videos," in *2019 IEEE 27th International Requirements Engineering Conference Workshops (REW)*, 2019. [Online]. Available: https://doi.org/10.1109/rew.2019.00039

[199] M. Sihag, Z. S. Li, A. Dash, N. N. Arony, K. Devathasan, N. Ernst, A. B. Albu, and D. Damian, "A Data-Driven Approach for Finding Requirements Relevant Feedback from TikTok and YouTube," in *2023 IEEE 31st International Requirements Engineering Conference (RE)*, 2023. [Online]. Available: https://doi.org/10.1109/re57278.2023.00020

[200] Y. Soepriyanto, M. Banurohman, and S. Sulthoni, "The Effectiveness of Screencast for Understanding Computer Command Interfaces," *Advances in Social Science, Education and Humanities Research/Advances in social science, education and humanities research*, 2021. [Online]. Available: https://doi.org/10.2991/assehr.k.211210.038

[201] W. A. Sugar, A. Brown, and K. J. Luterbach, "Examining the anatomy of a screencast: Uncovering common elements and instructional strategies," *The International Review of Research in Open and Distributed Learning*, 2010. [Online]. Available: https://doi.org/10.19173/irrodl.v11i3.851

[202] M. R. Taesiri, F. Macklon, and C. Bezemer, "CLIP meets GamePhysics: towards bug identification in gameplay videos using zero-shot transfer learning," in *International Conference on Mining Software Repositories (MSR)*, 2022. [Online]. Available: https://doi.org/10.1145/3524842.3528438

[203] M. R. Taesiri, F. Macklon, S. Habchi, and C. Bezemer, "Searching Bug Instances in Gameplay Video Repositories," *IEEE Transactions on Games*, 2024. [Online]. Available: https://doi.org/10.1109/tg.2024.3355285

[204] A. J. Tayeb and S. Haiduc, "Intelligent Semantic Matching (ISM) for Video Tutorial Search using Transformer Models," in *International Conference on Mining Software Repositories (MSR)*, 2025. [Online]. Available: https://doi.org/10.1109/msr66628.2025.00108

[205] A. Tayeb, S. Haiduc, and M. D. Alahmadi, "VidTutorAssistant: Automating Responses to Programming Tutorial Questions," Proceedings of the 41st IEEE/ACM International Conference on Automated Software Engineering (ASE '26), pp. 3382-3386, 2026. [Online]. Available: https://doi.org/10.1145/3832783.3834599

[206] A. J. Tayeb, M. D. Alahmadi, A. A. Alsulami, B. Alturki, R. Alakhtar, and H. Alsobhi, "VidToCode: LLM-Based Source Code Reconstruction From Programming Video Tutorial Transcripts," *IEEE Access*, 2026. [Online]. Available: https://doi.org/10.1109/access.2026.3741380

[207] C.-Y. Tsai, Y.-F. Yang, and C.-K. Chang, "Cognitive Load Comparison of Traditional and Distributed Pair Programming on Visual Programming Language," in *2015 International Conference of Educational Innovation through Technology (EITT)*, 2015. [Online]. Available: https://doi.org/10.1109/eitt.2015.37

[208] P. Vistisen and S. Poulsen, "Return of the Vision Video: Can Corporate Vision Videos Serve as Setting for Participation?" in *Nordic Design Research Conference*, 2017. [Online]. Available: https://doi.org/10.21606/nordes.2017.026

[209] T. Voerman, "The effect of timing of practice in video-based software training," *University of Twente (MSc thesis, Educational Science and Technology)*, 2020. [Online]. Available: https://purl.utwente.nl/essays/80748

[210] R. Xie, Z. Gao, C. Shi, Z. Shang, L. Chen, and Q. Li, "GUIDE: Resolving Domain Bias in GUI Agents through Real-Time Web Video Retrieval and Plug-and-Play Annotation," arXiv, 2026. [Online]. Available: https://doi.org/10.48550/arxiv.2603.26266

[211] Y. D. Xie, J. Mu, X. Ma, C. Zhang, L. Wang, M. Zhou, M. Liu, S. Qin, Q. Lin, S. Rajmohan, S. Han, and D. Zhang, "From Task to Tutorial: An Automated GUI Framework for Excel Tutorial Document and Video Creation," ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE), 2026. [Online]. Available: https://doi.org/10.1145/3803437.3805216

[212] W. Xiong, S. Gu, B. Ye, Z. Yue, L. Li, F. Song, S. Li, and H. Tian, "Video2GUI: Synthesizing Large-Scale Interaction Trajectories for Generalized GUI Agent Pretraining," arXiv, 2026. [Online]. Available: http://arxiv.org/abs/2605.14747v1

[213] H. Xu, O. Creighton, N. Boulila, and B. Bruegge, "From pixels to bytes," in *Proceedings of the ACM SIGSOFT 20th International Symposium on the Foundations of Software Engineering*, 2012. [Online]. Available: https://doi.org/10.1145/2393596.2393631

[214] Z. Xu, J. Li, D. Tan, M. Cole, I. Ashraf, S. Puri, M. Sabetzadeh, and S. Nejati, "Question Answering for Diagram-Rich Technical Meeting Videos," in *2026 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, 2026. [Online]. Available: https://doi.org/10.1109/icsme67971.2026.00093

