# Impact of Request Formats on Effort Estimation: Are LLMs Different Than Humans?

GÜL ÇALIKLI, University of Glasgow, United Kingdom
MOHAMMED ALHAMED, Applied Behaviour Systems LTD (Hexis), United Kingdom

Software development Effort Estimation (SEE) comprises predicting the most realistic amount of effort (e.g., in work hours) required to develop or maintain software based on incomplete, uncertain, and noisy input. Expert judgment is the dominant SEE strategy used in the industry. Yet, expert-based judgment can provide inaccurate effort estimates, leading to projects' poor budget planning and cost and time overruns, negatively impacting the world economy. Large Language Models (LLMs) are good candidates to assist software professionals in effort estimation. However, their effective leveraging for SEE requires thoroughly investigating their limitations and to what extent they overlap with those of (human) software practitioners. One primary limitation of LLMs is the sensitivity of their responses to prompt changes. Similarly, empirical studies showed that changes in the request format (e.g., rephrasing) could impact (human) software professionals' effort estimates. This paper reports the first study that replicates a series of SEE experiments, which were initially carried out with software professionals (humans) in the literature. Our study aims to investigate how LLMs' effort estimates change due to the transition from the traditional request format (i.e., "How much effort is required to complete X?") to the alternative request format (i.e., "How much can be completed in Y work hours?"). Our experiments involved three different LLMs (GPT-4, Gemini 1.5 Pro, Llama 3.1) and 88 software project specifications (per treatment in each experiment), resulting in 880 prompts, in total that we prepared using 704 user stories from three large-scale open-source software projects (Hyperledger Fabric, Mulesoft Mule, Spring XD). Our findings align with the original experiments conducted with software professionals: The first four experiments showed that LLMs provide lower effort estimates due to transitioning from the traditional to the alternative request format. The findings of the fifth and first experiments detected that LLMs display patterns analogous to anchoring bias, a human cognitive bias defined as the tendency to stick to the anchor (i.e., the "Y work-hours" in the alternative request format). Our findings provide crucial insights into facilitating future human-AI collaboration and prompt designs for improved effort estimation accuracy.

CCS Concepts: • **Software and its engineering → Software development process management**; **Risk management**.

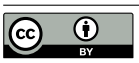
Additional Key Words and Phrases: Software effort estimation, Cognitive bias, Large Language Models (LLMs), Human judgement, Empirical software engineering

**ACM Reference Format:**
Gül Çalıklı and Mohammed Alhamed. 2025. Impact of Request Formats on Effort Estimation: Are LLMs Different Than Humans?. *Proc. ACM Softw. Eng.* 2, FSE, Article FSE051 (July 2025), 22 pages. https://doi.org/10.1145/3715771

## 1 Introduction

Software Effort Estimation (SEE) aims to predict the most realistic amount of effort (e.g., in person-hours) required to develop or maintain software based on incomplete, uncertain, and noisy input. Accurate effort estimation is crucial for facilitating software development's overall success by

Authors' Contact Information: Gül Çalıklı, University of Glasgow, Glasgow, United Kingdom, HandanGul.Calikli@glasgow.ac.uk; Mohammed Alhamed, Applied Behaviour Systems LTD (Hexis), Worcester, United Kingdom.




ACM 2994-970X/2025/7-ARTFSE051
https://doi.org/10.1145/3715771

ensuring that projects are delivered on time and within budget. Poor effort estimation increases costs due to project overruns and can lead to project failures.

Expert-based judgment is the most frequently applied effort estimation strategy for software projects in industry [Jørgensen 2004; Moløkken and Jørgensen 2003]. Indeed, software professionals can even involve their judgment while using formal effort estimation models [Jørgensen and Gruschke 2005]. For instance, formal models such as COCOMO require the software practitioners to predict the number of lines of code the future system will have [Aranda and Easterbrook 2005].

Expert judgment-based effort estimation involves employing heuristics, which are mental shortcuts humans use to make decisions. However, failure of heuristics leads to "cognitive biases" that are systematic deviations from optimal reasoning [Stanovich 2008] and can also defined as recurring errors in thinking, or patterns of bad judgment [Mohanani et al. 2020]. The term "cognitive bias" was first coined in the psychology literature by Tversky and Kahneman [Tverskey and Kahneman 1978]. Due to cognitive biases, software professionals' effort estimates can be overly optimistic (i.e., underestimating the required software development effort) [Jørgensen 2010] and over-confident (i.e., minimum-maximum effort interval that reflects the chosen confidence level can be pretty narrow) [Jørgensen 2010; Jørgensen et al. 2004].

One way to mitigate software practitioners' cognitive biases and facilitate more accurate effort estimations can be through human-AI collaboration. The rapid advancements in computing power, data availability, and learning algorithms (e.g., deep neural nets) unlock the potential of AI-based models to improve the accuracy of effort estimates. Furthermore, Large Language Models (LLMs), which are trained using large amounts of data, can outperform traditional formal models that rely on less data and assist software practitioners in effort estimations. However, it is crucial to understand LLMs' shortages to leverage them to help software professionals effectively estimate effort. One primary well-known shortage of LLMs is their output's sensitivity to prompt changes.

Similarly, irrelevant information can also hamper human judgement [Kahneman 2011]. Studies in software engineering report that changes in request formats strongly influence software professionals' judgment, leading to variations in effort estimates [Jørgensen 2006]. A series of controlled experiments Jørgensen and Halkjelsvik conducted showed that software professionals tend to make lower and presumably overoptimistic effort estimates when the traditional request format (i.e., "*How much effort is required to complete X?*") was changed to the alternative request format (i.e., "*How much can be completed in Y work-hours?*,").

In this paper, we replicated Jørgensen and Halkjelsvik's series of experiments [Jørgensen and Halkjelsvik 2010], which we call ORIGINAL STUDY in the rest of the paper, using three LLMs, which are GPT-4, GEMINI 1.5 PRO, LLAMA 3.1, respectively. We initially conducted a preliminary study preparing prompts identical to the tasks in the first and second experiments of the ORIGINAL STUDY. We observed that LLMs' responses showed similarities with responses of human participants in the ORIGINAL STUDY. To investigate further, we conducted a series of experiments having prepared 88 prompts (per treatment in each experiment) using 704 user stories from three open-source software projects (Hyperledger Fabric, Mulesoft Mule, Spring XD).

Our findings show that transitioning from traditional to alternative formats leads to lower effort estimates. Moreover, similar to what ORIGINAL STUDY observed in human participants' responses, in our experiments, we observed that one main reason for low effort estimates in the alternative format is that LLMs anchor their estimates to the time frame (30 work hours) mentioned in the posed question (i.e.,"*"How much can be completed in Y work-hours?"*)"). Our study is the first "Machine Psychology" study within the context of software effort estimation. "Machine Psychology" studies conduct experiments initially designed to test human participants to elicit LLMs' failure modes [Hagendorff 2023]. They can help design prompts to improve LLM-based effort estimation accuracy due to their focus on the correlation between the prompt input and the prompt output.

## 2 Background and Related Work

This section provides an overview of the empirical Software Engineering (SE) studies conducted in the last two decades, which showed the impact of irrelevant information or request format changes in requirements specification documents on software professionals' effort estimates. It also highlights the significant role of AI-based effort estimation techniques in the SE literature, including those that leverage LLMs. The section concludes with studies investigating the similarities between LLMs' erroneous behavior and human judgment.

### 2.1 Human Judgement in Software Effort Estimation

Various studies showed that expert judgment in software development estimation can deteriorate due to irrelevant information [Løhre and Jørgensen 2016] and request format changes [Jørgensen and Halkjelsvik 2010] in the requirements specification documents.

Irrelevant information can be obsolete requirements [Gren and Berntsson Svensson 2021], effort spent on the development of the old system the client wants to replace, client's budget [Jorgensen and Grimstad 2011], inexperienced software professionals' implausible comments about effort estimates [Aranda and Easterbrook 2005] and high productivity on the current task [Jørgensen and Grimstad 2008]. For instance, in the between-subject experiment (i.e., ANCHORING EXPERIMENT), Aranda and Easterbrook [Aranda and Easterbrook 2005] found that the participants' effort estimates varied significantly depending on the existence of the implausible effort estimates in the provided project setting documents and the effort estimates' magnitude (2 months vs. 20 months).

Request format changes comprise a variation in the wording when describing the task to be estimated [Jørgensen and Grimstad 2008] or inquiring about the effort estimate [Jørgensen and Halkjelsvik 2010] with no change in the requirement specification. In the ORIGINAL STUDY, Jørgensen and Halkjelsvik [Jørgensen and Halkjelsvik 2010] found that software professionals tend to make lower and presumably overoptimistic effort estimates for the traditional request format (i.e., "*How much effort is required to complete X?*") compared to the estimates for the alternative request format (i.e., "*How much can be completed in 1 week?*"). Irrelevant information or changes in request formats can trigger software practitioners' anchoring bias.

