University of Glasgow | School of Computing Science

Uncovering the Cognitive Biases of LLMs in Software Effort Estimation

Taylor Scott
2670341S

Abstract

Software effort estimation is a critical yet error-prone activity, in part due to the susceptibility of human estimators to cognitive biases including anchoring, over-optimism, and over-confidence. As large language models are increasingly considered for use in estimation workflows, whether they inherit these same behavioural tendencies is an open question this paper addresses directly. We present the first systematic replication of a broad family of established human software effort estimation bias experiments, using five frontier large language models as synthetic participants. We replicated sixteen experiments drawn from seven source papers, with requirements documents constructed from real open-source projects in the TAWOS dataset.

The results show that LLMs recreate many of the cognitive biases documented in the human literature, most robustly and consistently for anchoring, where all five models reproduce the directional patterns seen in human estimators. Over-confidence in prediction interval calibration is present at levels comparable to human professionals, and group discussion processes shift estimates upward in a direction consistent with the human literature. Two findings diverge from the human baseline: LLM anchoring is influenced by source credibility where humans in the original study were unaffected by it, and the counter-intuitive risk identification paradox documented for human estimators is absent across four experiments and all five models. Results vary meaningfully between models.

These findings surface both risks and opportunities for deploying LLMs in estimation practice. Anchoring represents a concrete risk when models are exposed to prior estimates or framed requests, while techniques such as external-perspective framing and extreme-limits elicitation consistently widen prediction intervals across all tested models. We make the replication package, including all experimental materials, data, and analysis scripts, available at: https://github.com/tjqscott/LLM-SEE.

1 Introduction

Software effort estimation (SEE) is a long-standing and central activity within software engineering, supporting planning, resource allocation, and general project control [6]. Persistent underestimation contributes to budget overruns, schedule slippage, and in some cases the collapse of projects altogether [21]. Although many algorithmic and model-based techniques exist, everyday practice still depends heavily on human judgement [2]. Human judgment is in turn prone to cognitive biases [19, 38], systematic deviations from rational reasoning that lead to predictable errors. In SEE, this means estimates can be skewed by irrelevant numerical cues, by undue optimism about task difficulty, or by unwarranted confidence in the precision of predictions. Research over several decades has demonstrated that human estimators are affected by anchoring [3], over-optimism [9], over-confidence [18], and related biases, as empirical studies have consistently shown [24, 26]. This study focuses on these three categories specifically, as they are the most extensively and consistently documented in the human SEE literature, providing the strongest basis for comparison with LLM behaviour.

Large Language Models (LLMs) are increasingly capable of generating structured reasoning and plausible effort estimates without task-specific training [28], and are already appearing in estimation workflows. Whether they inherit the same biases that have long affected human estimators is an open question with direct consequences for deployment. If LLMs prove less susceptible, they could support more accurate SEE and contribute to more reliable software projects. If they are comparably biased, efficiency gains remain available because LLMs can produce large numbers of estimates in far shorter time than human practitioners [10]. Either outcome matters for how LLMs are integrated alongside human estimators, or in narrower cases deployed as estimators in their own right.

Machine Psychology is a methodological framework that treats LLMs not as software objects to be debugged, but as participants in psychological experiments originally designed for humans [12]. By subjecting models to controlled experimental paradigms, researchers aim to identify human-like reasoning faults and characterise the behavioural tendencies of these systems [4, 11]. Studies within this framework have shown that LLMs can mirror human cognitive patterns, including the tendency to rely on heuristics and exhibit systematic irrationality [8]. Understanding where LLMs and humans share cognitive biases, and where they diverge, is important for designing estimation workflows that genuinely support software development practice rather than replicate human errors at scale [12]. Within SEE, the Machine Psychology literature is small: classic SEE experiments manipulating anchors, credibility cues, and confidence interval elicitation are well documented in the human literature [24, 26], yet only a handful of studies have applied these paradigms to LLMs, and most explore a single bias or a narrow experimental structure [8, 36].

This study responds to that gap with a systematic replication across a broad family of established human SEE bias experiments, using LLM participants in place of human subjects. We preserved the design of the original experiments so that the models' behaviour is comparable to the behaviour of human participants in the original studies. Requirements documents were constructed using real issues drawn from the TAWOS dataset [37], structured in the same format as the materials used in the original experiments, grounding the replications in realistic software tasks.

The experiments covered in this study span three bias categories (anchoring, over-optimism, and over-confidence) drawn from sixteen experiments across seven source papers. Section 2 reviews the relevant literature on SEE, cognitive biases, and Machine Psychology. Section 3 describes the experiment selection process, requirements document construction, model configuration, and statistical framework. Section 4 presents results organised by bias category, followed by a cross-cutting discussion. Section 5 draws out implications for SE researchers, practitioners, and educators, and Section 6 addresses threats to validity.

2 Background and Related Work

2.1 Software Effort Estimation

Software Effort Estimation (SEE) is the process of predicting the most realistic effort required to develop a software system based on input that is inherently incomplete, uncertain, and noisy [6]. This effort is typically expressed in units such as person-hours, representing the time and human resources necessary to complete a specific task or project. Producing realistic estimates is critical for successful project management, as inaccurate predictions lead to budget overruns, missed deadlines, and in extreme cases project failure [26].

2.1.1 Traditional Estimation Models. The methodologies for conducting estimation generally fall into two broad categories. Algorithmic approaches, such as Boehm's COCOMO [6], rely on mathematical formulae and historical data to derive estimates, and remain among the most widely recognised formal methods in the field [2]. Despite the existence of these models, the dominant strategy in industry remains expert-based estimation, including planning poker and direct expert judgement [2]. These approaches rely on the intuition and experience of practitioners to assess the complexity and scope of tasks.

2.1.2 AI-Based Estimation. More recently, LLMs have introduced new possibilities for estimation workflows. LLMs can generate plausible effort estimates from natural language requirements without any task-specific training [28], enabling human-AI collaboration in the estimation process [7]. Whether LLMs offer a genuine improvement over expert judgement, or simply replicate human tendencies including their cognitive limitations, remains an open question [28].

2.1.3 Limitations and Opportunities. Methodological constraints limit the existing body of SEE research, particularly regarding sample size and experimental power. Many studies are constrained by the difficulty of recruiting large numbers of qualified professional participants, often resulting in small or uneven groups [24]. Controlling for learning effects requires between-subjects designs, further increasing the recruitment burden and limiting statistical power [34].

The primary motivation for this study is to investigate whether and to what extent LLMs manifest cognitive biases similar to humans, and to understand where the two diverge. This matters for understanding how LLMs can most effectively support human developers during SEE. A secondary motivation is that LLMs offer a practical solution to the recruitment bottleneck: unlike human subjects, LLMs can be instantiated in large numbers, allowing experiments to be run at scale under controlled conditions to obtain statistically significant results [10]. Beyond scale, LLMs can serve as a low-cost pilot mechanism, allowing researchers to pre-screen experimental designs for promising effects before committing the resources required for human trials. The generalizability of current SEE results is also frequently limited by domain specificity and the use of student proxies [24]. LLM-based replication enables experiments across diverse project domains under consistent conditions, strengthening external validity.

2.2 Cognitive Biases

Cognitive biases are systematic deviations from rational reasoning, functioning as mental shortcuts that lead to predictable errors in judgement [38] [19]. Three categories of bias are particularly prevalent in software effort estimation and form the focus of this study.

Anchoring bias describes the tendency to rely too heavily on the first piece of numerical information encountered, regardless of its relevance [38]. For example, simply changing a request from "How much effort is required?" to "How much can be completed in Y hours?" acts as an anchor that systematically shifts effort estimates [15]. Over-optimism bias describes the tendency to underestimate the effort required for a task, often driven by overconfidence in one's own abilities; studies have shown that estimates for whole tasks can be substantially more optimistic than the sum of estimates for their constituent subtasks [9]. Over-confidence bias is distinct from over-optimism: where over-optimism concerns the point estimate being too low, over-confidence concerns the uncertainty around that estimate being too narrow, manifesting as prediction intervals that fail to account for the true range of possible outcomes [18]. The two can and do co-occur, but they are separable phenomena and are treated as such in this study.

2.2.1 Cognitive Biases in SEE. In the context of SEE, these biases result in inaccurate and inconsistent predictions, as practitioners' judgements can be swayed by irrelevant information or changes in how a request is framed. Unlike random errors, cognitive biases produce consistent, directional patterns of error [35]. Their presence imposes real costs on software projects, primarily through budget overruns and schedule delays [26].

Researchers have proposed various de-biasing techniques. Planning Poker has been shown to reduce anchoring effects compared to unstructured group consensus [13], as the simultaneous revealing of estimates prevents early speakers from anchoring the group. Other studied mitigations include directed questioning, task decomposition, and structured group discussion, which has been found to yield less optimistic estimates than individual expert judgement [16].

2.3 The TAWOS Dataset

Replication of SEE bias experiments with LLM participants requires realistic requirements documents that can serve as estimation inputs. Historically, research in this domain has been constrained by the use of limited, proprietary, or purely judgement-based datasets [16]. Early studies typically relied on expert estimates as their baseline, meaning researchers could assess only the consistency of estimates rather than their accuracy against actual outcomes [23].

The TAWOS dataset addresses this limitation [37]. TAWOS is a versatile repository comprising a large number of issues mined from 44 large open-source projects using Jira. Unlike previous datasets that often relied solely on story points as a relative measure of effort, TAWOS records the total time logged by developers against each issue, providing a measure of actual work hours. This grounds requirements documents in real project data rather than synthetic or proprietary inputs. For agile-format issue documents in particular, it enables objective evaluation of LLM estimates against actual historical effort, rather than against subjective expert baselines. The dataset covers a diverse range of projects across different domains, and its scale overcomes the limitations of the single-project case studies common in earlier literature. Although it is restricted to large projects that use Jira, its breadth makes it a robust foundation for this kind of empirical work [37].

2.4 Machine Psychology

As LLMs have grown in capability and complexity, their internal behaviour has become difficult to interpret fully. Machine Psychology [11] addresses this by treating LLMs as participants in psychological experiments originally designed for humans, shifting the focus from performance metrics to the investigation of emergent cognitive behaviour. Studies in this field have demonstrated that human-like reasoning biases have emerged in large language models, including the tendency to rely on heuristics rather than strict logic [12] [4], suggesting that the reasoning performed by these models shares characteristics with human cognition, including its potential for systematic irrationality [12].