[215] Y. Yan, "On Improving Management of Duplicate Video-Based Bug Reports," in *ACM/IEEE International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*, 2024. [Online]. Available: https://doi.org/10.1145/3639478.3639786

[216] L. Yan, A. Head, K. Milne, V. Le, S. Gulwani, C. Parnin, and E. R. Murphy-Hill, "The Invisible Mentor: Inferring User Actions from Screen Recordings to Recommend Better Workflows," CHI Conference on Human Factors in Computing Systems, 2026. [Online]. Available: https://doi.org/10.1145/3772318.3790294

[217] C. Yang, F. Thung, and D. Lo, "Efficient Search of Live-Coding Screencasts from Online Videos," in *2022 IEEE International Conference on Software Analysis, Evolution*

and Reengineering (SANER), 2022. [Online]. Available: https://doi.org/10.1109/saner53432.2022.00021

[218] S. Yang, J. Vermeulen, G. W. Fitzmaurice, and J. Matejka, "AQuA: Automated Question-Answering in Software Tutorial Videos with Visual Anchors," in *CHI Conference on Human Factors in Computing Systems*, 2024. [Online]. Available: https://doi.org/10.1145/3613904.3642752

[219] D. Zhang, X. Peng, B. Yalvaç, D. Eseryel, U. Nadeem, A. Islam, and D. Arceneaux, "Exploring the Impact of Peer-Generated Screencast Tutorials on Computer-Aided Design Education," in *ASEE Annual Conference & Exposition*, 2015. [Online]. Available: https://doi.org/10.18260/p.24074

[220] B. Zhang, Z. Shang, Z. Gao, W. Zhang, R. Xie, X. Ma, T. Yuan, X. Wu, S.-C. Zhu, and Q. Li, "TongUI: Internet-Scale Trajectories from Multimodal Web Tutorials for Generalized GUI Agents," in *arXiv*, 2025. [Online]. Available: http://arxiv.org/abs/2504.12679v4

[221] Y. Zhang, X. Guo, Y. Goh, J. Hu, Z. Chen, X. Wang, D. Gao, and M. Z. Shou, "ShowUI-Aloha: Human-Taught GUI Agent," arXiv, 2026. [Online]. Available: http://arxiv.org/abs/2601.07181v1

[222] D. Zhao, Z. Xing, C. Chen, X. Xia, and G. Li, "ActionNet: Vision-Based Workflow Action Recognition From Programming Screencasts," in *ACM/IEEE International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*, 2019. [Online]. Available: https://doi.org/10.1109/icse.2019.00049

[223] W. Zhao, S. Kim, N. Xu, and H. Jin, "Video Question Answering on Screencast Tutorials," International Joint Conference on Artificial Intelligence, 2020. [Online]. Available: https://doi.org/10.24963/ijcai.2020/148

[224] Y. Zhao, S. Talebipour, K. Baral, H. Park, L. Yee, S. A. Khan, Y. Brun, N. Medvidović, and K. Moran, "Avgust: automating usage-based test generation from videos of app executions," in *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 2022. [Online]. Available: https://doi.org/10.1145/3540250.3549134

[225] D. Zhao, Z. Xing, X. Xia, D. Ye, X. Xu, and L. Zhu, "SeeHow: Workflow Extraction from Programming Screencasts through Action-Aware Video Analytics," in *2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE), pp. 1946-1957*, 2023. [Online]. Available: https://doi.org/10.1109/icse48619.2023.00165

[226] D. Zhao, Z. Xing, Q. Lu, X. Xu, and L. Zhu, "SeeAction: Towards Reverse Engineering How-What-Where of HCI Actions from Screencasts for UI Automation," in *ACM/IEEE International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*, 2025. [Online]. Available: https://doi.org/10.1109/icse55347.2025.00144

[227] F. Zieris and L. Prechelt, "PP-ind: Description of a Repository of Industrial Pair Programming Research Data," arXiv preprint, 2020. [Online]. Available: https://doi.org/10.48550/arxiv.2002.03121

Table 6. Communication role of the recording by software engineering activity (number of studies).

Role	Studies	Bug reporting	Bug reproduction	UI design feedback	Code review	Documentation	Tutorial learning	Onboarding training	Requirements	Remote collaboration	Knowledge transfer	Program comprehension	Testing	Other
Reporting	25	16	9	3	-	-	1	-	1	1	-	-	11	1
Demonstration	71	5	5	-	-	11	50	18	7	1	6	4	8	1
Instruction	102	-	-	-	-	27	98	30	1	-	10	3	-	1
Explanation	26	3	1	2	1	8	12	-	5	2	9	1	1	-
Verification	11	4	2	1	-	-	-	-	5	-	-	1	8	-
Documentation	57	2	2	-	-	31	25	-	21	2	9	6	4	1
Collaboration	16	1	-	-	-	-	6	-	6	11	5	-	1	-
Elicitation	22	-	-	1	-	-	-	-	22	4	-	-	-	-
Other	4	-	-	-	-	-	-	-	1	1	-	3	-	3

Cells count distinct studies with both values; a dash means no study in this map. Columns with no study are omitted.

Table 7. Software engineering activity by research method (number of studies).