**Anchoring Bias in SEE.** Anchoring bias is the most common cognitive bias in software engineering studies [Mohanani et al. 2020] and among the leading causes for highly inaccurate and systematically over-optimistic software development effort estimates [Aranda and Easterbrook 2005; Jørgensen et al. 2004]. Anchoring bias manifests due to the failure of the anchoring and adjustment heuristic humans use to estimate uncertain quantities. According to the heuristic, humans start from a number (i.e., an anchor) and adjust their estimate by mentally moving away from the anchor. Anchoring bias manifests when the adjustment ends prematurely, staying close to the anchor [Epley and Gilovich 2006; Kahneman 2011]. In the ORIGINAL STUDY, Jørgensen and Halkjelsvik [Jørgensen and Halkjelsvik 2010] showed that software professionals provided with the alternative request format anchored their estimates to "1 week" and ended up with lower estimates due to insufficient adjustment (made by moving up) away from the anchor to estimate the effort required to complete the remaining requirements (user stories).

**Measuring Anchoring Bias.** Although many psychological phenomena are demonstrable through experiments, very few are measurable. The measurement of the effect of anchoring bias (i.e., anchoring) is among the few exceptions. One can measure the anchoring using the "anchoring index" [Kahneman 2011], which is the ratio $\Delta_{outcome}/\Delta_{anchors}$, where $\Delta_{anchors}$ is the difference between the two anchors used in the treatments in an experiment and $\Delta_{outcome}$ is the difference between the median values of participants' estimates for the two anchors. For instance, in the ANCHORING EXPERIMENT by Aranda and Easterbrook [Aranda and Easterbrook 2005] (mentioned

above in this subsection), the two anchors are 2 months and 20 months. Hence, $\Delta_{anchors}$ = 20 months - 2 months = 18 months. The mean effort estimate of participants provided with two months as the anchor was 5.1 months, and the mean estimate for those provided with 20 months as the anchor was 15.4 months. Therefore, $\Delta_{outcome}$ = 15.4 months - 5.1 months = 10.3 months, giving an anchoring index of 10.3/18 = 0.572, which we can express as a percentage: %57.2

### 2.2 AI-based Software Effort Estimation

While expert estimation continues to be the prevailing effort estimation strategy [Jørgensen 2004; Moløkken and Jørgensen 2003; Sarro et al. 2016], it's important to acknowledge that humans have cognitive limitations (e.g., memory, working memory). However, with the rapid advancements in computing power, data availability, and learning algorithms (e.g., deep neural nets), the potential of human-AI collaboration to provide more accurate effort estimations is a promising frontier. The most explored AI-based techniques include Artificial Neural Networks (ANN), Support Vector (SV) Machines/Regression, Decision Trees, and Case-based Reasoning (CBR) [Ali and Gravino 2019; Wen et al. 2012]. One of the AI-based approaches that stands out for providing effort estimates comparable to the best human expert-based results is the multi-objective evolutionary approach by Sarro et al. [Sarro et al. 2016]. The authors' proposed Search-based Software Engineering (SBSE) approach, which also outperformed CBR, linear regression, and regression trees, is designed to build a robust model. This model minimizes the uncertainty associated with the effort estimation model (by conducting a confidence interval analysis) and maximizes the estimation accuracy simultaneously, providing a reliable solution.

**Leveraging LLMs for Software Effort Estimation.** The effort estimation technique by Choetkiertikul et al. [Choetkiertikul et al. 2019] is close to LLM-based approaches as it combines Long Short-Term Memory (LSTM) and Recurrent Highway Network (RHN) architectures. Alhamed and Storer [Alhamed and Storer 2022] employ BERT tokenization and TF-IDF for feature extraction, training two Machine Learning models: BERT's feed-forward network linear classifier and Random Forests. The results obtained using BERT both for feature extraction and as a classifier are slightly better than other combinations (with no statistical significance) and show that the accuracy of expert-based estimates and the employed Machine Learning techniques are similar, with experts' performance being slightly better. Favero et al. [Fávero et al. 2022] fine-tune pre-trained models for software effort estimation, including BERT. However, the results suggest over-fitting, and the study needs further investigation into possible biases. Tawosi et al. [Tawosi et al. 2023] proposed an SBSE technique to leverage LLMs for Story Point (SP) effort estimation through few-shot learning. The authors' seach-based optimisation technique improved the estimation performance of the LLM by more than 50% on average (in terms of mean absolute error of the estimation) against a zero-shot setting.

### 2.3 LLMs' Complex Behaviour vs. Human Judgement

Since the human mind is complex, experimental psychologists craft experiments to reveal a systematic phenomenon about human thought processes without gaining complete mechanistic insight into the human minds they aim to analyze. Similarly, LLMs are becoming increasingly complex, exhibiting behavioral patterns similar to humans [Hagendorff 2023; Hagendorff et al. 2023]. Therefore, since gaining complete mechanistic insight into LLMs' behavior is a challenge, some recent studies treat LLMs as participants of psychology experiments, which were initially designed for humans, to investigate LLMs' behavioral patterns and uncover their previously undetected emergent characteristics [Binz and Schulz 2023; Hagendorff et al. 2023]. Hagendorff uses "Machine Psychology" as the umbrella term for the mentioned studies, which focus on the correlation between the prompt input and the prompt output when applying psychological tests. Some machine psychology studies

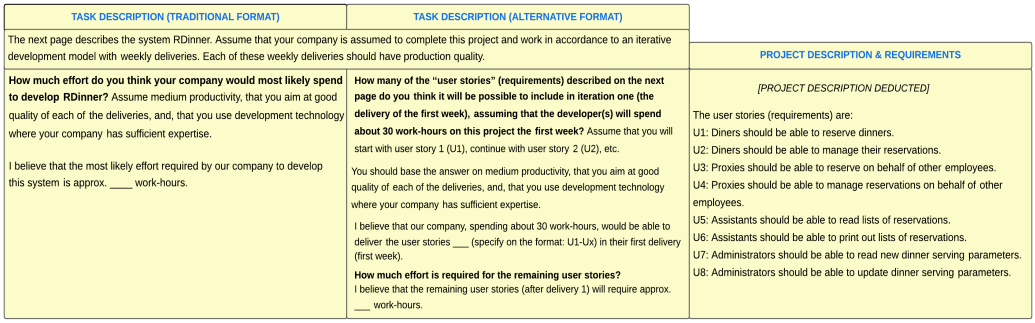



Fig. 1. Experiment task document the participants of STUDY1 in the ORIGINAL STUDY received contains the task description (first page) and the project description followed by eight user stories (second page). Project description on the second page is deducted due to space limitations. Task description for the traditional and alternative format groups differ while the contents of the second page remain the same for both traditional and alternative formats.

conduct experiments designed to test (human) cognitive biases [Binz and Schulz 2023; Dasgupta et al. 2022; Macmillan-Scott and Musolesi 2024; Talboy and Fuller 2023] to elicit LLMs' failure modes. Among the mentioned studies, the study by Jones and Steinhardt [Jones and Steinhardt 2024] focuses on anchoring bias within the context of LLM-based code generation. The study gets inspiration from anchoring bias, among other cognitive biases, to explain LLMs' erroneous behavior during code generation. Software professionals' anchoring bias is the most widely studied cognitive bias in software effort estimation [Mohanani et al. 2020]. Our study is the first machine psychology experiment conducted within the context of software effort estimation.

## 3 The ORIGINAL STUDY

The ORIGINAL STUDY [Jørgensen and Halkjelsvik 2010] aims to investigate the impact of transitioning from the traditional request format ("*How much effort is required to complete X?*) to the alternative request format ("*How much can be completed in Y work-hours?*"). The alternative format might be more relevant due to the wide usage of incremental software development methods (e.g., agile methods such as Scrum). ORIGINAL STUDY comprises four controlled between-subject experiments with human participants. The first experiment (STUDY1) is the original attempt to investigate the impact of the change in the request format on software effort estimation. The remaining experiments are replications of STUDY1. The second experiment (STUDY2) examines the robustness of STUDY1 results. The third and fourth experiments (STUDY3 and STUDY4) investigate the underlying mechanisms leading to the changes observed in the first two experiments (STUDY1 and STUDY2), more specifically the "anchoring bias." Below, we explain these four experiments briefly:

**STUDY1** In STUDY1, participants, who are software professionals, are provided with a two-page document containing the experimental task. Figure 1 shows the content of the document ORIGINAL STUDY participants received: The first page includes the task description and the effort estimation request in the traditional format for the control group (i.e., traditional group) and the alternative format for the treatment group (i.e., alternative group) and the second page is a requirements specification for a web-based reservation system to facilitate serving dinner to employees of companies who work late hours and comprises eight user stories. The participants were instructed to base their effort estimates on the average productivity of their company. The experiment findings suggest a statistically significant impact of the request format on effort estimates. Effort estimates of

the alternative group (median: 85 work hours) were lower than the effort estimates of the traditional group (median: 220 work hours) with statistical significance.