Current Machine Psychology literature faces limitations in scope. Studies frequently focus on a narrow set of dominant model families [12] and on abstract cognitive tasks not tied to expert domains [4]. The application of this methodology to software engineering, and to SEE in particular, remains an emerging field. The only existing Machine Psychology study in SEE is the work of Çalıklı et al. (2025) [8], which confirmed the presence of anchoring bias in LLMs by replicating the experimental design of Jørgensen and Halkjelsvik [17]. While foundational, a single study exploring a single bias is insufficient to characterise how LLMs behave across the full range of biases documented in the human literature, and further replications are needed [36].

3 Methodology

This section describes the complete experimental approach: the systematic process used to identify and select experiments from the literature, the construction of requirements documents, the selection and configuration of LLM participants, the prompt design and execution pipeline, the statistical analysis framework, and the notable deviations from original study designs.

3.1 Experiment Selection

The experiment selection process was conducted in two stages, following the systematic mapping methodology of Mohanani et al [24]. Figure 1 summarises the selection pipeline and the numerical drop-offs at each stage described below.

In the first stage, the 65 primary studies identified by Mohanani et al. were screened for relevance to Software Effort Estimation. Screening proceeded in three passes: title, abstract, and full text where prior steps were ambiguous. We identified sixteen papers as relevant. Notably, all sixteen contained a morphological relative of the word *estimate* in their abstract. The geographic concentration of this literature was striking: eight papers listed Magne Jørgensen as primary author, with his direct collaborator Kjetil Moløkken on a further four. The remaining four were authored by Haugen and Løhre, both of whom also worked within the University of Oslo research group, and Connolly and Aranda, the sole authors from outside this network.

In the second stage, we conducted a five-database literature search to identify relevant papers published after the Mohanani mapping. The search query combined `software` and `cognitive bias` as full-text terms, with `estim*` constrained to the abstract field (using wildcard syntax on databases that supported it, and an explicit morphological list on Science Direct). We downloaded results from ACM Digital Library, IEEE Xplore, Science Direct, Scopus, and Web of Science and uploaded them to Rayyan [27] for deduplication. After removing 19 duplicates, 144 papers remained; a relevance screening for papers concerning cognitive biases in SEE with a human experiment component left 19 papers.

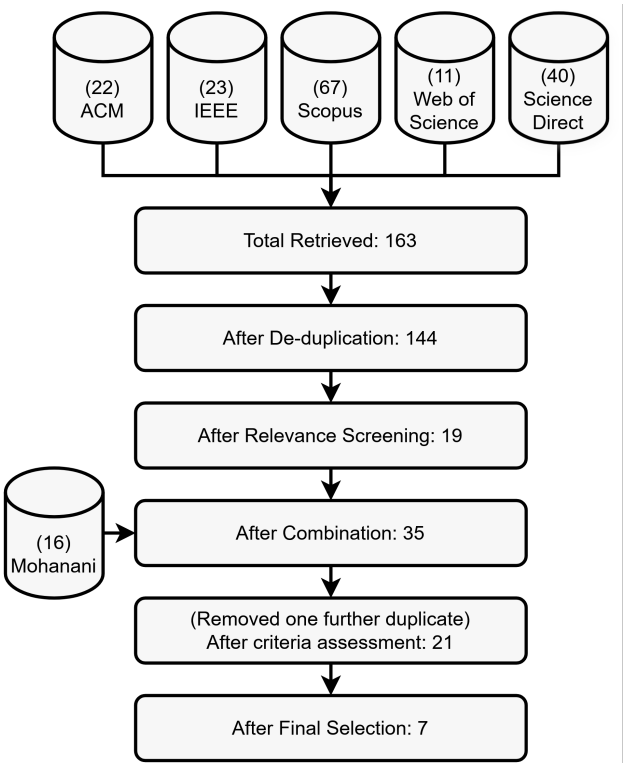

Figure 1: Experiment selection process.

Combined with the 16 primary studies, and after a final check excluding one further duplicate, 21 papers met the full criteria.

We performed data extraction across all 21 papers, documenting the cognitive biases investigated, experimental designs, participant populations, statistical methods, and reported findings. We then evaluated papers against the SIGSOFT Empirical Standards for Experiments [30], with antipatterns noted and essential, desirable, and extraordinary attributes counted. From this analysis, we selected high-quality papers spanning the three most prevalent cognitive bias categories in the SEE literature: Anchoring, Over-optimism, and Over-confidence. These three were prioritised because they are the most extensively documented in human SEE trials, providing the strongest basis for comparison. The full corpus of 21 papers also contains studies on related phenomena including the planning fallacy and optimism bias, sequence effects in estimation, hindsight bias, and wishful thinking; these represent productive directions for future LLM replication work. Papers covering these topics are included in the `experiment-selection` directory of the replication package [33] for reference. Selection further prioritised experimental quality, reproducibility, and suitability for LLM-based replication; all experiments reported within each selected paper were then replicated in full.

The final selection comprises 16 experiments drawn from 7 source papers: Aranda and Easterbrook [3], Løhre and Jørgensen [22], Haugen [13], Connolly and Dean [9], Jørgensen [17], Jørgensen et al [18], and Moløkken and Jørgensen [25]. These cover anchoring direction and magnitude, source credibility effects, consensus-based anchor mitigation, task decomposition effects on overoptimism and overconfidence, group discussion effects, risk identification effects, and confidence interval calibration.

The full data extraction table and selection justification are available in the `experiment-selection` directory of the replication package [33].

3.2 Model Selection and Configuration

We selected five frontier large language models as experimental participants. Selection was guided by the SWE-bench leaderboard [14], taking the top eleven models at the time of selection. From these, four were excluded to avoid redundancy within the same model family, and two were excluded due to prohibitively long inference times that made large-scale execution impractical. The final cohort of five comprises:

- **Google Gemini 3 Flash Preview**
- **Anthropic Claude Opus 4.6**
- **OpenAI GPT-5.2 Codex**
- **DeepSeek V3.2**
- **Moonshot Kimi K2.5**

All models were accessed through the OpenRouter API [1], which provides a unified interface to multiple providers. All models were configured with a temperature of zero, a standard choice for experimental reproducibility that minimises stochastic variation and ensures that observed differences between conditions reflect the experimental manipulation rather than random generation noise. Temperature zero does not guarantee strictly identical outputs across all runs due to implementation differences between providers [5], but in practice outputs are near-identical, making repeated runs within the same condition redundant. We therefore collected a single response per condition per document per model. Any model-specific execution considerations are described in the relevant subsections of Section 3.6.

3.3 Requirements Document Construction

We constructed two categories of requirements documents to serve different experimental purposes.

3.3.1 Narrative Requirements Documents. We constructed 34 narrative requirements documents for the Aranda, Løhre, and Jørgensen experiments. Each document corresponds to a real open-source software project drawn from the TAWOS dataset [37].

We extracted the source data for each document from TAWOS via a SQL query selecting the project name, project description, issue key, issue title, and creation date for the first 50 issues per project ordered chronologically, providing a representative early-development snapshot of each project's scope.

Construction followed a human-in-the-loop process using an LLM with internet access, which we instructed to research the project where appropriate to supplement the extracted issue data with further context. We used the original Aranda study requirements document as a format exemplar, coaching the model to follow its purpose rather than its structure, which initially caused it to mirror section headings too closely. Each generated document was reviewed before use. We dropped three projects at this stage: one that produced a document too short to serve as a credible specification, and two that formed a client-server pair from the same project whose requirements documents overlapped substantially. We retained the server document from each pair and discarded the client.

Each document was produced in two lengths: a full version and a brief version. For the brief versions, the same coaching process was repeated to prevent the model defaulting to a condensed version of the full document's structure. The full versions were piloted across a separate cohort of five models under the control condition to establish a baseline effort estimate per document. The control mean serves two purposes: it provides a baseline for hit-rate analyses in the Jørgensen 2002 family, and it calibrates anchor values for TAWOS-based Aranda conditions. Low anchors are set to approximately 20% of the control mean and high anchors to approximately 200%, rounded to plausible figures, replicating the spirit of the original Aranda design. Using a separate model cohort for piloting reduces the risk of confounding anchor calibration with experimental response.

3.3.2 Agile-Format Issue Documents. We constructed fifty-five agile-format issue documents for the Moløkken and Haugen experiments. Unlike the narrative documents, we produced these through a structured database query rather than an LLM-assisted drafting process.

Issues were extracted from the TAWOS dataset using a SQL query filtering for issues of type `Story` whose description begins with the string `"As a"` (with or without a leading quotation mark), excluding any issue whose description contains a URL, and requiring a non-zero `Resolution_Time_Minutes` value, which was selected over alternative effort fields due to its higher population rate across projects. The resulting records were exported with project identifier, issue key, title, description, and total effort; effort was subsequently converted to work-weeks to align with the Aranda baseline format.

We then grouped issues by project and wrote them to files in batches. We required a minimum of seven issues to form a valid document. Where a project yielded more than seven issues, we wrote batches of ten sequentially before considering the remainder; we wrote a trailing batch of 7–12 issues as a final file for that project, including cases where the entire project yielded between 7 and 12 issues. We discarded trailing batches of fewer than 7. This process produced 55 documents across multiple projects. Spring XD is heavily represented with 37 files, reflecting its large number of tracked releases in TAWOS; this over-representation is an artefact of the dataset and is acknowledged as a potential source of project-level dependence in results from those experiments.

Ground truth completion times for each document are sourced directly from TAWOS as the sum of `Resolution_Time_Minutes` across the issues in that file, converted to work-weeks, providing an objective baseline against which LLM estimates can be evaluated. All requirements documents are available in the `requirements-documents` directory of the replication package [33].

3.4 Prompt Design and Execution

3.4.1 General Structure. We assemble all prompts as Python f-strings at runtime and pass them as a single user-turn message with no system prompt. The general structure consists of: a role-framing preamble instructing the model to act as an experienced software developer or project manager as appropriate to the original study's participant population; any condition manipulation text; the requirements document in a fenced Markdown block; and a forced-format output instruction specifying the exact terminal string the model must reproduce. The output instruction always appears last, following established prompt design practice of placing format constraints at the point of generation.