Activity	Studies	Experiment	Quasi experiment	Survey	Interview	Case study	Mining	Tool evaluation	Literature review	Observational	Mixed	Other*
Bug reporting	19	5	-	1	1	-	4	15	-	-	4	-
Bug reproduction	10	2	-	-	-	-	-	10	-	-	-	-
UI design feedback	4	1	-	-	-	2	-	1	-	-	-	-
Code review	1	-	-	-	-	1	-	-	-	-	-	-
Documentation	46	1	-	1	-	-	-	33	1	1	6	3
Tutorial learning	116	32	11	5	2	4	2	36	-	4	12	8
Onboarding training	30	20	4	3	-	1	1	-	-	-	-	1
Requirements	39	13	-	1	-	9	3	7	4	1	1	-
Remote collaboration	13	4	3	-	-	1	2	1	-	1	2	-
Knowledge transfer	22	1	-	-	2	1	1	7	-	2	4	4
Program comprehension	10	3	-	-	1	-	-	5	-	1	1	1
Testing	17	4	-	-	1	-	1	14	-	-	1	-
Other	5	-	-	-	-	-	-	4	-	1	-	-

Cells count distinct studies with both values; a dash means no study in this map. Columns with no study are omitted.

A Additional tables

B Included studies

Table 16. The included studies, with the codes used in this paper.

ID	Study	Activity	Artifact	Level	Read
A043	Agrawal et al. 2022 [70]	Bug reporting, Remote collaboration	Other	2	Partial text

ID	Study	Activity	Artifact	Level	Read
B033	Alaboudi and LaToza 2019 [71]	Knowledge transfer, Program comprehension	Live stream recording, Screencast	1	Full text
B087	Alaboudi and LaToza 2019 [42]	Remote collaboration, Knowledge transfer	Live stream recording	2	Partial text
B154	Alaboudi and LaToza 2021 [72]	Program comprehension, Other	Live stream recording	2	Full text
B012	Alahmadi 2022 [73]	Documentation, Tutorial learning	Screencast, Tutorial video	2	Abstract
B011	Alahmadi 2023 [74]	Documentation, Tutorial learning	Screencast, Tutorial video	3	Abstract
B061	Alahmadi 2026 [75]	Documentation, Tutorial learning	Tutorial video	3	Abstract
B042	Alahmadi and Alshangiti 2024 [76]	Documentation, Tutorial learning	Screencast, Tutorial video	3	Abstract
B018	Alahmadi et al. 2020 [77]	Documentation, Tutorial learning	Tutorial video, Screencast	3	Abstract
B136	Alahmadi et al. 2020 [78]	Documentation, Tutorial learning	Screencast, Tutorial video	3	Full text
B090	Alahmadi et al. 2024 [79]	Knowledge transfer, Remote collaboration	Live stream recording	2	Abstract
B006	AlGhamdi et al. 2026 [80]	Tutorial learning	Tutorial video	4	Abstract
B037	Aureliano 2014 [81]	Tutorial learning	Screencast	1	Abstract
B074	Bai et al. 2026 [82]	Documentation, Tutorial learning	Screen recording bug report, Demo video, Tutorial video	3	Partial text
B174	Bao et al. 2016 [83]	Program comprehension, Other	Screencast	3	Full text
B002	Bao et al. 2019 [40]	Tutorial learning, Program comprehension, Documentation	Tutorial video, Screencast	4	Full text
B157	Bao et al. 2020 [84]	Documentation, Tutorial learning	Screencast, Tutorial video	3	Full text
C069	Beasley and Johnson 2022 [85]	Remote collaboration, Tutorial learning	Other	3	Partial text
A017	Bernal-Cárdenas et al. 2020 [38]	Bug reproduction, Testing	Screen recording bug report	1	Partial text
A020	Bernal-Cárdenas et al. 2023 [86]	Bug reproduction, Testing	Screen recording bug report	1	Partial text
C006	Broll et al. 2007 [87]	Requirements	Vision video	1	Full text
C097	Bruegge et al. 2008 [88]	Requirements, Tutorial learning	Demo video	1	Abstract
B053	Buffardi and Wang 2022 [89]	Tutorial learning	Tutorial video	3	Abstract
B024	Cheah 2019 [90]	Tutorial learning	Screencast	4	Abstract
B070	Cheah 2021 [91]	Tutorial learning	Screencast	4	Abstract