**STUDY2** Unlike STUDY1, the participants, who were also software professionals, were instructed to **(i)** assume that they would do all the work themselves and **(ii)** estimate the total effort after predicting the total number of user stories to be completed in the first week rather than the remaining effort. Although the findings were not statistically significant, the alternative group provided lower effort estimates (median: 120 work hours) than the traditional group's estimates (median: 200 work hours).

**STUDY3** Based on STUDY1 and STUDY2 findings, ORIGINAL STUDY hypothesizes that the software professionals in the alternative groups anchor their estimates to the duration of 1 week (30 work hours) mentioned in the first question, and insufficient adjustment leads to lower effort estimates. Therefore, STUDY3 aims to investigate the impact of anchoring bias on low-effort estimates. Unlike the previous two studies, in STUDY3: **(i)** the participants are students enrolled in an introductory psychology course; **(ii)** experimental task comprises reading and taking notes from a book chapter; **(iii)** there are two alternative groups: low-anchor and high-anchor groups, which were supposed to estimate how many pages they could complete in 2 hours and 5 hours, respectively. The findings indicated that the effort estimates of the low-anchor alternative group are lower than those of the high-anchor alternative group with statistical significance.

**STUDY4** ORIGINAL STUDY further investigates the impact of software professionals' anchoring bias on their low effort estimates for the alternative format by conducting STUDY4. Suppose low effort estimates for the alternative request format are due to anchoring to the duration mentioned in the first question (as STUDY3 findings suggest). In that case, we should also observe an increase in effort estimates as the duration increases from low to medium. STUDY4 is a replication of STUDY3, with the following modifications: **(i)** participants were students in a history of philosophy course, and the task was writing by hand 24 pages of a book; **(ii)** participants were supposed to provide an effort estimate to an imaginary friend who would do the reading task; **(iii)** there were two groups: the alternative low-anchor group was supposed to estimate the number of pages completed in the first half an hour and medium-anchor group was supposed to make a similar estimation for the first 2 hours. The findings indicate that the medium-anchor group's estimates (median: 420 minutes) were higher than the low-anchor group's estimates (median: 270 minutes) with statistical significance.

To summarize, the increase in effort estimates of software professionals in the alternative groups as anchor values increase from low to high in STUDY1 and from low to medium in STUDY2 show that software professionals anchor on the duration mentioned in the first question of the alternative request format.

## 4 Methodology

This section explains the preliminary study and present the research questions, followed by the experimental design and generation of the prompts.

### 4.1 Preliminary Study

As a preliminary study, we prepared the prompts as input to GPT-4, GEMINI 1.5 PRO and LLAMA 3.1 (for both traditional and alternative request formats) using the exact tasks in the first two experiments (STUDY1 and STUDY2) of the ORIGINAL STUDY. We executed the prompt for each request format 30 times for each experiment (STUDY1 and STUDY2) using the default settings (`temperature` = 1.0). Our findings follow the same trend as the original experiments: As shown in Table 1, we obtained higher effort estimates for the traditional format (Mann-Whitney-U test: $p < 0.0001$) for all three LLMs (GPT-4, LLAMA 3.1 and GEMINI 1.5 PRO) with large effect sizes (Cliff's $\delta = 1.0000$).

Table 1. Preliminary study results and results of the ORIGINAL STUDY conducted with (human) participants. All effort estimates are median values.

| | STUDY1 effort estimates | | | STUDY2 effort estimates | | |
|---|---|---|---|---|---|---|
| | *traditional* | *alternative* | *Sig.* | *traditional* | *alternative* | *Sig.* |
| GPT-4 | 2,000 | 120 | *** | 750 | 120 | *** |
| LLAMA 3.1 | 1,200 | 150 | *** | 400 | 135 | *** |
| GEMINI 1.5 PRO | 160 | 80 | *** | 320 | 117 | *** |
| humans | 220 | 85 | ** | 200 | 120 | 0.19 |

*significance codes:* '***' $p < 0.001$, '**' $p < 0.01$

For STUDY1, the traditional and alternative format effort estimates of GEMINI 1.5 PRO are comparable with those of the ORIGINAL STUDY conducted with (human) software professionals. On the other hand, the traditional format estimates of GPT-4 and LLAMA 3.1 are at least nine and five times as high as the traditional effort estimates made by (human) software professionals in the ORIGINAL STUDY. Traditional format estimates of GPT-4 and LLAMA 3.1 are higher than GEMINI 1.5 PRO's traditional format estimates, with statistical significance Mann-Whitney-U test: $p < 0.0001$) with large effect sizes (Cliff's $\delta = 1.0000$).

In STUDY2 we observed a decrease in the traditional effort estimates of both GPT-4 and LLAMA 3.1 (Mann-Whitney-U test: $p < 0.0001$) with large effect sizes (Cliff's $\delta = 0.9333$ and Cliff's $\delta = 1.0000$). In contrast, there is an increase in the traditional format effort estimates of GEMINI 1.5 PRO (Mann-Whitney-U test: $p < 0.05$) with small effect sizes (Cliff's $\delta = -0.3200$). When we compare each LLM's alternative format effort estimates for STUDY1 and STUDY2, there is no statistically significant change in the estimates of GPT-4 (Mann-Whitney-U test: $p = 0.1050$). At the same time, we observe a decrease in the alternative format estimates of LLAMA 3.1 (Mann-Whitney-U test: $p < 0.0001$) with large effect size (Cliff's $\delta = 1.0000$)and an increase in GEMINI 1.5 PRO's estimates (Mann-Whitney-U test: $p < 0.0001$) with large effect size (Cliff's $\delta = -0.6956$). Yet, in STUDY2 effort estimates of each LLM, there is still a statistically significant decrease due to transitioning from the traditional request format to the alternative one for all three models (GPT-4, LLAMA 3.1 and GEMINI 1.5 PRO) (Mann-Whitney-U test: $p < 0.0001$) with large effect sizes (Cliff's $\delta = 1.0000$, Cliff's $\delta = 1.0000$ and Cliff's $\delta = 0.9867$). On the other hand, ORIGINAL STUDY does not report a statistically significant decrease in effort estimates due to transitioning from the traditional to the alternative format ($p = 0.19$). We also observe that all three LLMs' alternative format effort estimates approach to the (human) software professionals' effort estimates in the ORIGINAL STUDY.

### 4.2 Research Questions

The main research question in the ORIGINAL STUDY asks whether a change from the traditional format to the alternative format affects effort estimates. Since LLMs are sensitive to changes in the prompt, it is evident that a change in the request format will affect their effort estimates. However, one needs to investigate *how* effort estimates will change (e.g., Do we obtain lower or higher estimates? Or will the change in estimates be random?). Moreover, preliminary study results show that GPT-4 gives lower effort estimates due to the transition from traditional to alternative formats. However, we must investigate to what extent one can generalize such a decrease in LLMs' effort estimates due to transitioning to the alternative format across different requirement specifications from various open-source software projects. Therefore, we formulate our first research question as follows:

**$RQ_1$.** Does a change from the traditional request format "*How much effort is required to complete X?*" to the alternative format "*How much can be completed in Y work-hours?*" lead to *lower* effort estimates by LLMs?

To ensure that LLMs effectively assist software professionals in effort estimations, it is crucial to understand the underlying mechanisms leading to changes in effort estimation. More specifically, we need to understand to what extent such mechanisms show similarities with human judgment. STUDY3 and STUDY4 findings in the ORIGINAL STUDY show that anchoring bias is among the primary causes of software professionals' cognitive biases leading to lower effort estimates when they were provided with the alternative request format. As mentioned in Section 2.1, anchoring bias is among the main reasons of software professionals' inaccurate effort estimates [Mohanani et al. 2020].

Furthermore, as mentioned in Section 2.3 recent studies, which employ classical psychological tests and experiments to probe LLMs' psychological processes, found similarities between LLMs' failures and human cognitive biases, including anchoring bias [Echterhoff et al. 2024; Jones and Steinhardt 2024]. In the preliminary study we conducted with LLMs, the difference between the effort estimates for traditional and alternative formats are at least three times as high as the difference between the results for obtained in STUDY1 of ORIGINAL STUDY for the traditional and alternative formats (see Table 1). Such high difference might be an indication of LLMs' usage of anchoring and adjustment heuristic and manifestation of anchoring bias due to insufficient adjustment for the prompts generated for the alternative format. Therefore, we formulate the research question $RQ_2$ as follows:

**$RQ_2$.** Does *anchoring bias* contribute to the observed change in LLMs' effort estimation due to transition from the traditional request format to the alternative request format?