The output format varies by experiment. Aranda conditions require a terminal line of the form `I ESTIMATE THE PROJECT WILL`

`TAKE <number> MONTHS TO DELIVER.` followed by `CONFIDENCE: <number>%`. Jørgensen 2002 conditions require `MOST_LIKELY: <number>`, `MINIMUM: <number>`, `MAXIMUM: <number>`. Connolly conditions require probability estimates over task completion time, both for the task as a whole and broken down by subtask. Each experimental condition differs from its comparison condition by a single prompt element, with all other text held constant.

3.4.2 Output Parsing. We parsed responses using Python `re.search` on the raw response string, with experiment-specific regular expressions targeting the expected terminal format. A unit-stripping pre-processor runs before the primary regex, converting common time suffixes (`h` for hours, `d` for days, `w` for weeks) and removing `<` prefixes from less-than formatted values. Responses that failed to parse were logged for retrospective recovery; timeout responses were recorded as missing data. Parsing failures were used to iteratively improve extraction utilities, and a success rate above 90% was achieved across all models and conditions, making the residual failure rate acceptable.

All experiments were executed asynchronously using Python's `asyncio` library. An example prompt is shown in Figure 2. All prompts and execution code are available in the `experiments` directory of the replication package [33].

3.5 Statistical Analysis

We implemented statistical analysis in R [29]. Effort estimation data has been consistently found to deviate from normality [26], making parametric tests inappropriate. All comparisons use the paired Wilcoxon signed-rank test, as the same document appears in both conditions being compared across every experiment. For example, the same project brief is estimated under a low anchor and under the control condition. This document-level pairing is preserved in all analyses, making the paired form the correct choice throughout.

The single exception is the Connolly H3a hypothesis, which asks whether the subtask-to-whole-task gap differs from zero rather than comparing two groups. This uses the one-sample Wilcoxon signed-rank test against a reference of zero.

Where an experiment involves multiple comparisons within a single family of hypotheses, we adjusted p-values using Rom's method [31] to control the family-wise error rate. Rom's method is a step-down procedure that maintains strong control over the family-wise error rate while remaining more powerful than comparable correction methods.

We report effect sizes as Cliff's δ throughout, the appropriate non-parametric effect size measurement for Wilcoxon-family comparisons. Cliff's δ ranges from -1 to 1, with conventional thresholds of $|\delta| < 0.147$ (negligible), < 0.33 (small), < 0.474 (medium), and ≥ 0.474 (large) [32]. We present results as box plots for each experiment, enabling direct visual comparison between conditions. All analysis scripts are available in the `statistical--analysis` directory of the replication package [33].

3.6 Deviations from Original Designs

Replicating human experiments with LLM participants requires several adaptations that are universal across this study. While human trials typically allowed participants an hour or more to deliberate, LLM responses are generated in seconds, though executing experiments at scale across five models still required hours of wall-clock time due to API constraints. We replaced human participant recruitment with model instantiation. Most significantly, all experiments use requirements documents derived from the TAWOS dataset rather than the proprietary or otherwise inaccessible materials used in the original studies. TAWOS provides real open-source project data, improving reproducibility and, for agile-format documents, enabling objective evaluation against actual historical effort. Where the original study materials were publicly available, as in the case of Aranda (2005), we used these as format exemplars to guide the construction of TAWOS-based equivalents rather than as direct experimental inputs. Three more substantive deviations from specific original designs are detailed below.

3.6.1 Haugen (2006): Planning Poker. The original Haugen study compared unstructured group estimation (where developers volunteered estimates sequentially, each seeing prior estimates) against planning poker (where estimates are revealed simultaneously, after which the high and low estimators justify their reasoning and a consensus is reached). Participants estimated individual user stories; four development teams participated.

We made two adaptations. First, we grouped user stories by release rather than estimating them individually, meaning each estimation unit contains between 7 and 12 stories, a range guaranteed by the batching logic described in Section 3.3. This grouping was also financially motivated: running the Haugen experiment at the original story-level granularity was projected to cost as much as all other experiments combined, making it practically implausible.

Second, since temperature zero makes repeated calls to the same model with identical prompts return near-identical outputs, a naïve four-developer replication would collapse to a single repeated value. To introduce meaningful variance, we used four distinct developer personas: a senior developer tending toward optimism, a project manager experienced with overruns, a junior developer uncertain about edge cases, and a QA engineer focused on testing and production quality. This introduces qualitatively different perspectives rather than the natural variance found between human team members, and we explicitly treat this as a limitation. We implemented the unstructured process as four sequential sub-waves, each developer's prompt accumulating prior estimates. The planning poker process fired all four persona estimates simultaneously, followed by a justification wave from the high and low estimators, and a consensus wave incorporating the revealed estimates and justifications.

3.6.2 Jørgensen et al. (2002) Study D. Study D of the original paper tested whether removing ego-investment from estimation changes overconfidence, implemented by having participants estimate projects they had no personal involvement in. This manipulation has no natural analogue for an LLM, so in its place we designed a new experiment using an external-consultant framing: the model is asked to provide an independent estimate for a colleague's project, with no personal stake in the outcome. This tests whether perspective framing affects interval width, which is related to but distinct from the original ego-investment manipulation, and we interpret results accordingly.

3.6.3 Jørgensen (2009) Study D. Study D of the original Jørgensen (2009) paper used a conference organisation scenario to test whether the risk identification effect persists outside the software domain, a cross-domain manipulation that does not translate to the TAWOS-based paradigm used here. In its place, we designed a new experiment that holds the domain constant and varies only the relationship framing, using the same external-consultant

Løhre & Jørgensen (2014) — Experiment 3, Low Credibility Condition

You are a software developer asked to estimate the effort required to complete the project below. Assume you will carry out the development work yourself, using the programming language, tools, and database you know best.

An administrative person in your company, with no background in software development, is responsible for logging all work over 10 hours. Without looking at the requirements, they ask whether you think this will take less than 10 work-hours.

```requirements
# Alloy Framework
```

Alloy Framework is an Apache-licensed model–view–controller application framework built on top of Titanium that provides a simple model for separating the application user interface, business logic and data models.

(1) The mobile UI developer interacts with the Alloy Framework via a Node.js command-line compiler script. No graphical layout builder or drag-and-drop UI designer is required for this compiler.
(2) The developer provides a project directory containing declarative XML markup for views, CSS styling files, and CommonJS JavaScript files for business logic.
(3) The compiler must only process files located on the local disk; fetching remote stylesheets or scripts over the network during compilation is out of scope.
(4) The compiler parses the markup and CSS, utilising a Sizzle-based engine to resolve DOM element selections. It automatically evaluates platform-specific attributes (e.g., `data-ti-platform="iPhone"`) and wraps the corresponding generated code in conditional logic, dropping non-matching nodes from the AST to save memory. For this iteration, assume the maximum allowed depth for nested UI widgets is 10 levels.
(5) The compiler merges classes, IDs, and inline styles into unified rules and translates the entire project into standard, imperative Titanium JavaScript API calls.
(6) All generated files are then forcefully written to a designated "generated" folder within the project's Resources directory. Dynamic runtime compilation or evaluation of CSS files on the mobile device itself is out of scope.

```

End your response with exactly:

```
MOST_LIKELY: <number> work-hours
MINIMUM: <number> work-hours
MAXIMUM: <number> work-hours
```

**Figure 2: Example prompt for the Løhre & Jørgensen (2014) Experiment 3 low-credibility condition. The anchor sentence (italicised) is the only element that varies across credibility conditions; all other text is held constant.**

prompt described in Section 3.6.2. This tests whether an external perspective changes the effect of risk identification rather than the original domain-generalisability question. We do not compare results directly to the original Study D human baseline.

## 4 Results & Discussion

This section presents the results of the 16 replicated experiments, organised by bias category: Anchoring (Aranda, Løhre), Over-optimism (Haugen, Moløkken, Jørgensen 2009), and Overconfidence (Connolly, Jørgensen 2002). All statistical comparisons use the paired Wilcoxon signed-rank test, as the same document appears in both conditions across every experiment. The single exception is Connolly H3a, which uses a one-sample Wilcoxon against zero. We applied Rom's correction within each experiment's family of comparisons. We report effect sizes as Cliff's $\delta$. A high-level summary of all experiments and their verdicts is provided in Table 1. Full numerical results and figures are available in the `results` and `figures` directories of the replication package [33].

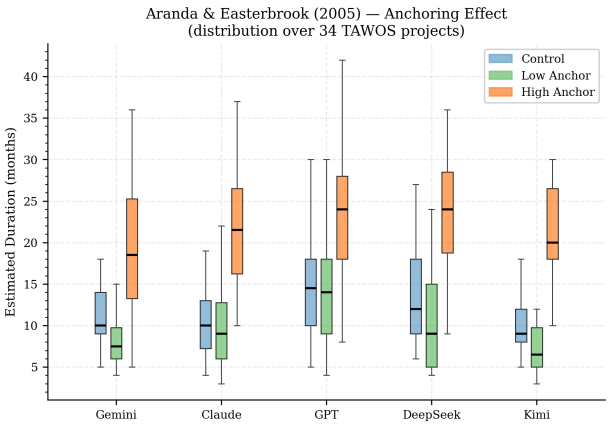


**Figure 3: Aranda & Easterbrook (2005) — anchoring effect across models.**

### 4.1 Anchoring

*4.1.1 Aranda & Easterbrook (2005).* Figure 3 shows a consistent anchoring pattern across all five models. The high-anchor condition produces a large and significant increase in estimates relative to control for all models (Gemini: $\delta = 0.584$; Claude: $\delta = 0.849$; GPT: $\delta = 0.499$; DeepSeek: $\delta = 0.751$; Kimi: $\delta = 0.803$; all $p < 0.001$ after Rom's correction). The low-anchor condition produces a smaller suppressive effect. This effect reaches significance after correction for Gemini ($\delta = -0.343$, medium), GPT ($\delta = -0.141$, negligible), DeepSeek ($\delta = -0.235$, small), and Kimi ($\delta = -0.424$, medium), but not for Claude ($\delta = -0.132$, negligible). This directional asymmetry, a substantially stronger pull toward the high anchor than the low anchor, mirrors the pattern found
```

in the original human study [3], where the high-anchor condition yielded a statistically significant shift but the low-anchor condition did not differ significantly from control.