ID	Study	Activity	Artifact	Level	Read
B052	Cheah and Leong 2019 [92]	Tutorial learning	Screencast	4	Full text
B108	Chen 2020 [93]	Tutorial learning, Onboarding training	Tutorial video	4	Full text
B022	Chen et al. 2021 [94]	Tutorial learning, Knowledge transfer	Live stream recording	2	Partial text
A019	Chen et al. 2022 [95]	Bug reproduction, Tutorial learning	Screen recording bug report, Tutorial video, Other	1	Partial text
A028	Cherkashyn 2026 [96]	Testing, Bug reproduction	Other	1	Abstract
B135	Chi 2016 [97]	Tutorial learning, Documentation	Tutorial video	2	Abstract
A002	Cooper et al. 2021 [4]	Bug reporting, Testing	Screen recording bug report	3	Partial text
A011	Costa et al. 2024 [98]	Bug reporting	Screen recording bug report, Session recording	3	Partial text
C003	Creighton et al. 2006 [2]	Requirements	Vision video	1	Abstract
C055	Drosos and Guo 2021 [99]	Tutorial learning, Knowledge transfer	Live stream recording	2	Abstract
B043	Drosos et al. 2024 [100]	Tutorial learning	Tutorial video	2	Partial text
B088	Dybvik 2008 [101]	Tutorial learning, Onboarding training	Tutorial video	2	Abstract
B050	Désiron et al. 2026 [102]	Tutorial learning, Onboarding training	Tutorial video	4	Partial text
B128	Ellmann et al. 2017 [103]	Documentation, Knowledge transfer	Screencast	2	Full text
B030	Escobar-Avila et al. 2017 [104]	Documentation, Tutorial learning	Tutorial video	2	Abstract
A095	Etemadi 2023 [105]	Bug reporting	Screen recording bug report	1	Partial text
B186	Faas et al. 2018 [106]	Tutorial learning, Knowledge transfer	Live stream recording	2	Abstract
B155	Fan et al. 2026 [107]	Documentation, Knowledge transfer	Tutorial video	2	Abstract
B013	Feijóo-García and Gardner-Mccune 2020 [108]	Tutorial learning, Knowledge transfer	Tutorial video	2	Abstract
A007	Feng and Chen 2022 [65]	Bug reproduction, Bug reporting	Screen recording bug report	4	Partial text
A012	Feng et al. 2023 [39]	Bug reproduction, Bug reporting	Screen recording bug report	4	Partial text
B010	Feng et al. 2023 [109]	Tutorial learning, Documentation	Tutorial video, Screencast	3	Partial text
A030	Feng et al. 2026 [110]	Bug reproduction, Bug reporting	Screen recording bug report	1	Partial text
C002	Fricker et al. 2015 [111]	Requirements, Knowledge transfer	Session recording	2	Abstract
C103	Galster et al. 2018 [112]	Onboarding training, Tutorial learning	Tutorial video	1	Abstract
B1209	Garrett 2020 [45]	Tutorial learning, Onboarding training	Tutorial video	4	Abstract

ID	Study	Activity	Artifact	Level	Read
B161	Ghilay 2015 [62]	Tutorial learning, Onboarding training	Screencast	2	Abstract
B113	Granowski 2025 [113]	Documentation	Screencast	1	Full text
B125	Green et al. 2012 [114]	Tutorial learning	Screencast	3	Abstract
B114	Hu et al. 2023 [115]	Tutorial learning, Documentation	Tutorial video, Screencast	3	Abstract
B145	Huang et al. 2026 [116]	Documentation, Knowledge transfer	Session recording, Screencast	2	Abstract
C058	Høegh 2008 [59]	UI design feedback	Other	3	Abstract
C068	Jang et al. 2026 [117]	Remote collaboration, Tutorial learning	Other	2	Abstract
B112	Jie 2015 [118]	Tutorial learning, Onboarding training	Tutorial video	4	Abstract
C005	Jirotka and Luff 2006 [119]	Requirements	Other	1	Abstract
B051	Kadriu et al. 2020 [120]	Tutorial learning, Knowledge transfer	Tutorial video	2	Abstract
C011	Karras 2018 [121]	Requirements	Vision video	2	Partial text
C012	Karras and Schneider 2018 [41]	Requirements	Vision video	1	Partial text
C014	Karras and Schneider 2020 [122]	Requirements	Vision video	1	Partial text
C007	Karras et al. 2016 [123]	Requirements	Session recording	2	Abstract
C010	Karras et al. 2016 [124]	Requirements	Vision video	1	Partial text
C009	Karras et al. 2017 [125]	Requirements	Demo video	4	Partial text
C086	Karras et al. 2018 [49]	Requirements	Vision video	4	Full text
C035	Karras et al. 2019 [126]	Requirements	Vision video	1	Full text
C013	Karras et al. 2020 [7]	Requirements	Vision video	3	Partial text
C017	Karras et al. 2020 [127]	Requirements	Vision video	2	Full text
C019	Karras et al. 2021 [128]	Requirements	Vision video	2	Partial text
B015	Kefalas and Stamatopoulou 2018 [129]	Tutorial learning	Screencast	2	Abstract
B079	Khandwala and Guo 2018 [130]	Tutorial learning	Screencast	2	Abstract
B072	Kharisma 2020 [131]	Tutorial learning	Screencast	1	Abstract
C008	Kiesling et al. 2016 [132]	Requirements	Session recording	1	Full text
B023	Kokinda and Boyer 2025 [133]	Tutorial learning, Knowledge transfer	Live stream recording	2	Partial text
B147	Kokinda and Rodeghero 2023 [134]	Knowledge transfer, Tutorial learning	Live stream recording	2	Full text