### 4.3 Experiments

Below we explain the experiments we conducted, mapping them to the RQs they address and showing how they relate to the experiments in the ORIGINAL STUDY study. Table 2 also provides an overview of the experimental setup. We conducted experiments using GPT-4 (128k context window and 1.8 trillion parameters), GEMINI 1.5 PRO (`gemini-pro-001` with 128k context window and 1.5 trillion parameters) and LLAMA 3.1 (`llama-3.1-405b` with 128k context window and 405 billion parameters). During the experiments we set the `temperature` parameter to 0 to increase the replicability of the results and minimize the variance between models' responses.

**LLM-STUDY1** This experiment is the replication of STUDY1 in the ORIGINAL STUDY and addresses $RQ_1$. The treatment is the request format (i.e., traditional vs. alternative). We instruct each LLM (GPT-4, GEMINI 1.5 PRO, LLAMA 3.1) that they are a project manager and the developer(s) in their company will develop the user stories in the requirements specification.

**LLM-STUDY2** LLM-STUDY2 also addresses $RQ_1$ and aims to check the robustness of LLM-STUDY1 findings. LLM-STUDY2 is a replication of STUDY2 in the ORIGINAL STUDY. Therefore, the settings of LLM-STUDY2 are identical to LLM-STUDY1 settings except that we instruct each LLM to **(i)** assume that they would do all the work themselves and **(ii)** estimate the total effort rather than the remaining effort for the alternative request format after predicting the total number of user stories to be completed in the first week.

**LLM-STUDY3** Since LLMs are sensitive to prompt changes, it is crucial to investigate the impact of each change in the prompt. Such investigations can help us develop guidelines to design prompts for accurate effort estimation. Therefore, we conduct LLM-STUDY3 by only making the change **(i)** in

Table 2. Overview of the experiments

| Exp. ID | LLM's role | | task assignment | | alternative request format: effort estimation | | alternative request format: anchor | | |
|---|---|---|---|---|---|---|---|---|---|
| | *project manager* | *developer* | *developer(s)* | *LLM* | *remaining tasks* | *all tasks* | *low* | *medium* | *high* |
| LLM-STUDY1 | ✓ | | ✓ | | ✓ | | ✓ | | |
| LLM-STUDY2 | | ✓ | | ✓ | | ✓ | ✓ | | |
| LLM-STUDY3 | | ✓ | | ✓ | ✓ | | ✓ | | |
| LLM-STUDY4 | ✓ | | ✓ | | | ✓ | ✓ | | |
| LLM-STUDY5 | ✓ | | ✓ | | ✓ | | | ✓ | ✓ |

the LLM-STUDY1 settings (i.e., we instruct GPT-4 to assume that they are a software developer and will do all the development work themselves).

**LLM-STUDY4** This experiment aims to investigate the individual impact of instructing each LLM to estimate the total effort rather than the remaining effort in the alternative request format (for the same reasons why we conduct LLM-STUDY3). LLM-STUDY4 is the last experiment that checks the robustness of LLM-STUDY1 and addresses $RQ_1$.

**LLM-STUDY5** LLM-STUDY5 addresses $RQ_2$ by investigating to what extent one can explain the phenomena observed in LLM-STUDY1 by the manifestation of anchoring bias. In the prompts we use for the alternative request format, the first question asks how many and which of the user stories can be finished in iteration one. We set the iteration length to 1 week in the four experiments mentioned above, similar to the settings in STUDY1 and STUDY2 in the ORIGINAL STUDY. In this experiment, we prepare two prompts for the alternative request format: one for the medium anchor and the other for the high anchor. As iteration length in Scrum is mostly less than a month (4 weeks) [Sutherland 2014], we set the medium anchor to 2 weeks and high anchor to the extreme case, which is four weeks. We will compare the medium- and high-anchor effort estimates with those of low-anchor (1 week) estimates we will have obtained among LLM-STUDY1 results.

Table 3. Distribution of the number of generated prompts among software projects (per treatment per experiment).

| repository | project name | # of prompts |
|---|---|---|
| Hyperledger | Fabric | 14 |
| Mulesoft | Mule | 2 |
| Spring | XD | 72 |
| | TOTAL | 88 |

### 4.4 Preparing the Dataset of Prompts

We prepared our dataset of prompts referring to the experiment task in STUDY1 of the ORIGINAL STUDY (see Figure 1). Similar to STUDY1 experiment task in the ORIGINAL STUDY, each prompt for LLM-STUDY1 starts with a task description followed by the project description and eight user stories. The content of the task descriptions for the traditional and alternative formats are similar to those in the ORIGINAL STUDY. To prepare the user stories, we referred to the TAWOS dataset [Tawosi et al. 2022a], which is a dataset of agile open-source software project issues mined from Jira repositories. Among the issues labelled as "Story" in the TAWOS dataset, we tried to select those that are in line with the definition of a user story (i.e., an informal, general explanation of a software feature

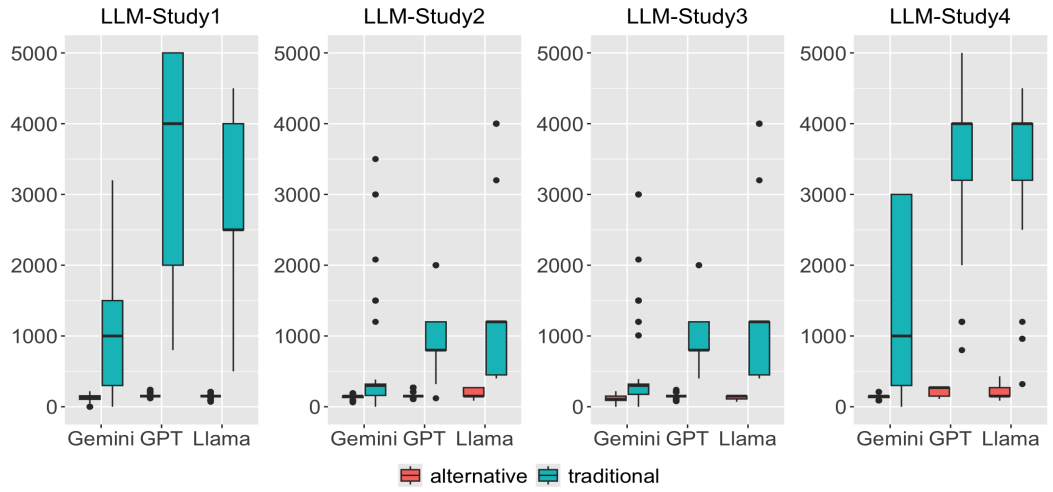



Fig. 2. Box plots comparing effort estimates for *traditional* and *alternative* request formats for each LLM (GPT-4, Llama 3.1 and Gemini 1.5 Pro) for the first four experiments (LLM-STUDY1–LLM-STUDY4).

written from the perspective of the end user or customer [Rehkopf 2020]). Initially, we retrieved stories that comply with the standard user story format "*As a* [*user role*] *I want to* [*functionality*] *so that I can* [*benefit*]," which resulted in 104 user stories. We ordered the retrieved user stories in ascending order according to the date they were created and grouped each of the eight consecutive user stories to include in a single prompt, resulting in 13 prompts. Since 13 prompts do not suffice to obtain statistically significant analyses results, we relaxed the user story selection criteria and retrieved issues labelled as "Story" that start with "As a." We obtained 704 user stories from three open source projects (i.e., Hyperledger Fabric, Mulesoft Mule, Spring XD) as result of our second attempt and prepared 88 prompts for LLM-STUDY1. We made the modifications summarized in Table 2 and explained in Section 4.3 for the remaining four experiments, resulting in 5 x 88 = 440 prompts per treatment, which is 880 prompts in total. Table 3 shows the distribution of 88 prompts (per treatment in each experiment) among the three software projects.

Table 4. Median effort estimates (in work hours) for studies LLM-STUDY1-LLM-STUDY4.

| | GPT-4 | | Llama 3.1 | | Gemini 1.5 Pro | |
|---|---|---|---|---|---|---|
| **Exp. ID** | *traditional* | *alternative* | *traditional* | *alternative* | *traditional* | *alternative* |
| LLM-STUDY1 | 4,000 | 150 | 2,850 | 150 | 1,000 | 135 |
| LLM-STUDY2 | 800 | 150 | 1,200 | 150 | 300 | 150 |
| LLM-STUDY3 | 800 | 150 | 1,200 | 150 | 300 | 105 |
| LLM-STUDY4 | 4,000 | 270 | 4,000 | 150 | 1,000 | 150 |

$p < 0.0001$

## 5 Results

We used the Mann-Whitney U test to compare effort estimates (e.g., traditional vs. alternative format estimates to answer $RQ_1$ and low- vs. medium- vs. high-anchor estimates to answer $RQ_2$) since Anderson-Darling test results provided evidence that none of the three LLMs' effort estimates

were normally distributed. For instance, Anderson-Darling test results indicated that traditional ($A = 7.4261$, $p < 0.0001$) and alternative ($A = 3.0402$, $p < 0.0001$) format effort estimates of GPT-4 for LLM-STUDY1 were not normally distributed. We report all remaining Anderson-Darling test results in our replication package [1].