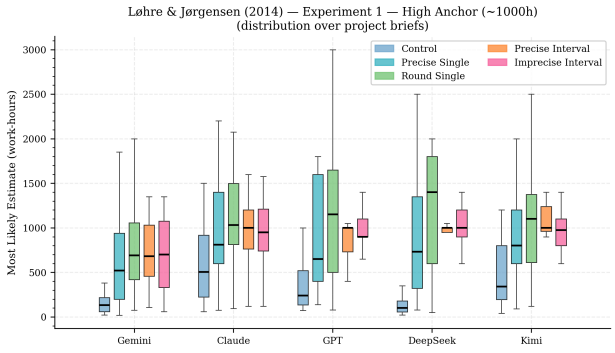

Figure 4: Løhre & Jørgensen (2014) Experiment 1 — high anchor, precision conditions.

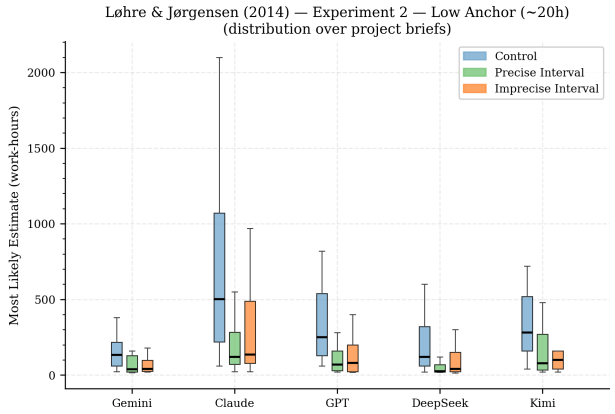

Figure 5: Løhre & Jørgensen (2014) Experiment 2 — low anchor, precision conditions.

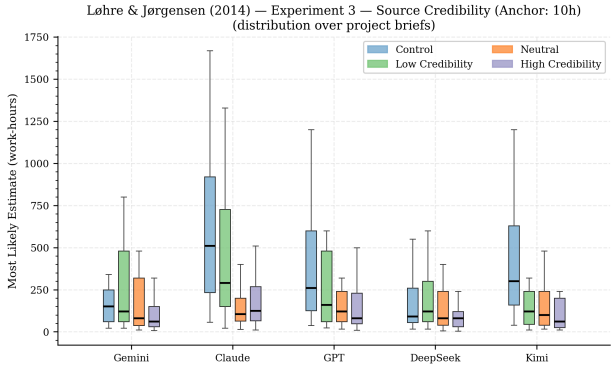

Figure 6: Løhre & Jørgensen (2014) Experiment 3 — source credibility conditions.

4.1.2 Løhre & Jørgensen (2014). Experiment 1 (Figure 4) confirms that a high anchor (~1000h) significantly elevates estimates above control for all models and all anchor formats, with large effect sizes throughout (all $p < 0.001$ after Rom's correction). The round single anchor produces slightly higher estimates than the precise single anchor across all models, a difference that is significant after correction for Gemini, Claude, DeepSeek, and Kimi, though with small effect sizes (δ ranging from -0.133 to -0.274). Precision of interval anchors (precise vs imprecise) has no significant effect for Gemini, Claude, GPT, or Kimi; only DeepSeek shows a significant difference ($\delta = -0.140$, negligible). This suggests that anchor value drives the effect far more strongly than its format.

Experiment 2 (Figure 5) shows that the low anchor (~20h) significantly suppresses estimates below control for all models (all $p < 0.05$ after Rom's correction), with medium effect sizes for Gemini, Kimi ($|\delta| \approx 0.43$–0.47) and large effects for Claude, GPT, and DeepSeek ($|\delta| \approx 0.50$–0.61). The difference between precise and imprecise interval presentations is not significant for any model, consistent with Experiment 1.

Experiment 3 (Figure 6) tests whether source credibility modulates the anchoring effect, using a low (10h) anchor attributed to sources of varying credibility. The original human study found no significant differences between credibility conditions [22]. In the LLM replication, credibility framing matters: low-credibility vs high-credibility comparisons reach significance for Gemini, Claude, GPT, and DeepSeek, with small to medium effect sizes. The gradient is not fully monotonic (the neutral-to-high step is significant only for Kimi), but the consistent low-versus-high pattern across four of five models represents a clear departure from the null human result. The underlying anchoring suppression is also present: Claude and Kimi show large significant suppression relative to control across all credibility conditions, Gemini and Kimi show significant suppression under high credibility specifically, and DeepSeek alone shows no significant suppression in any condition.

4.2 Over-optimism

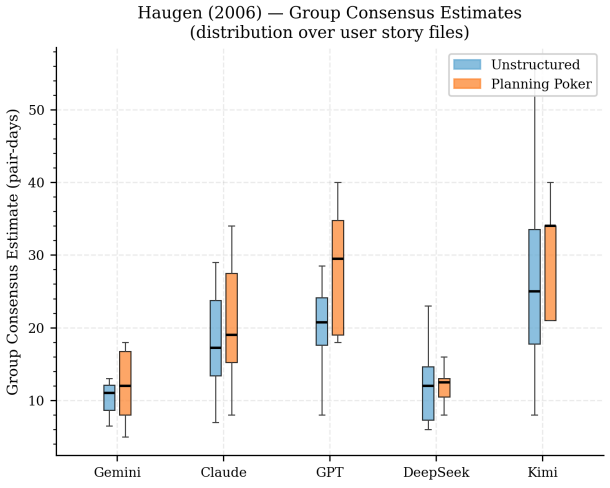

Figure 7: Haugen (2006) — group consensus estimates, unstructured vs planning poker.

4.2.1 Haugen (2006). The Haugen replication addresses whether planning poker produces different estimates to unstructured group discussion, and whether the estimation method affects variance among individual developers [13]. Higher estimates indicate reduced over-optimism, the desirable direction for estimation accuracy.

Figure 7 shows that planning poker tends to produce higher group consensus estimates than unstructured estimation. This difference is only statistically significant for GPT after Rom's correction ($\delta = 0.370$, medium), with all other models showing negligible effects. Within each condition, the group consensus

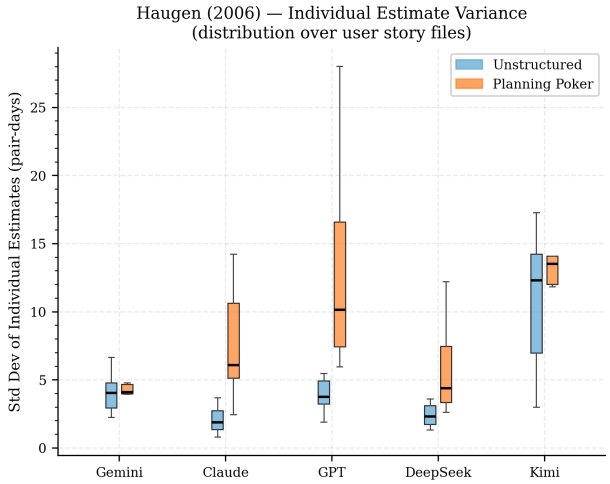

Figure 8: Haugen (2006) — standard deviation of individual estimates, unstructured vs planning poker.

does not meaningfully inflate or deflate relative to the individual mean: where significance is reached after correction, effect sizes are negligible throughout, suggesting the simulated consensus process approximates averaging rather than genuine negotiation.

Figure 8 shows a clearer pattern for individual variance. Planning poker produces significantly higher variance than unstructured estimation for GPT after Rom's correction ($\delta = 0.760$, large). Claude shows a comparably large effect ($\delta = 0.840$) that does not reach significance, reflecting the small paired sample rather than a weaker effect. For Gemini, DeepSeek, and Kimi the difference is negligible. This is consistent with the persona-framing mechanism described in Section 3.6.1: the simultaneous reveal in planning poker exposes the full range of persona perspectives, whereas in unstructured estimation each persona's estimate is anchored by prior estimates, compressing variance.

These results are difficult to compare directly to the original human study due to the substantial deviations in implementation. The results suggest that the simulated group consensus process may approximate averaging rather than genuine negotiation, which represents a fundamental limitation of persona-based LLM group simulation.

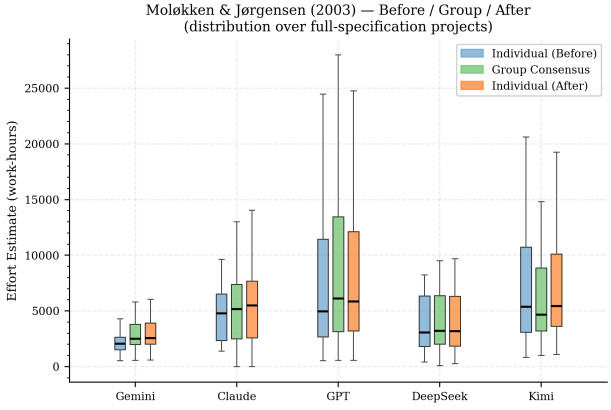

Figure 9: Moløkken & Jørgensen (2003) — individual, group, and post-discussion estimates.

4.2.2 Moløkken & Jørgensen (2003). The Moløkken experiment tests whether group discussion shifts estimates relative to individual estimates [25]. Higher post-discussion estimates indicate that group discussion is doing its job.

Figure 9 and the statistical results show that for Gemini, Claude, GPT, and DeepSeek, the group consensus is significantly higher than the pre-discussion individual mean (all significant after Rom's correction), though effect sizes are small to negligible (δ ranging from 0.038 to 0.294). This indicates that the group process systematically shifts estimates upward in the expected direction, consistent with the original human study. Kimi does not show a significant shift ($p = 0.35$, $\delta = 0.004$).

Post-discussion individual estimates also converge significantly toward the group consensus for Gemini (large effect, $\delta = -0.494$), GPT (medium, $\delta = -0.436$), and DeepSeek (medium, $\delta = -0.410$). Claude shows a smaller but significant convergence ($\delta = -0.253$). Kimi again shows no significant convergence ($\delta = -0.163$, $p = 0.22$). The consistent direction of convergence across the models where it is significant suggests that exposure to the group estimate anchors subsequent individual reasoning.

These findings broadly align with the human study's conclusion that group discussion reduces individual over-optimism, though the effect sizes observed here are smaller than those reported in the original [25].