ID	Study	Activity	Artifact	Level	Read
C029	Kopczyńska and Shi 2023 [135]	Requirements	Vision video	2	Abstract
B160	Kueker and Moore 2023 [136]	Tutorial learning	Screencast	4	Abstract
B020	Kumaravel et al. 2019 [137]	Tutorial learning	Tutorial video, Screencast	2	Abstract
C056	Kuramoto et al. 2022 [57]	Requirements, Remote collaboration	Screen recording bug report, Other	3	Full text
A031	Kuramoto et al. 2024 [3]	Bug reporting	Screen recording bug report, Other	3	Partial text
C073	Käfer et al. 2017 [52]	Tutorial learning, Knowledge transfer	Tutorial video	4	Full text
B111	Lamontagne et al. 2021 [138]	Tutorial learning, Onboarding training	Tutorial video	4	Abstract
C071	Law 2017 [139]	Code review, Tutorial learning	Screencast	2	Full text
B027	Lee et al. 2008 [140]	Tutorial learning	Screencast	4	Full text
B082	Li et al. 2016 [141]	Program comprehension, Other	Screencast	2	Full text
B117	Li et al. 2020 [142]	Tutorial learning	Tutorial video	2	Abstract
B019	Li et al. 2024 [143]	Tutorial learning	Tutorial video, Live stream recording	3	Partial text
B078	Li et al. 2025 [144]	Tutorial learning	Tutorial video	1	Full text
A046	Lin et al. 2019 [145]	Bug reporting, Testing	Other	1	Partial text
B031	Lin et al. 2022 [146]	Tutorial learning, Program comprehension	Screencast	3	Abstract
B158	Lin et al. 2024 [147]	Tutorial learning	Tutorial video	2	Partial text
B165	Liu et al. 2023 [148]	Documentation, Tutorial learning	Tutorial video	2	Abstract
A035	Liu et al. 2025 [149]	Testing, Bug reporting	Screen recording bug report, Other	1	Partial text
A040	Liu et al. 2025 [150]	Testing, Bug reporting	Screen recording bug report, Other	1	Partial text
B047	Liu et al. 2025 [46]	Tutorial learning	Tutorial video	4	Partial text
B1211	Lloyd and Robertson 2011 [50]	Tutorial learning	Screencast	4	Abstract
A068	Lu et al. 2025 [151]	Bug reporting, Bug reproduction, Testing	Other	1	Partial text
B008	Lu et al. 2025 [152]	Tutorial learning, Other	Tutorial video, Screen recording bug report	3	Partial text
A075	Lubos et al. 2026 [153]	UI design feedback, Testing	Session recording, Screen recording bug report	2	Partial text
B140	Luo et al. 2026 [154]	Tutorial learning	Tutorial video, Screencast	4	Partial text

ID	Study	Activity	Artifact	Level	Read
C046	MacLeod et al. 2015 [6]	Documentation, Knowledge transfer	Screencast	2	Partial text
C045	MacLeod et al. 2017 [5]	Documentation, Knowledge transfer	Screencast	2	Partial text
B032	Majgaard 2017 [155]	Tutorial learning, Onboarding training	Tutorial video	1	Abstract
B085	Malkadi et al. 2020 [156]	Documentation, Tutorial learning	Screencast, Tutorial video	3	Full text
C031	May et al. 2024 [157]	Requirements, Tutorial learning	Vision video	1	Abstract
B045	McGowan et al. 2023 [158]	Tutorial learning	Screencast	1	Abstract
B091	Meij 2018 [159]	Tutorial learning, Onboarding training	Tutorial video	4	Full text
B041	Meij and Maseland 2021 [160]	Tutorial learning, Onboarding training	Tutorial video	4	Abstract
B084	Meij and Meij 2014 [53]	Tutorial learning, Onboarding training	Tutorial video	4	Abstract
B073	Meij and Meij 2016 [8]	Tutorial learning, Onboarding training	Tutorial video	4	Abstract
B109	Meij and Meij 2016 [161]	Tutorial learning, Onboarding training	Tutorial video	4	Abstract
B068	Meij and Nuketayeva 2023 [162]	Tutorial learning, Onboarding training	Tutorial video	4	Abstract
B069	Meij and Voerman 2024 [163]	Tutorial learning, Onboarding training	Tutorial video	4	Partial text
B034	Meij et al. 2017 [164]	Tutorial learning, Onboarding training	Tutorial video	4	Abstract
B1216	Mestre 2012 [60]	Tutorial learning	Screencast	3	Abstract
B016	Mohorovicic and Tijan 2011 [165]	Tutorial learning	Screencast	1	Abstract
A045	Moslehi et al. 2018 [166]	Program comprehension, Requirements, Bug reporting	Screencast	1	Abstract
B143	Moslehi et al. 2020 [167]	Documentation, Program comprehension	Screencast	3	Partial text
B075	Murphy-Hill 2012 [168]	Tutorial learning, Knowledge transfer	Screencast	1	Abstract
C020	Nagel and Karras 2021 [169]	Requirements	Vision video	4	Partial text
C021	Nagel et al. 2021 [68]	Requirements, Remote collaboration	Vision video, Async video message	4	Partial text
C022	Nagel et al. 2023 [170]	Requirements, Remote collaboration	Vision video	4	Abstract
C023	Nagel et al. 2023 [66]	Requirements, Remote collaboration	Vision video, Async video message	4	Partial text
C026	Nagel et al. 2023 [171]	Requirements	Vision video	4	Partial text