### 5.1 $RQ_1$ - LLMs' Lower SEE due to Transition from Traditional to Alternative Format

To answer $RQ_1$, we compared traditional format effort estimates with the alternative format effort estimates for each LLM (GPT-4, LLAMA 3.1, GEMINI 1.5 PRO). Below, we explain the first four experiments we conducted to answer $RQ_1$.

**LLM-STUDY1.** We initially conducted LLM-STUDY1, where we instructed each LLM that they were a project manager and the developer(s) in their company would develop the user stories in the requirements specification. As the first row in Table 4 reports, the median LLM-STUDY1 traditional effort estimates for all three LLMs are higher than the corresponding alternative format effort estimates with statistical significance. For GPT-4, median values for the traditional and alternative format estimates are 4000 and 150 work-hours (Mann-Whitney-U test: $p < 0.0001$), respectively with a large effect size (Cliff's $\delta = 1.0000$). The difference between median values of GPT-4's traditional and alternative format estimates are higher than the difference reported for STUDY1 in the ORIGINAL STUDY conducted with (human) software professionals: STUDY1 traditional and alternative effort estimates were 220 and 85 work hours, respectively. The difference between the median values of traditional and alternative format estimates is lower for LLAMA 3.1 (2, 850 vs. 150 work hours), yet statistically significant (Mann-Whitney-U test: $p < 0.0001$) with a large effect size (Cliff's $\delta = 1.0000$). The weakest difference is between GEMINI 1.5 PRO's median traditional and alternative format effort estimates (1, 000 vs. 150 work hours), which is also statistically significant (Mann-Whitney-U test: $p < 0.0001$) with a large effect size (Cliff's $\delta = 0.9321$). To summarize, the difference between median values of traditional and alternative effort estimates remains at least six times higher than the corresponding difference ORIGINAL STUDY reported. Moreover, as Figure 2 depicts, we observe a higher variation among each LLM's traditional format estimates (i.e., the estimates are dispersed) than their alternative format estimates. Among all three LLMs' traditional format effort estimates, the dispersion among GPT-4's effort estimates (spread of effort estimates) is the highest. GPT-4's effort estimate has a median value of 4,000 work hours (IQR 2,000–5,000). At the same time, traditional format effort estimates of LLAMA 3.1 and GEMINI 1.5 PRO have median values of 2,800 work hours (IQR 2,500 – 4,500) and 1,000 (IQR 300 – 1,552), respectively.

> ***Finding 1.*** *Traditional format effort estimates are higher than alternative format estimates for all three LLMs. LLMs' traditional format effort estimates exhibit higher dispersion than their alternative format estimates, where the dispersion among GPT-4's traditional format effort estimates is the highest.*

**LLM-STUDY2.** To test the robustness of our LLM-STUDY1 findings, we conducted LLM-STUDY2 with the following changes: **(i)** we changed LLM's role to developer (from project manager) and instructed LLMs that they will do the software development themselves (change $C_1$); and **(ii)** the prompt in the alternative format asked to estimate the "total effort" rather than the "remaining effort" (change $C_2$). As reported in the second row of Table 4, all three LLMs' traditional format effort estimates are higher than alternative format effort estimates, with statistical significance. The median value for traditional format effort estimate of GPT-4 (800 work hours) is higher than its median alternative

[1] https://zenodo.org/records/14283482

format effort estimate (150 work hours) (Mann-Whitney-U test: $p < 0.0001$) with large effect size (Cliff's $\delta = 0.9814$), while effort estimates of Llama 3.1 and Gemini 1.5 Pro follow a similar trend. Moreover, we observe a decrease in each LLM's traditional format effort estimate compared to the corresponding estimate in llm-study1: The traditional format estimate of gpt-4 decreases from a median value of 4,000 work hours to 800 work hours (Mann-Whitney-U test: $p < 0.0001$) with large effect size (Cliff's $\delta = 0.9500$). Traditional format estimates of Llama 3.1 and Gemini 1.5 Pro follow a similar trend both with large effect sizes (Cliff's $\delta = 0.8310$ and Cliff's $\delta = 0.5790$).

We also observe that the differences between LLMs' traditional and alternative format estimates decrease: The least difference is the one between Gemini 1.5 Pro's traditional and alternative format effort estimates (300 vs. 150 work hours), where traditional estimates are still higher than alternative ones (Mann-Whitney-U test: $p < 0.0001$) also with a medium–large effect size (Cliff's $\delta = 0.5863$). Yet, such a difference is comparable with the difference between (human) software professionals' traditional and alternative effort estimates in study2 of the original study (200 vs. 120 work hours). Moreover, as Figure 2 depicts, we observe a decrease in the dispersion among traditional format effort estimates of each LLM (gpt-4, Llama 3.1, Gemini 1.5 Pro) compared to the variations in the estimates observed in llm-study1. The largest dispersion is for Llama 3.1 (IQR 450 − 1,200), followed by gpt-4 (IQR 800 − 1,200). The spread of traditional format effort estimates is the smallest for Gemini 1.5 Pro with a median value of 300 work hours (IQR 160 − 320). Furthermore, compared to llm-study1 findings, there are no statistically significant changes in the alternative format effort estimates of gpt-4 (Mann-Whitney-U test: $p = 0.5200$) or the estimates of Gemini 1.5 Pro (Mann-Whitney-U test: $p = 0.8600$).

> ***Finding 2.*** *Traditional format effort estimates are still higher than the alternative effort estimates after changing LLMs' role to "developer" (from "project manager") and instructing to estimate the "total effort" (rather than the "remaining effort"). There is also a decrease in traditional format estimates and the estimates' dispersion compared to llm-study1 traditional format estimates.*

**llm-study3.** To investigate the individual impact of the role ("project manager" vs. "software developer") assigned to an LLM, we conducted llm-study3 and we did not observe any statistically significant changes in the traditional format effort estimates of gpt-4 (Mann-Whitney-U test: $p = 0.0510$), Llama 3.1 (Mann-Whitney-U test: $p > 0.99$) or Gemini 1.5 Pro (Mann-Whitney-U test: $p = 0.2850$) compared to llm-study2 findings. Yet, we observed slight changes in the alternative format effort estimates: As reported in the third row of Table 4, median values for gpt-4's and Llama's effort estimates stay the same, while the median value for Gemini 1.5 Pro's alternative format effort estimate decreases from 150 to 105 work hours (Mann-Whitney-U test: $p < 0.01$) with small effect size (Cliff's $\delta = 0.2370$). Such a decrease implies an increase in the minimum difference between traditional and alternative format estimates among all three LLMs from 300 vs. 150 work hours to 300 vs. 150 work hours. We can observe the impact of the change in LLMs' role from "project manager" to developer" (who is supposed to do the development of the user stories provided in the prompts) by comparing llm-study3 findings with the findings of llm-study1: We observe a decrease in the traditional format effort estimates with statistical significance together with a reduction in the spread of effort estimates: gpt-4's median traditional format effort estimates decreased to 800 work hours (IQR 800 −1,200) compared to the llm-study1 traditional format estimates with a median value of 4,000 (IQR 2,000 − 5,000) with statistical significance (Mann-Whitney-U test: $p < 0.00001$) with large effect size (Cliff's $\delta = 0.9660$). Similarly, Llama 3.1's

traditional format effort estimates for LLM-STUDY3, which have a median value of 1,200 work hours (IQR 450 – 1,200), are lower compared to the corresponding LLM-STUDY1 findings with a median value of 2,850 work hours (IQR 2,500 – 4,500) (Mann-Whitney-U test: $p < 0.00001$) with large effect size (Cliff's $\delta = 0.9660$). Moreover, we observe a decrease in GEMINI 1.5 PRO's traditional format effort estimates to a mean value of 300 work hours (IQR 175 – 320) from its LLM-STUDY1 traditional format estimates, where we obtained a mean value of 1,000 work hours (IQR 300 – 1552) Mann-Whitney-U test: $p < 0.00001$) with large effect size (Cliff's $\delta = 0.4970$). Regarding alternative effort estimates, in LLM-STUDY3, we observe that there is a decrease in GPT-4's effort estimates with a median of 120 work hours compared to 150 work hours in LLM-STUDY1Mann-Whitney-U test: $p < 0.05$) and small effect size (Cliff's $\delta = 0.1490$). Although we do not observe the change in the median values of LLAMA 3.1's and GEMINI 1.5 PRO's alternative format effort estimates compared to LLM-STUDY1, we observe a statistically significant difference (Mann-Whitney-U test: $p < 0.00001$) and (Mann-Whitney-U test: $p < 0.01$) with small effect sizes (Cliff's $\delta = 0.1490$) due to a decrease in data spread of LLAMA 3.1 (IQR 150 – 150) and that of GEMINI 1.5 PRO (IQR 135 – 155).