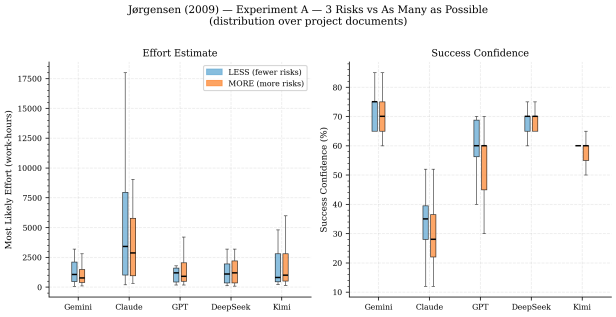

Figure 10: Jørgensen (2009) Experiment A — effort and confidence, fewer vs more risks.

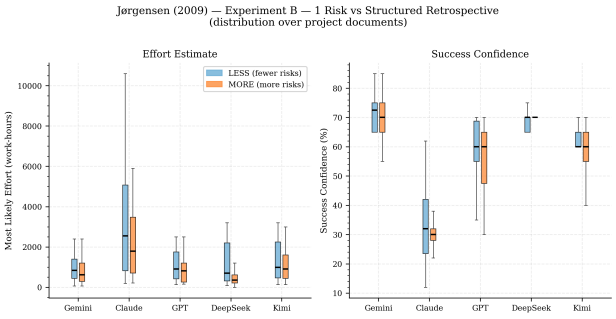

Figure 11: Jørgensen (2009) Experiment B — effort and confidence, fewer vs more risks.

4.2.3 Jørgensen (2009). The Jørgensen (2009) family tests whether more extensive risk identification paradoxically leads to lower effort estimates and higher success confidence, a counter-intuitive human effect attributed to illusion-of-control mechanisms [17]. The manipulation check confirms that the MORE condition successfully elicited more risk content than LESS in all experiments and models (large effects throughout).

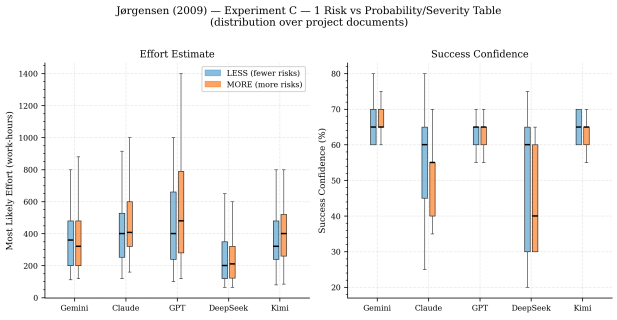

Figure 12: Jørgensen (2009) Experiment C — effort and confidence, fewer vs more risks.

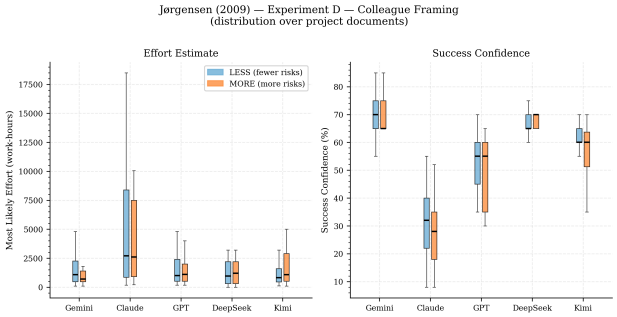

Figure 13: Jørgensen (2009) Experiment D — effort and confidence, colleague framing.

Across all four experiments (Figures 10–13), the results are predominantly null. Effort estimates do not differ significantly between LESS and MORE conditions for most model-experiment combinations after Rom's correction. Where significance is reached, effect sizes are negligible to small and inconsistent in direction across models. Success confidence similarly shows no consistent pattern. This contrasts sharply with the original human finding, in which MORE risk identification led to significantly lower effort estimates and higher confidence across a family of four experiments.

The absence of the paradoxical effect is consistent across experiments A through D, including the colleague-framing variant (Experiment D). As noted in Section 3.6.3, Experiment D tests a different hypothesis from the original Study D and results are not compared to the human baseline directly. The null findings across Experiments A–C suggest that the illusion-of-control mechanism driving the human paradox [17] does not manifest in LLMs under these conditions.

4.3 Over-confidence

4.3.1 Connolly & Dean (1997). The Connolly replication tests three things: whether task decomposition affects whole-task estimates, whether less-than and greater-than question wording shift prediction intervals, and whether an extreme-limits elicitation procedure produces wider intervals than standard fractile elicitation [9]. Study 1 manipulates wording and order; Study 2 re-elicits intervals using the extreme-limits procedure after showing each model a summary of its own Study 1 response, replicating the feedback step in the original human design.

Whole-task medians (Figure 14) show no general decomposition effect. Gemini and Claude are insensitive to order. GPT, DeepSeek, and Kimi each show significant shifts in at least one

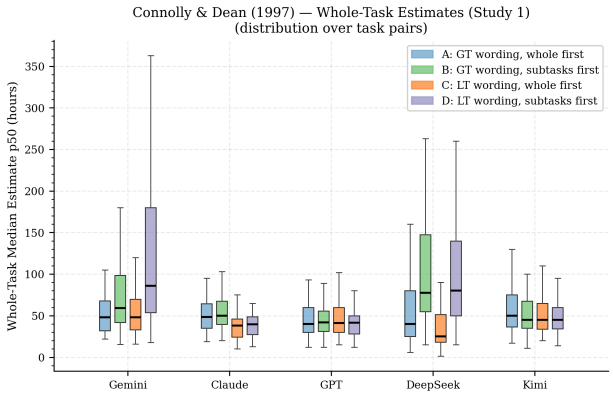

Figure 14: Connolly & Dean (1997) Study 1, whole-task median estimate across four conditions.

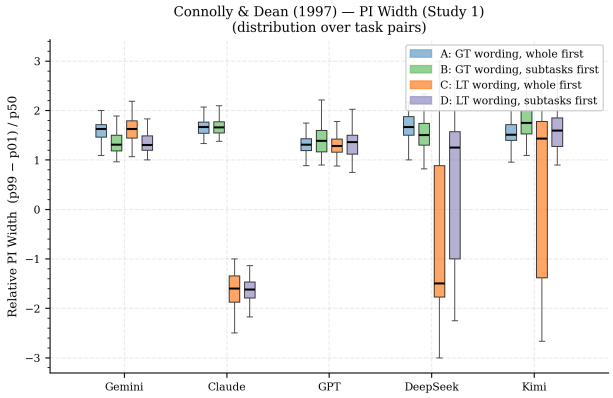

Figure 15: Connolly & Dean (1997) Study 1, relative PI width across four conditions.

condition, but in inconsistent directions (GPT and Kimi move opposite ways under GT wording), so no coherent cross-model pattern emerges.

The PI width results (Figure 15) expose a coherence problem under less-than wording. Claude produces strongly negative relative widths in conditions C and D, meaning the upper bound (p99) falls below the lower bound (p01). DeepSeek and Kimi show the same failure to a lesser extent. This is a wording-induced failure of interval validity, most severe in Claude but not unique to it. Among the models that do produce valid intervals, wording effects are limited: only Kimi (C vs A, $\delta = 0.230$, small) and DeepSeek (D vs B, $\delta = 0.192$, small) reach significance after correction, in opposing directions. The systematic wording effect from the original human study does not replicate.

Study 2 responses (Figures 16 and 17) show that whole-task medians remain comparable to Study 1, while PI widths are significantly wider in both conditions for all models (all $p < 0.001$), with effect sizes up to large for GPT, DeepSeek, and Kimi in condition A. The widening is driven by the elicitation change rather than any shift in point estimates, replicating the human finding that explicitly prompting for extreme plausible bounds produces wider intervals [9].

The subtask additive bias (Figure 18) is centred near zero for most models and conditions, with small effect sizes where significance is reached. Among the significant effects, negative gaps (subtask sums below the holistic estimate) outnumber positive ones, but the pattern is not consistent enough across models or

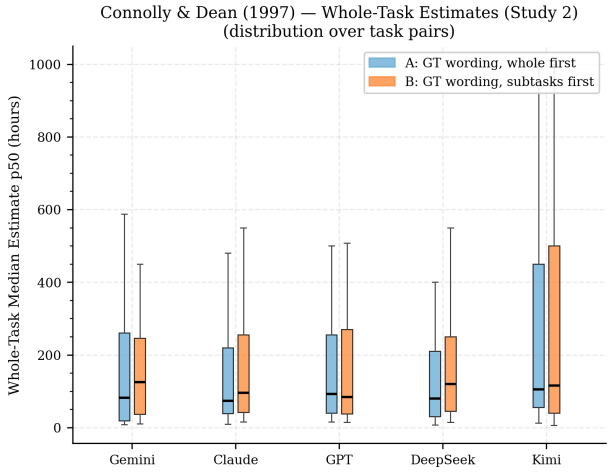

Figure 16: Connolly & Dean (1997) Study 2, whole-task median estimate under the extreme-limits procedure.

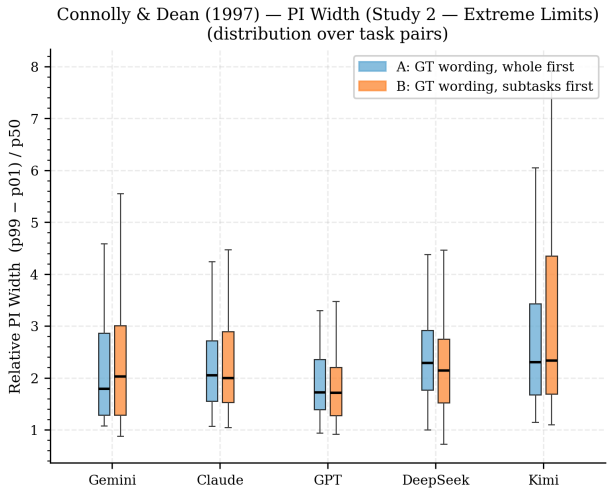

Figure 17: Connolly & Dean (1997) Study 2, relative PI width under the extreme-limits procedure.

conditions to support a general direction. The uniformly positive additive bias seen in the human study does not replicate.

Across both studies, the Connolly replication produces one clean match to humans (extreme-limits widen intervals) but no clear replication of the decomposition, wording, or additive-bias effects. The wording manipulation exposes interval-validity failures in three of five models under less-than framing, a finding the original human design did not surface and that represents a practical concern for LLM-based interval elicitation.