ID	Study	Activity	Artifact	Level	Read
C024	Nagel et al. 2024 [67]	Requirements, Remote collaboration	Vision video, Async video message	4	Partial text
C030	Nagel et al. 2024 [172]	Requirements	Vision video	2	Abstract
B098	Naumann et al. 2024 [173]	Tutorial learning, Documentation	Screencast, Tutorial video	2	Full text
B056	Nong et al. 2019 [174]	Documentation, Tutorial learning	Tutorial video	2	Abstract
C060	Nørgaard and Hornbæk 2009 [58]	UI design feedback	Other	3	Abstract
B151	Obaidi et al. 2025 [63]	Tutorial learning, Documentation	Tutorial video	2	Full text
B092	Ott et al. 2018 [175]	Documentation, Tutorial learning	Tutorial video, Screencast	3	Abstract
B049	Ouh et al. 2022 [176]	Tutorial learning, Documentation	Tutorial video	3	Full text
A094	Paikari 2023 [56]	Bug reporting	Screen recording bug report	3	Partial text
B081	Pal and Iyer 2015 [177]	Tutorial learning, Onboarding training	Screencast	3	Abstract
B097	Pal and Iyer 2015 [178]	Tutorial learning, Onboarding training	Screencast	3	Abstract
B048	Palaigeorgiou and Despotakis 2010 [179]	Tutorial learning	Screencast, Tutorial video	2	Full text
B100	Pankov 2018 [180]	Tutorial learning, Onboarding training	Tutorial video	4	Abstract
B120	Park et al. 2017 [181]	Tutorial learning, Documentation	Screencast	1	Full text
B063	Park et al. 2018 [182]	Tutorial learning	Screencast	3	Abstract
B152	Parra 2023 [183]	Documentation, Knowledge transfer	Tutorial video	1	Abstract
B044	Pejović et al. 2026 [16]	Tutorial learning, Onboarding training	Tutorial video	2	Abstract
B093	Perraud 2024 [184]	Tutorial learning, Documentation	Tutorial video	1	Abstract
B021	Perraud et al. 2024 [185]	Tutorial learning	Tutorial video	3	Partial text
C038	Pham et al. 2012 [186]	Requirements	Demo video	1	Abstract
A022	Pham et al. 2016 [187]	Bug reproduction, Testing, Other	Session recording, Other	1	Partial text
B083	Ponzanelli et al. 2019 [188]	Documentation, Tutorial learning	Tutorial video	3	Abstract
B060	Powell and Wimmer 2015 [189]	Tutorial learning	Screencast	3	Abstract
B096	Ragazou and Karasavvidis 2021 [43]	Tutorial learning, Onboarding training	Tutorial video	4	Full text
B153	Ragazou and Karasavvidis 2021 [44]	Tutorial learning, Onboarding training	Tutorial video	4	Abstract

ID	Study	Activity	Artifact	Level	Read
B029	Ragazou and Karasavvidis 2022 [190]	Tutorial learning, Onboarding training	Tutorial video	3	Abstract
B067	Ragazou and Karasavvidis 2023 [191]	Tutorial learning, Onboarding training	Tutorial video	4	Abstract
B077	Rechert et al. 2024 [192]	Knowledge transfer, Documentation	Screencast	1	Abstract
B089	Sabab et al. 2020 [193]	Tutorial learning	Tutorial video	2	Abstract
B025	Sakinah et al. 2025 [194]	Tutorial learning, Onboarding training	Tutorial video	3	Abstract
C040	Santos et al. 2020 [195]	Requirements, Documentation	Other	2	Abstract
C061	Schenk et al. 2014 [196]	Remote collaboration	Session recording, Other	1	Full text
C025	Schmedes et al. 2022 [197]	Requirements	Vision video	4	Abstract
C018	Schneider and Bertolli 2019 [198]	Requirements	Vision video	1	Abstract
C015	Schneider et al. 2019 [51]	Requirements	Vision video	4	Partial text
A041	Shi et al. 2023 [54]	Testing, Bug reporting, Program comprehension	Demo video, Other	4	Partial text
A096	Shi et al. 2023 [48]	Requirements, Testing	Demo video	4	Abstract
C034	Shi et al. 2024 [47]	Requirements, Testing	Demo video, Screen recording bug report	4	Partial text
C036	Sihag et al. 2023 [199]	Requirements	Other	3	Partial text
B028	Soepriyanto et al. 2021 [200]	Tutorial learning	Screencast	3	Abstract
B133	Sugar et al. 2010 [201]	Tutorial learning, Documentation	Screencast	2	Abstract
A059	Taesiri et al. 2022 [202]	Bug reporting, Testing	Other	1	Partial text
A058	Taesiri et al. 2024 [203]	Bug reporting, Testing	Other	1	Abstract
B123	Tayeb and Haiduc 2025 [204]	Documentation, Tutorial learning	Tutorial video	2	Abstract
B149	Tayeb et al. 2024 [61]	Tutorial learning, Onboarding training	Tutorial video	2	Full text
B004	Tayeb et al. 2026 [205]	Tutorial learning, Documentation	Tutorial video	3	Full text
B005	Tayeb et al. 2026 [206]	Documentation, Tutorial learning	Tutorial video	3	Abstract
C065	Tsai et al. 2015 [207]	Remote collaboration, Tutorial learning	Other	3	Abstract
C037	Vistisen and Poulsen 2017 [208]	Requirements, UI design feedback	Vision video	1	Abstract
B026	Voerman 2020 [209]	Tutorial learning, Onboarding training	Tutorial video	4	Abstract