> ***Finding 3.*** *Having changed LLMs' role to "developer" (from "project manager"), we observe a decrease in traditional format effort estimates and estimates' dispersion (compared to LLM-STUDY1). There is also a decrease in the alternative format estimates' spread (inter-quartile ranges). Yet, traditional format estimates remain higher than alternative format estimates with statistical significance.*

**LLM-STUDY4.** We conducted LLM-STUDY4 to investigate the individual impact of instructing LLMs to estimate the "total effort" rather than the "remaining effort" in the prompt for the alternative request format (change $C_2$). Since the change $C_2$ was applied only to the prompts prepared for the alternative request format, the traditional format prompts remained the same as those in LLM-STUDY1. Therefore, as reported in the fourth row of Table 4 and depicted in Figure 2, we did not observe statistically significant changes in LLM-STUDY4 effort estimates: There is no statistically significant change in the effort estimates of GPT-4.0 (Mann-Whitney-U test: $p = 0.5990$), LLAMA 3.1 Mann-Whitney-U test: $p = 0.3490$) or GEMINI 1.5 PRO (Mann-Whitney-U test: $p = 0.5090$). Regarding alternative format estimates, GPT-4's median estimates increased to 270 work hours from 150 work hours in LLM-STUDY1 (Mann-Whitney-U test: $p < 0.00001$) with medium–large effect size (Cliff's $\delta = -0.5990$). We also observe an increase in LLAMA 3.1's alternative effort estimates although median values remained the same (Mann-Whitney-U test: $p < 0.00001$) with medium–high effect size (Cliff's $\delta = -0.3930$). Although there is an increase in GEMINI 1.5 PRO's alternative format estimates to a median value of 150 work hours from (135 work hours in LLM-STUDY1 findings), such a change is not statistically significant (Mann-Whitney-U test: $p = 0.0608$).

> ***Finding 4.*** *Changing alternative format prompt instructions to estimate the "total effort" (rather than the "remaining effort"), we observed an increase in GPT-4's and LLAMA 3.1's alternative format estimates.*

### 5.2 $RQ_2$ - Contribution of Anchoring Bias to the Observed Change in LLMs' SEE

ORIGINAL STUDY showed that (human) software professionals in the alternative treatment group manifest anchoring bias: They anchor software effort estimates to 30 work hours (1 week) and end

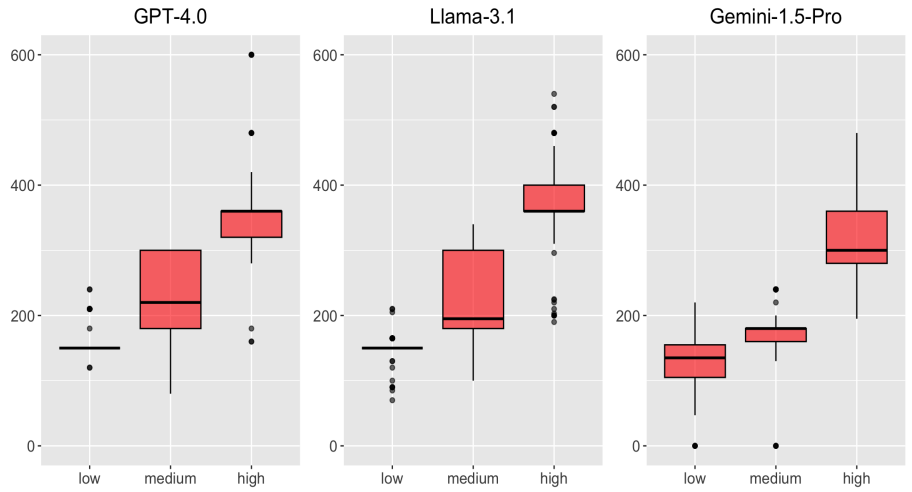



Fig. 3. Box plots comparing effort estimates for "low," "medium" and "high" anchor *alternative* request formats for each LLM (GPT-4, LLAMA 3.1 and GEMINI 1.5 PRO) for LLM-STUDY5).

up with lower effort estimates due to insufficient adjustment away from the anchor. As empirical evidence to support their findings, ORIGINAL STUDY showed that alternative format effort estimates increase as the value of the anchor increases and stay close to the anchors simultaneously. Hence, in the ORIGINAL STUDY, the alternative format estimates remained lower than traditional format estimates while exhibiting an upward trend with the increasing values of the anchors.

To answer $RQ_2$, we selected "low" (1 week = 30 work hours), "medium" (2 weeks = 60 work hours), and "high" (4 weeks = 120 work hours) anchors and investigated whether: **(i)** there is an increase in the alternative format effort estimates as we increase the time frame mentioned in the prompt from 30 work hours to 60 work hours and then to 120 work hours; **(ii)** alternative format estimates remain lower than the corresponding traditional format estimate for all three (low, medium, and high anchors). We used the low anchor effort estimates of LLM-STUDY1 in the statistical analyses since experimental settings of both studies (LLM-STUDY1 and LLM-STUDY5) are identical except that LLM-STUDY5 uses medium and high anchors instead of low anchor as shown in Table 2 and explained in Section 4.3.

As depicted in Figure 3 and reported in Table 5, there is an increase in all three LLMs' (GPT-4, LLAMA 3.1 and GEMINI 1.5 PRO) alternative effort estimates as we move from low anchor to medium anchor and then to high anchor. Such an upwards trend is statistically significant based on the pairwise comparisons we conducted using Mann-Whitney U Test with Rom's method. GPT-4's alternative effort estimates for medium anchor (median = 220 work hours) are higher than its low-anchor effort estimates (Mdn = 150 work hours) (Mann-Whitney-U test: $p < 0.00001$) with large effect size (Cliff's $\delta = 1.0000$) and lower than its high anchor effort estimates (Mdn = 360 work hours) textcolorred(Mann-Whitney-U test: $p < 0.00001$) with large effect size (Cliff's $\delta = -0.9286$). As Table 5 reports, such upwards trend also exists in LLAMA 3.1's and GEMINI 1.5 PRO's alternative effort estimates due to transitioning from low to medium (Mann-Whitney-U test: $p < 0.00001$) with large effect sizes (Cliff's $\delta = 1.0000$ and Cliff's $\delta = 0.8435$) and then to high anchors (Mann-Whitney-U test: $p < 0.00001$) with large effect sizes (Cliff's $\delta = 0.9970$ and Cliff's $\delta = -0.9872$).

To investigate whether each LLM's anchored effort estimates stay lower than the corresponding traditional format effort estimates, we compared each of the low-, medium- and high-anchor effort

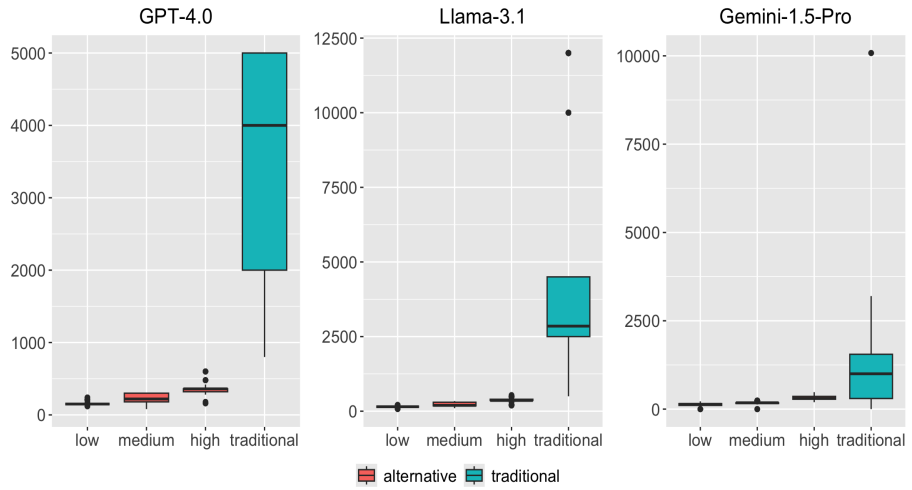



Fig. 4. Box plots comparing effort estimates for *traditional* request format, and estimates for *alternative* request formats with "low," "medium" and "high" anchors for each LLM (GPT-4, Llama 3.1 and Gemini 1.5 Pro) in LLM-STUDY5.

estimates of that LLM with its traditional format estimate also by using Mann-Whitney U test with Rom's method. Our statistical analyses results confirm what Figure 4 depicts: low-, medium- and high-anchor effort estimates are lower than the corresponding traditional format estimate with statistical significance. We used the traditional effort estimates obtained in LLM-STUDY1 in our analyses since both studies LLM-STUDY1 and LLM-STUDY5 have identical settings for traditional format effort estimates. GPT-4 has the highest difference between traditional effort estimates and low-, medium- and high-anchor effort estimates: median value of the traditional effort estimate is 4,000 work-hours, whereas high-anchor estimate's median value is 360 work hours (Mann-Whitney-U test: $p < 0.00001$) with large effect size (Cliff's $\delta = 1.0000$). The difference between Llama 3.1's traditional effort estimate (Mdn = 2,850 work hours) and high-anchor effort estimate (Mdn = 360 work hours) is also statistically significant (Mann-Whitney-U test: $p < 0.00001$) with large effect size (Cliff's $\delta = 1.0000$). The gap between Gemini 1.5 Pro's traditional effort estimate (Mdn = 1,000 work hours) and high-anchor effort estimate (Mdn = 300 work hours) is the lowest among all three LLMs, yet results are statistically significant (Mann-Whitney-U test: $p < 0.00001$) with large effect sizes (Cliff's $\delta = 0.5560$).