4.3.2 Jørgensen et al. (2002). Study A (Figure 19) reports the relative 90% prediction interval widths descriptively: Gemini (median = 0.772), Claude (1.014), GPT (1.000), DeepSeek (0.720), Kimi (1.050). These values are comparable to or narrower than those reported in the original human study [18], where professional software developers produced similarly tight intervals. DeepSeek and Gemini in particular produce notably narrow intervals, suggesting a degree of over-confidence in their uncertainty estimates.

Study B (Figure 20) shows that the group consensus estimate is significantly higher than the mean of individual role estimates for

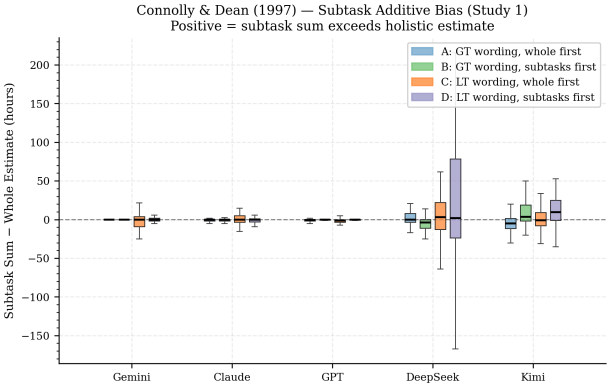

Figure 18: Connolly & Dean (1997) Study 1, subtask additive bias. Positive values indicate subtask sums exceed the holistic whole-task estimate.

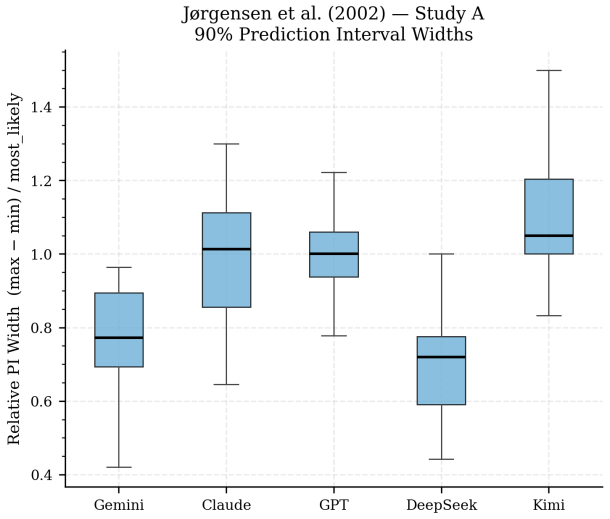

Figure 19: Jørgensen et al. (2002) Study A — relative 90% PI width per model.

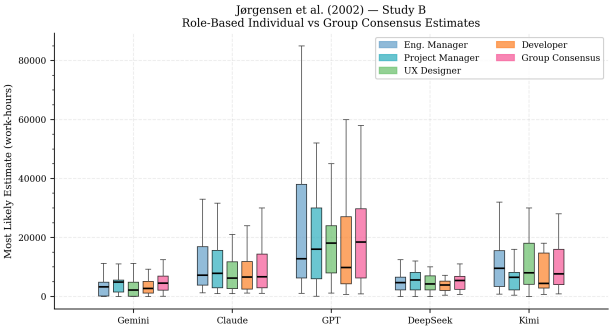

Figure 20: Jørgensen et al. (2002) Study B — most-likely estimates by role and group consensus.

all models [18] (all significant after Rom's correction), with large effects for Gemini ($\delta = 0.709$) and Claude ($\delta = 0.600$), a medium effect for DeepSeek ($\delta = 0.469$) and Kimi ($\delta = 0.345$), and a small effect for GPT ($\delta = 0.200$). The direction is consistently upward: the group consensus exceeds the individual mean, suggesting that the group consensus process tempers the over-optimism of individual role personas.

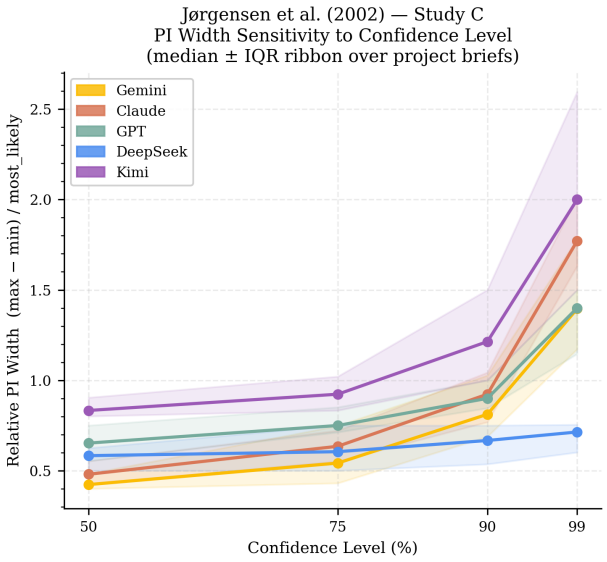

Figure 21: Jørgensen et al. (2002) Study C — PI width sensitivity to confidence level.

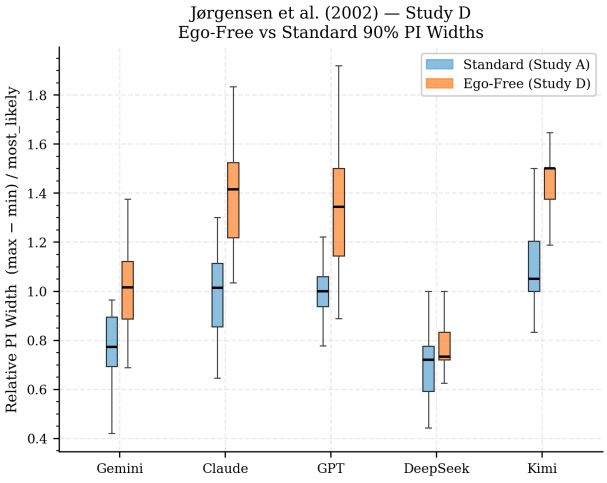

Figure 22: Jørgensen et al. (2002) Study D — PI width, standard vs ego-free framing.

Study C (Figure 21) shows that four of the five models increase their relative PI width significantly as the requested confidence level rises from 50% to 99% [18], with each step (50→75, 75→90, 90→99) significant and carrying a large effect size for Gemini, Claude, and GPT. Kimi shows significant growth from 75% upward. DeepSeek is the clear exception: none of the three steps reaches significance (δ values 0.108–0.173, all negligible to small), indicating that its interval width is largely insensitive to the requested confidence level.

Study D (Figure 22) uses the external-consultant framing described in Section 3.6.2, testing whether an outside perspective widens prediction intervals rather than reproducing the original ego-investment manipulation. The ego-free framing produces significantly wider intervals than the standard Study A framing for all models. Effect sizes are large for Claude ($\delta = 0.863$), GPT ($\delta = 0.697$), Kimi ($\delta = 0.594$), and Gemini ($\delta = 0.583$), and small for DeepSeek ($\delta = 0.259$). Models widen their intervals when prompted to adopt an external viewpoint, much as a human estimator might when explicitly asked to set aside personal stakes in an outcome.

4.4 Discussion

The results across 16 replicated experiments show that LLMs recreate many of the cognitive biases documented in the human SEE literature, most clearly and consistently for anchoring. Alongside these recreations, the replication surfaces two distinct departures from human behaviour that offer genuinely new insight into LLMs as estimators.

Anchoring is the most robustly replicated bias. Across the Aranda and Løhre experiments, all five models show consistent, significant shifts in the expected direction, and the directional asymmetry seen in humans (stronger pull toward high anchors than low) also appears in the LLM results [3]. Anchor value dominates format in Løhre Experiments 1 and 2, matching the human pattern [22]. The standout departure is Løhre Experiment 3. The original human study found no effect of source credibility on anchoring; four of the five LLMs show significant differences between low and high credibility conditions. LLM anchoring is influenced by source credibility where humans in the original study were unaffected by it, which is a meaningful distinction when deploying them in contexts where prompts routinely carry framing cues.

Over-optimism shows a more mixed picture, with a clear departure of its own. Moløkken's group discussion effect is directionally consistent with humans in four of five models, though effect sizes are smaller than in the original [25]. Haugen's planning poker process produces higher consensus estimates than unstructured estimation in all five models, directionally matching the human finding even where significance is not reached [13]. The risk identification paradox is the departure: across four experiments and all five models, the counter-intuitive human effect that more extensive risk identification leads to lower estimates and higher confidence is absent [17]. Every model-experiment combination fit the null hypothesis, suggesting the illusion-of-control mechanism proposed for the human result does not operate in LLMs under prompt-elicited risk identification.

Over-confidence replicates in several forms. Jørgensen (2002) Study A shows LLM prediction intervals comparable in width to those of human professionals, confirming that intervals are not appropriately wide [18]. Study C shows most models widen intervals as the requested confidence level rises, with DeepSeek as an outlier insensitive to the manipulation. Study D shows that external-perspective framing widens intervals across all models. Connolly Study 2, in which models received feedback on their Study 1 responses and were re-elicited using an extreme-limits procedure, shows wider intervals across all models, replicating the human effect [9]. The Connolly decomposition and wording effects do not cleanly replicate, and less-than wording produces invalid relative intervals in three of five models.

Taken together, LLMs recreate much of what the human SEE bias literature has documented. Anchoring transfers strongly and directly, over-confidence in interval calibration is present at levels comparable to human professionals, and group discussion effects on over-optimism move in the expected direction. Two departures stand out as distinct insights rather than simple non-replications: credibility framing influences LLM estimates in ways it does not influence experienced humans, and the risk identification paradox does not appear in LLMs at all. Substantial variance

Table 1: Summary of replicated experiments.