ID	Study	Activity	Artifact	Level	Read
B007	Worlitz et al. 2016 [64]	Tutorial learning, Onboarding training	Tutorial video	4	Full text
B106	Worlitz et al. 2018 [55]	Tutorial learning, Onboarding training	Tutorial video	3	Full text
B055	Xie et al. 2026 [210]	Tutorial learning	Tutorial video	2	Abstract
B057	Xie et al. 2026 [211]	Documentation, Tutorial learning	Tutorial video	3	Partial text
B009	Xiong et al. 2026 [212]	Tutorial learning	Tutorial video	3	Abstract
C087	Xu et al. 2012 [213]	Requirements	Vision video	1	Abstract
C043	Xu et al. 2026 [214]	Requirements, Knowledge transfer	Other	3	Partial text
A001	Yan 2024 [215]	Bug reporting	Screen recording bug report	1	Abstract
A003	Yan et al. 2024 [37]	Bug reporting	Screen recording bug report	1	Partial text
B107	Yan et al. 2026 [216]	Tutorial learning, Documentation	Screen recording bug report, Screencast	3	Partial text
B066	Yang et al. 2022 [217]	Tutorial learning, Documentation	Screencast	3	Full text
B086	Yang et al. 2024 [218]	Tutorial learning, Documentation	Tutorial video	3	Partial text
B059	Zhang et al. 2015 [219]	Tutorial learning	Screencast	3	Abstract
B159	Zhang et al. 2025 [220]	Tutorial learning	Tutorial video	3	Partial text
B148	Zhang et al. 2026 [221]	Documentation, Knowledge transfer	Screencast, Session recording	2	Abstract
B094	Zhao et al. 2019 [222]	Tutorial learning, Documentation	Screencast	3	Full text
B101	Zhao et al. 2020 [223]	Tutorial learning, Documentation	Tutorial video, Screencast	3	Full text
A088	Zhao et al. 2022 [224]	Testing, Documentation	Demo video, Screen recording bug report	1	Partial text
B038	Zhao et al. 2023 [225]	Tutorial learning, Program comprehension, Documentation	Screencast, Tutorial video, Live stream recording	3	Partial text
A029	Zhao et al. 2025 [226]	Bug reproduction, Testing	Screencast, Screen recording bug report	1	Partial text
C070	Zieris and Prechelt 2020 [227]	Remote collaboration	Session recording, Screencast	1	Full text

Record ids link to the dataset. Level = evidence level (1 no comparison to 4 controlled comparison). Read = how much of the study was read.

Table 8. Software engineering activity by outcome measured (number of studies).

Activity	Studies	Comprehension	Task completion	Reproduction accuracy	Communication time	Clarification effort	Error rate	Learning performance	Cognitive load	Satisfaction	Tool accuracy	Adoption use	Other
Bug reporting	19	1	1	3	8	3	-	-	-	3	13	1	4
Bug reproduction	10	-	-	4	3	-	-	-	-	1	8	-	2
UI design feedback	4	2	-	-	-	-	-	-	-	2	1	1	1
Code review	1	1	-	-	-	-	-	-	-	1	-	-	-
Documentation	46	1	4	-	3	-	-	2	-	13	35	3	9
Tutorial learning	116	4	12	-	7	-	-	45	10	48	38	13	17
Onboarding training	30	-	3	-	1	-	-	24	7	18	-	2	1
Requirements	39	11	2	-	5	1	1	1	2	11	5	3	23
Remote collaboration	13	3	-	-	3	1	-	2	1	2	1	1	10
Knowledge transfer	22	1	-	-	2	-	-	2	-	6	7	7	11
Program comprehension	10	2	2	-	2	-	-	1	-	2	6	-	2
Testing	17	1	3	2	3	-	1	-	-	3	12	1	3
Other	5	-	1	-	-	-	-	-	-	-	3	-	2

Cells count distinct studies with both values; a dash means no study in this map. Columns with no study are omitted.

Table 9. Software engineering activity by evidence level (number of studies).

Activity	Studies	1	2	3	4
Bug reporting	19	11	1	4	3
Bug reproduction	10	8	-	-	2
UI design feedback	4	1	1	2	-
Code review	1	-	1	-	-
Documentation	46	6	16	23	1
Tutorial learning	116	13	31	42	30
Onboarding training	30	2	4	5	19
Requirements	39	15	8	4	12
Remote collaboration	13	2	4	3	4
Knowledge transfer	22	4	16	1	1
Program comprehension	10	2	2	4	2
Testing	17	12	1	1	3
Other	5	1	2	2	-

Cells count distinct studies with both values; a dash means no study in this map. Columns with no study are omitted.

Table 10. Comparators used in the included studies (none = no comparison).

Comparator	Studies	Share of included studies (%)
None	96	48.5
Text	23	11.6
Screenshot	0	0.0
Text and screenshot	4	2.0
Other medium	10	5.1
Tool baseline	49	24.7
Human baseline	16	8.1

A study with several values counts once in each; 198 studies in total.

Table 11. Evidence-gap matrix: software engineering activity by outcome.