Table 5. Effort estimates (median values) for "low," "medium" and "high" anchors of alternative format with anchoring indices (AI $_{l \to m}$ and AI $_{m \to h}$) vs. effort estimate for the traditional format for each LLM (GPT-4, Llama 3.1, Gemini 1.5 Pro)

| | **effort estimates** | | | | | |
|---|---|---|---|---|---|---|
| | | *alternative* | | | | |
| **LLMs** | *traditional* | low | medium | high | AI $_{l \to m}$ | AI $_{m \to h}$ |
| GPT-4 | 4,000 | 150 | 220 | 360 | 3.50 | 2.17 |
| Llama 3.1 | 2,850 | 150 | 195 | 360 | 1.50 | 4.00 |
| Gemini 1.5 Pro | 1,000 | 135 | 180 | 300 | 1.50 | 2.00 |

Another observation providing an evidence for each LLM's anchoring is that low-, medium- and high-anchor effort estimates' interquartile-ranges are far smaller than the interquartile range of the corresponding traditional format estimate. GPT-4 and LLAMA 3.1 mostly give an effort estimate of 150 work hours, hence having a mean of 150 work hours (IQR 150 – 150); whereas GPT-4's medium-anchor effort estimates has the highest interquartile range among effort estimates obtained for LLM-STUDY5 with a median value of 220 work hours (IQR 180 – 300), which is followed by llama's medium anchor effort estimates with a median value of 195 work hours (IQR 180 – 300). However, as Figure 4 shows, the mentioned alternative format effort estimates are quite low, compared to the spread of corresponding traditional format effort estimates. For instance, GEMINI 1.5 PRO has the traditional format effort estimates with the smallest interquartile range with a median value of 1,000 work hours (IQR 300 – 1,552), whereas its high-anchor effort estimates has the highest spread among all three alternative format effort estimates with a median of 300 work hours (IQR = 280 – 360).

**Measuring Anchoring Bias.** We also calculated the anchoring indices. Anchoring index (AI) is used to measure the effect of anchoring in psychology literature [Kahneman 2011] (see Section 2.1). For instance, the difference between GPT-4's median effort estimates for low and medium anchors is $\Delta_{outcome} = 220 - 150 = 70$ work hours and the difference between low and medium anchors is $\Delta_{anchors} = 60 - 30 = 30$ work hours. Then the anchoring index becomes $AI_{l \to m} = \Delta_{outcome}/\Delta_{anchors} = 3.5$. The range of AI was generally 0–1 [Kahneman 2011]. AI = 0 indicates that there is no anchoring effect; whereas $AI = 1$ indicates that there is a strong anchoring effect. Moreover, AI > 1 indicates that there is an extremely significant anchoring effect [Zong and Guo 2022]. Similarly, transition from 60 work hours (medium anchor) to 120 work hours (high anchor) has an anchoring effect of $\Delta_{outcome}/\Delta_{anchors} = 2.17$, where $\Delta_{outcome} = 350 - 220 = 130$ work hours and $\Delta_{anchors} = 120 - 60 = 60$ work hours. Table 5 summarizes the AI values for LLAMA 3.1 and GEMINI 1.5 PRO in addition to GPT-4's AI values. In Table 5, $AI_{l \to m}$ stands for anchoring index measured for the transition from low to medium anchors; $AI_{m \to h}$ stands for anchoring index measured for the transition from medium to high anchors.The AI values in Table 5

> ***Finding 5***. *There is empirical evidence supporting LLMs' anchoring: Alternative format effort estimates increase while transitioning from low- to medium- and then to high-anchor . Yet, alternative format effort estimates remain lower than the traditional effort estimates. Anchoring index (AI) values observed also indicate the existence of highly significant anchoring effects.*

### 5.3 Variability among LLMs' Estimates

As mentioned in section 5.1, there is variability among traditional format effort estimates across models (GPT-4, GEMINI 1.5 PRO, LLAMA 3.1) for all four studies (LLM-STUDY1–LLM-STUDY4). For instance, for LLM-STUDY1, the traditional format efforts estimates of GEMINI 1.5 PRO differ from those of GPT-4 (Mann-Whitney-U test: $p < 0.0001$) and LLAMA 3.1 (Mann-Whitney-U test: $p < 0.0001$) with large effect sizes (Cliff's $\delta = 0.7686$ and Cliff's $\delta = 0.7151$). However, we do not observe a statistically significant difference among GPT-4 and LLAMA 3.1 effort estimates (Mann-Whitney-U test: $p = 0.6404$).

The variability among all three models' estimates diminishes or decreases with smaller effect sizes due to transitioning from traditional to the alternative format. For instance, in LLM-STUDY1, alternative format estimates of GEMINI 1.5 PRO also differ from those of GPT-4 (Mann-Whitney-U test: $p < 0.01$) but with a small effect size (Cliff's $\delta = 0.2291$); whereas differences among alternative format estimates of GEMINI 1.5 PRO and LLAMA 3.1 diminish (Mann-Whitney-U test: $p = 0.0995$).

We do not observe a statistically significant difference among alternative format effort estimates of GPT-4 and those of LLAMA 3.1 (Mann-Whitney-U test: $p = 0.1591$). However, for the remaining three studies (LLM-STUDY2–LLM-STUDY4), we observe differences among alternative format estimates of all three models but with smaller effect sizes compared to those of the corresponding traditional format estimates. For instance, for LLM-STUDY2, alternative effort estimates of GEMINI 1.5 PRO differ from those of GPT-4 (Mann-Whitney-U test: $p < 0.01$) with small effect size ( Cliff's $\delta = 0.2487$) and LLAMA 3.1 GPT-4 (Mann-Whitney-U test: $p < 0.0001$) with larger effect size (Cliff's $\delta = 0.4672$). We also observe a difference among alternative efforts estimates of GPT-4 and LLAMA 3.1 (Mann-Whitney-U test: $p < 0.05$) with small effect size (Cliff's $\delta = -0.2027$). We report detailed findings of the remaining three studies (LLM-STUDY2–LLM-STUDY4) in the replication package [2].

***Finding 6***. *Traditional format estimates of GEMINI 1.5 PRO differ from those of GPT-4 and LLAMA 3.1 for all four studies (LLM-STUDY1–LLM-STUDY4) with large effect sizes. Due to transitioning from the traditional to the alternative format, we observe differences among all three LLMs' estimates with smaller effect sizes for studies LLM-STUDY2–LLM-STUDY4 and no statistically significant differences for LLM-STUDY1.*

## 6 Threats to Validity

**Construct Validity.** The content and number of user stories in the generated prompts are inline with the task load software practitioners usually plan to complete in a sprint during widely used iterative software development frameworks such as Scrum. We used user stories from existing open-source software projects and defined them in the prompts as new features to be developed for these projects. While analysing LLM-STUDY5 results together with the LLM-STUDY1 results for the alternative format to answer $RQ_2$, we used the "anchoring index" to measure the magnitude of anchoring to investigate the extent to which anchoring bias contributed to the observed change in LLMs' effort estimation due to transitioning from the traditional format to alternative one.

**Internal Validity.** For each experiment, we made a separate API call to each LLM (GPT-4, GEMMA, LLAMA 3.1) for each prompt in the traditional format and each one in the alternative format. To check for the robustness of LLM-STUDY1 findings, we conducted LLM-STUDY2, LLM-STUDY3, and LLM-STUDY4 to investigate the individual and combined impacts of the following two factors: **(i)** LLM's role (i.e., project manager, or developer who will do the development task) and **(ii)** requesting the effort estimate for the remaining or entire user stories in the alternative format. ORIGINAL STUDY conducted STUDY3 and STUDY4 with participants who were not software professionals and whose tasks were not related to software development, probably since recruiting human participants from specific demographics (software professionals) was challenging. Moreover, ORIGINAL STUDY conducts two experiments to investigate how effort estimates for the alternative format change for low, medium, and high anchors, using two different tasks. However, manifestation of cognitive biases are context-specific [Kahneman and Tversky 1979]: A specific cognitive bias might manifest in a particular task (e.g., effort estimation for a reading task) but not manifest for a different task (e.g., effort estimation for a software development task). In our experiments, the following design decisions helped mitigate threats to internal validity while investigating whether anchoring bias is a main cause of the decrease in effort estimates while transitioning from traditional to alternative formats: **(i)** We kept tasks identical throughout all five experiments; and **(ii)** we conducted a single experiment (LLM-STUDY5) to investigate how effort estimates change for low, medium and high anchors.