#	Source	Experiment	Bias	Key manipulation	Verdict
1	Aranda & Easterbrook	Anchoring direction & magnitude	Anchoring	Control vs. low anchor vs. high anchor	Replicated, all models: high anchor large (δ = 0.50–0.85); low anchor medium
2	Løhre & Jørgensen 1	High anchor, precision format	Anchoring	High anchor; precise vs. round vs. interval format	Replicated, all models: large effects; anchor value dominates format
3	Løhre & Jørgensen 2	Low anchor, precision format	Anchoring	Low anchor; precise vs. imprecise interval format	Replicated, all models: medium to large suppression; format negligible
4	Løhre & Jørgensen 3	Source credibility	Anchoring	Low vs. neutral vs. high credibility anchor	Not replicated: LLMs sensitive to credibility framing; humans showed no effect
5	Haugen	Planning poker vs. unstructured	Over-optimism	Unstructured vs. planning poker group process	Partially replicated: directionally consistent; significant for GPT only
6	Moløkken & Jørgensen	Group discussion effects	Over-optimism	Individual vs. group consensus vs. post-discussion	Replicated, all models: group consensus higher; convergence (4/5 models), small δ
7	Jørgensen (2009) A	Risk identification paradox	Over-optimism	Fewer vs. more risks identified	Not replicated: null across all models
8	Jørgensen (2009) B	Risk paradox, retrospective checklist	Over-optimism	Fewer vs. more risks with retrospective checklist	Not replicated: null across all models
9	Jørgensen (2009) C	Risk paradox + severity table	Over-optimism	Fewer vs. more risks with severity table	Not replicated: null across all models
10	Jørgensen (2009) D	Risk paradox, colleague framing	Over-optimism	Fewer vs. more risks, external-consultant framing	Not replicated: null across all models
11	Jørgensen et al. (2002) A	PI calibration, hit rate	Over-confidence	Standard 90% PI framing	Replicated, all models: PI widths comparable to human professionals
12	Jørgensen et al. (2002) B	Role-based vs. group consensus	Over-confidence	Individual roles vs. group consensus	Replicated, all models: group consensus exceeds individual mean (δ = 0.20–0.71)
13	Jørgensen et al. (2002) C	PI sensitivity to confidence level	Over-confidence	Varying confidence level (50%–99%)	Replicated, inconsistent: 4/5 models widen with level; DeepSeek insensitive
14	Jørgensen et al. (2002) D	Ego-free framing	Over-confidence	Standard vs. external-perspective framing	Replicated, all models: ego-free framing widens PIs (δ = 0.26–0.86)
15	Connolly & Dean 1	Fractile wording & task order	Over-confidence	GT vs. LT wording; whole-task vs. subtasks-first	Not replicated, inconsistent: LT wording causes pathological PI inversion (Claude); decomposition effects model-specific
16	Connolly & Dean 2	Extreme-limits elicitation	Over-confidence	Standard vs. extreme-limits elicitation	Replicated, all models: extreme-limits procedure widens intervals

between models runs throughout the results, with DeepSeek notably insensitive to several manipulations where other models respond and different models producing different patterns in the Connolly wording conditions. No single model can be treated as a drop-in substitute for a human estimator, and the pattern of which biases transfer and which do not is itself a useful map for future work.

5 Implications

5.1 Implications for SE Researchers

This study substantially extends the Machine Psychology foundation in SEE beyond the single prior study [8], and the pattern of results offers concrete direction for future work.

The most grounded contribution is the demonstration of LLMs as pilot participants for SEE bias experiments. Across 16 experiments replicated across 5 models, the full experimental pipeline ran in a timeframe and at a scale that would be impractical with human recruitment. For researchers designing new bias experiments, this supports a workflow in which experimental designs and materials are refined on LLM participants before committing to expensive human trials. Within this specific context, the scale of LLM participation also enables higher-powered tests than typical human SEE studies, where sample sizes are often constrained by the cost and scarcity of qualified professionals [26].

For well-matched biases, the case for synthetic participants becomes reasonable to consider more seriously. Anchoring is the

clearest candidate: all five models matched human directional patterns across every anchoring experiment in this study, including the high-low asymmetry seen in Aranda and the value dominating format as seen in Løhre. A researcher designing a new anchoring manipulation could plausibly run the full experiment on LLM participants, with human validation reserved for the specific effect of interest rather than the full design. We do not claim that LLMs replace humans in general, but the anchoring evidence supports exploring synthetic-only designs in targeted cases where we have shown the bias to transfer robustly.

The two clearest departures (credibility framing sensitivity in Løhre Experiment 3, and the absence of the risk identification paradox across Jørgensen (2009)) are concrete open questions that warrant systematic follow-up. Neither finding was anticipated from the human literature, and both emerged directly from the replication paradigm, suggesting that extending the replication cohort to further biases and further models is likely to surface further distinct LLM behaviours.

Finally, the replication package itself is a research artefact [33]. It is organised to support extension: additional models can be added to the cohort, additional experiments can be slotted into the existing statistical framework, and the whole pipeline can be rerun as models evolve. Characterising how LLM behaviour on these experiments shifts across model generations is a natural next step.

5.2 Implications for SE Practitioners

LLMs are already used in estimation workflows, and the results here offer both warnings and concrete techniques for practitioners who are deploying them.

The primary practical warning is anchoring. All five models are significantly pulled toward numerical values present in the prompt, with effects stronger toward high anchors than low. This matters because estimation requests in practice routinely arrive with numbers already present: prior estimates from a previous sprint, budget figures, framed questions, rough figures from adjacent projects. Each of these functions as an anchor. The Løhre Experiment 3 results add a second layer to this warning. LLMs anchor on low-credibility sources where experienced humans appear to discount them, and anchor even more strongly on high-credibility sources. A number attributed to a senior developer or a formal estimate will have a larger effect on the LLM's output than the same number attributed to an administrator, but both will pull the estimate. Practitioners should assume that any number in the prompt will affect the estimate, with the magnitude of that effect sensitive to how the number is framed.

Two concrete techniques demonstrated in this study consistently widen prediction intervals across all models:

- **External-consultant framing.** Prompting the model to estimate as an outside consultant rather than as the responsible developer widens intervals across all models, with large effects in four of five.
- **Extreme-limits elicitation.** Explicitly prompting for the most extreme plausible upper and lower bounds rather than standard fractiles produces wider intervals across all models, replicating the human effect.

Both techniques are cheap, require no additional tooling, and are worth adopting in any workflow where uncertainty quantification matters.

Many estimation workflows already include risk identification as a step. The non-replication of the risk identification paradox in LLMs is worth flagging here: the counter-intuitive human effect where listing more risks leads to lower estimates and higher confidence does not appear in LLMs. This is not a feature. The human paradox is thought to arise from an illusion-of-control mechanism triggered by listing risks [17]; LLMs not showing this paradox does not mean they are handling risk better, only that prompt-elicited risk listing does not engage the same underlying process. Practitioners who rely on risk identification steps for their protective value should not assume the same value carries over when an LLM performs them.

More broadly, the similarities between LLM and human estimation behaviour documented in this study (anchoring, over-confidence in interval calibration at human-professional levels, upward movement under group discussion) suggest that LLMs are not a fundamentally alien reasoning system when deployed alongside humans in estimation tasks. Practitioners familiar with human bias mitigation will recognise much of what LLMs do, and workflows that include LLMs as contributors need not treat them as a special case for most bias categories. The novel departures documented here are specific items to watch for, not grounds for wholesale rejection. The usual rule for high-stakes estimates applies: human oversight remains necessary, particularly in contexts (proprietary codebases, novel domains, accountability-bearing decisions) that fall outside what any bias study has tested.

5.3 Implications for SE Educators

SEE is a standard topic in professional software engineering courses, and the role of LLMs in estimation is increasingly relevant to what those courses cover [20]. This study contributes to that adaptation in three connected ways: it provides evidence, it provides materials, and it introduces a paradigm.

The evidence is directly teachable content. LLMs do not solve the over-confidence and over-optimism problems that the SEE literature has documented for decades [26]; they reproduce several of them. Students deploying LLMs in estimation workflows need to understand that the standard failure modes do not go away when a human is replaced with a language model. The Løhre credibility finding is particularly teachable because it extends rather than mirrors the human pattern: LLMs anchor on numbers in prompts with a gradient that responds to source credibility, a sensitivity that experienced humans appear to discount. Students can be shown both the human result and the LLM result and asked to reason about why the two systems respond differently to the same framing cue.

The replication package [33] provides ready-made material for lab sessions. Experiments are straightforward to run; a student can observe anchoring emerge from a model they interact with daily in a single lab session, which is a more direct encounter with the bias than any description. The same materials that serve as a research extension for researchers serve as a teaching resource here, and the prompts are documented to a level that allows students to vary them and observe the effect on model behaviour.

Beyond specific content, the Machine Psychology paradigm itself is teachable. Students designing and running their own bias experiments on LLMs learn both empirical method and critical evaluation of AI tools in a single exercise, and the setup prepares them for a working environment in which they will be estimating alongside, not just using, language models. This points toward a distinct strand within existing SE curriculum adaptation efforts, in which synthetic participation and LLM bias evaluation are

taught as a recognisable topic rather than an incidental addition to other AI-tools content. The paper does not demonstrate the educational effectiveness of such a strand, but the materials and the paradigm to support it are both available.

6 Threats to Validity

Internal validity. We executed all experiments at temperature zero, minimising stochastic variation and ensuring that observed differences between conditions can be attributed to the experimental manipulation rather than random generation noise. A consequence of this design is that we collected a single response per condition per document per model, rather than averaging over repeated samples. Temperature zero does not guarantee strictly identical outputs across repeated calls due to implementation differences between providers [5], and we treat this as an assumption the design rests on rather than one the study verifies. Query-level caching by the OpenRouter API or underlying providers could in principle cause identical prompts to return cached responses; OpenRouter operates statelessly and each prompt in this study contains a unique requirements document, so cache collisions across observations are practically implausible, but we note the risk. Anchor values in TAWOS-based Aranda conditions were calibrated from a separate pilot cohort of models under control-condition estimates, reducing but not eliminating the possibility that anchor magnitudes are confounded with experimental-cohort behaviour.

Construct validity. Every experiment in this study substitutes LLM responses for human responses, and every requirements document is derived from the TAWOS dataset rather than the materials used in the original studies. Both substitutions are uniform methodological assumptions rather than experiment-specific concerns, and findings throughout the paper rest on them. Beyond these uniform substitutions, three experiments involve named adaptations that deviate further from the originals: the Haugen group process is implemented using persona-based sequential prompting, the Jørgensen (2002) Study D ego-investment manipulation is replaced by external-consultant framing, and the Jørgensen (2010) Study D cross-domain manipulation is replaced by colleague framing. Results from these three experiments should be read alongside the corresponding details in Section 3.6.

External validity. The five models selected represent high-performing frontier systems at the time of the study but do not cover the full landscape of available LLMs. Results may not generalise to smaller, domain-specific, or open-weight models, and LLM behaviour on these experiments is likely to shift as model generations evolve. The TAWOS dataset is drawn from large open-source projects using Jira; findings may not generalise to proprietary or differently structured codebases. The agile-format issue documents cover a limited number of projects, with Spring XD disproportionately represented as an artefact of the dataset. This introduces project-level dependence in results from the Moløkken and Haugen experiments that cannot be fully controlled for.