Activity	Comprehension	Task completion	Reproduction accuracy	Communication time	Clarification effort	Error rate	Learning performance	Cognitive load	Satisfaction	Tool accuracy	Adoption use	Other
Bug reporting	1 (0/1)	1 (0/1)	3 (1/1)	8 (3/3)	3 (2/0)	-	-	-	3 (1/2)	13 (1/2)	1	4 (2/0)
Bug reproduction	-	-	4 (0/1)	3 (0/2)	-	-	-	-	1 (0/1)	8 (0/2)	-	2
UI design feedback	2 (2/0)	-	-	-	-	-	-	-	2 (1/0)	1	1	1 (1/0)
Code review	1	-	-	-	-	-	-	-	1	-	-	-
Documentation	1 (0/1)	4 (3/0)	-	3 (2/1)	-	-	2 (2/0)	-	13 (7/1)	35 (22/1)	3	9 (1/0)
Tutorial learning	4 (0/2)	12 (6/4)	-	7 (4/3)	-	-	45 (15/25)	10 (3/7)	48 (12/17)	38 (24/2)	13 (3/0)	17 (2/1)
Onboarding training	-	3 (1/2)	-	1 (1/0)	-	-	24 (4/18)	7 (1/6)	18 (1/13)	-	2	1 (0/1)
Requirements	11 (0/9)	2 (0/2)	-	5 (1/4)	1	1 (0/1)	1	2 (0/2)	11 (2/4)	5 (2/0)	3	23 (2/7)
Remote collaboration	3 (0/3)	-	-	3 (0/2)	1	-	2 (2/0)	1 (1/0)	2 (1/1)	1	1	10 (1/4)
Knowledge transfer	1	-	-	2 (1/1)	-	-	2 (0/1)	-	6 (1/1)	7 (1/0)	7	11
Program comprehension	2 (0/2)	2 (1/1)	-	2 (0/2)	-	-	1 (1/0)	-	2 (0/2)	6 (3/1)	-	2
Testing	1 (0/1)	3 (0/3)	2	3 (1/1)	-	1 (0/1)	-	-	3 (0/2)	12 (1/0)	1	3 (0/1)
Other	-	1 (1/0)	-	-	-	-	-	-	-	3 (2/0)	-	2

Each cell gives the number of studies; in parentheses, the number with a non-controlled comparison (level 3) and with a controlled comparison (level 4). A dash means no study identified in this map, which is not evidence that none exist. Columns with no study are omitted.

Table 12. Evidence-gap matrix: software engineering activity by artifact type.

Activity	Screencast	Screen recording bug report	Vision video	Demo video	Tutorial video	Live stream recording	Async video message	Session recording	Other
Bug reporting	1	12 (4/2)	-	1 (0/1)	-	-	-	1 (1/0)	9 (1/1)
Bug reproduction	1	7 (0/2)	-	-	1	-	-	1	4
UI design feedback	-	1	1	-	-	-	-	1	2 (2/0)
Code review	1	-	-	-	-	-	-	-	-
Documentation	27 (15/1)	3 (2/0)	-	2 (1/0)	31 (19/1)	1 (1/0)	-	2	1
Tutorial learning	46 (23/8)	4 (3/0)	1	2 (1/0)	77 (28/24)	7 (2/0)	-	-	4 (2/0)
Onboarding training	3 (2/0)	-	-	-	27 (3/19)	-	-	-	-
Requirements	1	2 (1/1)	25 (1/9)	5 (0/3)	-	-	3 (0/3)	3	5 (3/0)
Remote collaboration	1	1 (1/0)	4 (0/4)	-	-	2	3 (0/3)	2	6 (3/0)
Knowledge transfer	8	-	-	-	5 (0/1)	8	-	3	1 (1/0)
Program comprehension	8 (4/1)	-	-	1 (0/1)	2 (1/1)	3 (1/0)	-	-	1 (0/1)
Testing	1	9 (1/1)	-	4 (0/3)	-	-	-	2	9 (0/1)
Other	2 (1/0)	1 (1/0)	-	-	1 (1/0)	1	-	1	1

Each cell gives the number of studies; in parentheses, the number with a non-controlled comparison (level 3) and with a controlled comparison (level 4). A dash means no study identified in this map, which is not evidence that none exist. Columns with no study are omitted.

Table 13. Studies with a comparator: direction of the reported result by evidence level.

Evidence level	Favors video	Mixed	No difference	Favors alternative	Not applicable	Total
1: no comparison	0	0	0	0	17	17
2: qualitative or descriptive	2	0	0	1	17	20
3: observational or non-controlled comparison	5	5	2	1	29	42
4: controlled comparison	10	7	3	0	3	23

Counts of studies by reported direction. They are not effect sizes and are not pooled; studies differ in design, population and outcome.

Table 14. Who produces and who consumes the recordings in the included studies.

Role	Group	Studies	Share of included studies (%)
Producer	Developer	78	39.4
Producer	Tester	13	6.6
Producer	End user	47	23.7
Producer	Student	11	5.6
Producer	Designer	4	2.0
Producer	Researcher	51	25.8
Producer	Tool or agent	8	4.0
Producer	Other	2	1.0
Consumer	Developer	64	32.3
Consumer	Tester	0	0.0
Consumer	End user	33	16.7
Consumer	Student	76	38.4
Consumer	Designer	0	0.0
Consumer	Researcher	6	3.0
Consumer	Tool or agent	41	20.7
Consumer	Other	0	0.0

A study can name several groups. 198 studies in total.

Table 15. Included studies by reported size of the data basis (participants or items; parsed from the data basis text, see the data files).

Reported n	Studies	Share of included studies (%)
1-9	10	5.1
10-29	28	14.1
30-99	32	16.2
100-999	22	11.1
1000 or more	13	6.6
not reported	25	12.6
unparsed	68	34.3

198 studies in total.