7 Conclusions

This study asked whether large language models, increasingly used in estimation workflows, exhibit the cognitive biases that have been documented in human software effort estimation for decades. We answered the question through a systematic replication of 16 experiments drawn from 7 source papers, using 5 frontier models as participants and TAWOS-derived requirements documents as experimental materials. The scale of replication extends prior Machine Psychology work in SEE from a single study on a single bias [8] to a broader corpus spanning anchoring, over-optimism, and over-confidence.

LLMs recreate much of the human SEE bias literature. Anchoring transfers most robustly, present across all models and all anchoring experiments. Over-confidence in interval calibration is comparable to human professionals, and group discussion effects on over-optimism move in the expected direction. Two findings diverge from the human baseline in ways that are themselves informative. LLM anchoring is influenced by source credibility where humans in the original study were unaffected by it, and the risk identification paradox documented in the human literature is absent across four experiments and all five models.

Alongside these findings, the replication package [33] is organised to support further work: the cohort can be extended to more models, the pipeline can be rerun as models evolve, and the experimental framework can accommodate further biases from the corpus of candidate papers identified during selection. The state of LLM behaviour on these experiments is a moving target, and the materials here are intended to let others continue mapping it.

We make the replication package, including all experimental materials, data, and analysis scripts, available at: https://github.com/tjqscott/LLM-SEE.

Acknowledgments

I would like to thank Dr Handan Gül Çalıklı for supervision and guidance throughout this project.

References

[1] [n. d.]. OpenRouter: A Unified Interface for Large Language Models. https://openrouter.ai Accessed: 2025.

[2] Sahar Alturki and Fazal E-amin. 2025. Comprehensive Analysis of Software Effort Estimation Techniques: Evolving Trends, Key Challenges, and Prospective Directions. *International Journal of Computer Applications* 186, 68 (Feb. 2025), 42–48. doi:10.5120/ijca2025924523

[3] Jorge Aranda and Steve Easterbrook. 2005. Anchoring and adjustment in software estimation. *ACM SIGSOFT Software Engineering Notes* 30, 5 (Sept. 2005), 346–355. doi:10.1145/1095430.1081761

[4] Marcel Binz and Eric Schulz. 2023. Using cognitive psychology to understand GPT-3. *Proceedings of the National Academy of Sciences* 120, 6 (Feb. 2023). doi:10.1073/pnas.2218523120

[5] Bjarni Haukur Bjarnason, André Silva, and Martin Monperrus. 2026. On Randomness in Agentic Evals. doi:10.48550/ARXIV.2602.07150

[6] Barry W. Boehm. 1984. Software Engineering Economics. *IEEE Transactions on Software Engineering* SE-10, 1 (Jan. 1984), 4–21. doi:10.1109/tse.1984.5010193

[7] Thanh-Long Bui, Hoa Khanh Dam, and Rashina Hoda. 2025. An LLM-based multi-agent framework for agile effort estimation. doi:10.48550/ARXIV.2509.14483

[8] Gül Calikli and Mohammed Alhamed. 2025. Impact of Request Formats on Effort Estimation: Are LLMs Different Than Humans? *Proceedings of the ACM on Software Engineering* 2, FSE (June 2025), 1114–1135. doi:10.1145/3715771

[9] Terry Connolly and Doug Dean. 1997. Decomposed Versus Holistic Estimates of Effort Required for Software Writing Tasks. *Management Science* 43, 7 (July 1997), 1029–1045. doi:10.1287/mnsc.43.7.1029

[10] Danica Dillion, Niket Tandon, Yuling Gu, and Kurt Gray. 2023. Can AI language models replace human participants? *Trends in Cognitive Sciences* 27, 7 (July 2023), 597–600. doi:10.1016/j.tics.2023.04.008

[11] Thilo Hagendorff, Ishita Dasgupta, Marcel Binz, Stephanie C Y Chan, Andrew Lampinen, Jane X Wang, Zeynep Akata, and Eric Schulz. 2023. Machine Psychology. (2023). arXiv:2303.13988 [cs.CL]

[12] Thilo Hagendorff, Sarah Fabi, and Michal Kosinski. 2023. Human-like intuitive behavior and reasoning biases emerged in large language models but disappeared in ChatGPT. *Nature Computational Science* 3, 10 (Oct. 2023), 833–838. doi:10.1038/s43588-023-00527-x

[13] N.C. Haugen. [n. d.]. An Empirical Study of Using Planning Poker for User Story Estimation. In *AGILE 2006 (AGILE'06)*. IEEE, 23–34. doi:10.1109/agile.2006.16

[14] Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. 2024. SWE-bench: Can Language Models Resolve Real-world GitHub Issues?. In *Proceedings of the International Conference on Learning Representations*. https://arxiv.org/abs/2310.06770

[15] Magne Jørgensen and Torleif Halkjelsvik. 2010. The effects of request formats on judgment-based effort estimation. *J. Syst. Softw.* 83, 1 (Jan. 2010), 29–36.

[16] M. Jørgensen. 2004. A review of studies on expert estimation of software development effort. *Journal of Systems and Software* 70, 1–2 (Feb. 2004), 37–60. doi:10.1016/s0164-1212(02)00156-5

[17] Magne Jørgensen. 2010. Identification of more risks can lead to increased over-optimism of and over-confidence in software development effort estimates. *Information and Software Technology* 52, 5 (May 2010), 506–516. doi:10.1016/j.infsof.2009.12.002

[18] Magne Jørgensen, Karl Halvor Teigen, and Kjetil Moløkken. 2004. Better sure than safe? Over-confidence in judgement based software development effort prediction intervals. *Journal of Systems and Software* 70, 1–2 (Feb. 2004), 79–93. doi:10.1016/s0164-1212(02)00160-7

[19] Daniel Kahneman. 2011. *Thinking, Fast and Slow.* Farrar, Straus and Giroux, New York.

[20] Vassilka D. Kirova, Cyril S. Ku, Joseph R. Laracy, and Thomas J. Marlowe. 2024. Software Engineering Education Must Adapt and Evolve for an LLM Environment. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1*. 666–672. doi:10.1145/3626252.3630927

[21] Albert L Lederer and Jayesh Prasad. 1992. Nine management guidelines for better cost estimating. *Commun. ACM* 35, 2 (1992), 51–59. doi:10.1145/129630.129638

[22] Erik Løhre and Magne Jørgensen. 2016. Numerical anchors and their strong effects on software development effort estimates. *Journal of Systems and Software* 116 (June 2016), 49–56. doi:10.1016/j.jss.2015.03.015

[23] Tim Menzies, Amer El-Rawas, Justin Di Stefano, and Bojan Cukic. 2002. Metrics that matter. *IEEE Transactions on Software Engineering* 28, 8 (2002), 801–802. doi:10.1109/TSE.2002.1027796

[24] Rahul Mohanani, Iflaah Salman, Burak Turhan, Pilar Rodriguez, and Paul Ralph. 2020. Cognitive Biases in Software Engineering: A Systematic Mapping Study. *IEEE Transactions on Software Engineering* 46, 12 (Dec. 2020), 1318–1339. doi:10.1109/tse.2018.2877759

[25] Kjetil Moløkken and Magne Jørgensen. 2003. Software Effort Estimation: unstructured group discussion as a method to reduce individual biasis.. In *PPIG*. 4.

[26] Kjetil Moløkken-Østvold and Magne Jørgensen. 2003. A review of surveys on software effort estimation. In *Proceedings of the 2003 International Symposium on Empirical Software Engineering*. IEEE. doi:10.1109/ISESE.2003.1237981

[27] Mourad Ouzzani, Hossam Hammady, Zbys Fedorowicz, and Ahmed Elmagarmid. 2016. Rayyan — a web and mobile app for systematic reviews. *Systematic Reviews* 5, 210 (2016). doi:10.1186/s13643-016-0384-4

[28] Luka Pavlič, Vasilka Saklamaeva, and Tina Beranič. 2024. Can Large-Language Models Replace Humans in Agile Effort Estimation? Lessons from a Controlled Experiment. *Applied Sciences* 14, 24 (Dec. 2024), 12006. doi:10.3390/app142412006

[29] R Core Team. 2024. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. https://www.R-project.org/

[30] Paul Ralph et al. 2020. ACM SIGSOFT Empirical Standards. *arXiv preprint arXiv:2010.03525* (2020).

[31] Dror M Rom. 1990. A sequentially rejective test procedure based on a modified Bonferroni inequality. *Biometrika* 77, 3 (1990), 663–665.

[32] Jeanine Romano et al. 2006. Appropriate statistics for ordinal level data: Should we really be using t-test and Cohen's d for evaluating group differences on the NSSE and other surveys? *Assessment* 13 (2006).

[33] Taylor Scott. 2025. LLM-SEE: Replication Package for "Uncovering the Cognitive Biases of LLMs in Software Effort Estimation". https://github.com/tjqscott/LLM-SEE.

[34] Dag IK Sjøberg, Jo E. Hannay, Ove Hansen, Vigdis By Kampenes, Amela Karahasanovic, Nils-Kristian Liborg, and Anette C. Rekdal. 2005. A survey of controlled experiments in software engineering. *IEEE Transactions on Software Engineering* 31, 9 (2005), 733–753. doi:10.1109/TSE.2005.97

[35] Webb Stacy and Jean MacMillan. 1995. Cognitive bias in software engineering. *Commun. ACM* 38, 6 (June 1995), 57–63.

[36] Yoshiki Takenami, Yin Jou Huang, Yugo Murawaki, and Chenhui Chu. 2025. How Does Cognitive Bias Affect Large Language Models? A Case Study on the Anchoring Effect in Price Negotiation Simulations. doi:10.48550/ARXIV.2508.21137

[37] Vali Tawosi, Afnan Al-Subaihin, Rebecca Moussa, and Federica Sarro. 2022. A versatile dataset of agile open source software projects. In *Proceedings of the 19th International Conference on Mining Software Repositories (MSR '22)*. ACM, 707–711. doi:10.1145/3524842.3528029

[38] Amos Tversky and Daniel Kahneman. 1974. Judgment under Uncertainty: Heuristics and Biases. *Science* 185, 4157 (1974), 1124–1131. doi:10.1126/science.185.4157.1124