[2]https://zenodo.org/records/14283482

**Statistical Conclusion Validity.** We conducted the Anderson-Darling test to test for normality. Since normality assumptions did not hold, we applied the Mann-Whitney U-test for comparisons of effort estimates (e.g., traditional vs. alternative effort estimates). We conducted the Mann-Whitney U-Test together with Rom's method for multiple comparisons, such as the pairwise comparisons of alternative format effort estimates for low, medium, and high anchors to investigate whether effort estimates increase as we switch from low to medium and finally to high anchor.
**External Validity.** In our experiments, we used GPT-4 and Llama 3.1, which machine psychology experiments use to investigate LLMs' cognitive biases [Echterhoff et al. 2024; Macmillan-Scott and Musolesi 2024] and illusions [Hagendorff et al. 2023]. We also used Gemini 1.5 Pro, whose inclusion in machine psychology studies [Harrison 2024] is recommended to investigate LLMs' limitations and capabilities due to their linguistic proficiency and human-like intuitive thinking. Finally, we prepared prompts to estimate the effort needed to develop new features for three open-source software projects (Hyperledger Fabric, Mulesoft Mule, and Spring XD).

## 7 Discussions

### 7.1 Shifting Focus from Cognitive Biases to Heuristics

SEE is a "complex" task [Snowden and Boone 2007] since it aims to predict the most realistic effort required to develop or maintain software based on incomplete, uncertain, and noisy input. "Irregular" environments [Kahneman 2011] where the cause-and-effect relationships are unclear characterize "complex" tasks. Due to changing requirements, variance in team performance, and external dependencies) [O'Connor and Lepmets 2015], as any "complex" task, SEE requires employing heuristics [Gigerenzer and Brighton 2012] that can lead to biases [Kahneman 2011]. Our experiments assume an incremental software development approach at the core of most agile approaches. Yet, our research findings remain relevant regardless of evolving SE practices/estimation methods due to the complex nature of SEE. Moreover, the complex nature of SEE mandates a shift of focus from cognitive biases to identifying the right heuristics to tackle complex tasks.

Gigerenzer indicates that besides humans' cognitive limitations, the task environment is the main reason for their reliance on heuristics [Gigerenzer 2008; Gigerenzer and Brighton 2012]. Gigerenzer [Gigerenzer 2008] also shows that the human mind can achieve better prediction performance by employing simple heuristics that require less information and computation or take less time than more complex formal strategies. Recent discoveries in cognitive psychology can explain the rationale behind the claim that expert-based judgment gives more accurate software effort. Therefore, one should ask: "*In which environments will a heuristic succeed and in which environments will fail?*" If the sample size is small or if there is noise in the data, then simple heuristics are merely to outperform complex strategies that use more information and computation [Gigerenzer and Brighton 2012]. Hence, expert judgment can be more accurate in estimating software effort since data is usually incomplete, sparse, and noisy.

While Kahneman and Tversky define cognitive biases as a deviation of human judgment from a rational norm, often taken as a law of logic or probability, Gigerenzer argues that a biased mind that uses heuristics, which are not identical to formal logic or probability calculus rules, can make more accurate decisions to tackle uncertainty. Gigerenzer shifts the focus from identifying the manifestation of cognitive biases and their consequences to *identifying the building blocks of heuristics as adaptive tools to form new heuristics that can tackle uncertainty in different environments.* The research on cognitive biases in software engineering has focussed on the heuristics and biases program of Kahneman and Tversky, who mainly used the term with a negative connotation [Mohanani et al. 2020]. Software Engineering (SE) researchers mainly focus on identifying cognitive biases and their consequences. A significant number of experiments have investigated cognitive biases in

software effort estimation. Systematic replication of such studies conducted with (human) software professionals using LLMs can give SE researchers clues on which changes in format and content deteriorate and which ones improve LLMs' effort estimation performance. We can also identify the magnitude of such changes' impact through systematic investigation. For instance, according to our findings, changes in LLMs' assigned role ("project manager" vs. "developer") or asking LLMs to calculate the "total effort" rather than "remaining effort" in the alternative format has less impact compared to the anchoring effect the time frames ("1 week," "2 weeks," and "4 weeks") mentioned in the prompts. Replication of further SE experiments conducted with humans can provide us with findings to form an adaptive toolbox of heuristics for more accurate software development effort estimations. We can use such generated heuristics to prepare prompt engineering guidelines software practitioners can follow while using LLMs for effort estimations.

### 7.2 In-Context Learning for Analogy-based SEE

We can use anchoring and adjustment heuristics to our benefit. Our findings indicate a strong influence of the anchoring effect on LLMs' effort estimation. One possible solution is to provide LLMs in the prompts with anchors that can lead to more accurate effort estimations, which is similar to "analogy-based software effort estimation," which (human) software professionals widely employ (i.e., practitioners find previously completed software projects that are similar to the one to be estimated and then derive the estimate from the values of these projects.) Due to the rapid advancements in computing power and data availability, LLMs can employ analogy-based software effort estimation without (cognitive) limitations humans might encounter. For instance, one can employ In-Context Learning (ICL) (or few-shot learning) by providing content from actual software projects such as requirements specs or single user stories together with their actual effort estimates and project descriptions. There is a large amount of data that one can retrieve from open source software repositories as well as existing datasets (e.g., TAWOS dataset [Tawosi et al. 2022a], which contain proxies for actual spent effort (e.g., effort time [Tawosi et al. 2022b] for issues. Moreover, we can employ existing Natural Language Processing (NLP) techniques (in combination with few-shot learning or ICL) [Parry et al. 2024] to retrieve user stories with their proxies of actual effort spent and include them in the prompts. Employing few-shot learning techniques in combination with search-based software effort estimation techniques is another promising direction. As we mentioned in Section 2.2, Tawosi et al. [Tawosi et al. 2023] use Search-Based methods to optimize the number and combination of examples that can improve an LLM's performance in estimating story points (SPs) of user stories. Such improvements are also in line with Gigerenzer's view of generating an adaptive toolbox of heuristics we mentioned in Section 7.1. Moreover, properly using the anchoring and adjustment heuristic can also remove variability in effort estimates across different models. There can be variability in effort estimates across various models. Our findings in Section **??** indicate that variability in models' effort estimates decreases due to anchoring.

### 7.3 Human-LLM Collaboration for SEE

Our findings support that human-LLM collaboration can provide more accurate effort estimates than LLMs' or humans' isolated estimates: LLMs manifest anchoring bias during SEE. Yet, due to their computational power, LLMs' anchoring bias is more mitigable, for instance, by generating prompts (e.g., using ICL) containing similar projects with actual effort spent. Software practitioners can also enter in the prompts project team's details (e.g., number of developers, familiarity with tools/programming languages/frameworks) to improve LLMs' accuracy. Practitioners can use LLMs' output as an "anchor" for their estimations and employ the anchoring and adjustment heuristic to obtain accurate estimates (i.e., refine LLMs' estimate). Practitioners and LLMs can also collaboratively play planning poker [Mahnič and Hovelja 2012]. Yet, conducting future experiments

is crucial to investigate the efficiency and effectiveness of human-LLM collaboration settings for SEE.

### 7.4 Optimism Bias

Our findings also indicate possible manifestation of optimism bias. Optimism bias is the tendency to produce unrealistically optimistic estimates, contributes to software projects exceeding their schedules and budgets [Mohanani et al. 2020]. Investigating optimism bias requires baselines (actual effort spent) for which we used "Effort Time" values in the TAWOS dataset. Although our results, which is in the replication package, provides some indications that alternative format might lead to over-optimistic estimates, they remain inconclusive due to small sample size. Therefore, investigating optimism bias within SEE context requires designing and conducting further experiments.

## 8 Conclusions

In this study, we investigated the impact of request formats on LLMs' effort estimation by replicating an experiment originally conducted with (human) software professionals (i.e., ORIGINAL STUDY). Specifically, we investigated how LLMs' effort estimates change due to the transition from the traditional request format (i.e., "How much effort is required to complete X?") to the alternative request format (i.e., "How much can be completed in Y work hours?").

To this aim, we initially conducted a preliminary study where we prepared the prompts as input to LLMs (GPT-4, LLAMA 3.1, GEMINI 1.5 PRO) using the exact tasks in the ORIGINAL STUDY. Having obtained results in line with the ORIGINAL STUDY findings, we generated 88 software project specifications (requirements specs) from 704 user stories that we retrieved from the TAWOS dataset. We conducted five between-subject experiments (traditional vs alternative request formats as the two treatments), generating 88 prompts per treatment for each experiment, resulting in 880 prompts.

Our findings show that LLMs' responses align with (human) software professionals' responses in the ORIGINAL STUDY: Traditional format estimates are higher than alternative format estimates with statistical significance. In addition, anchoring effects are one leading cause of lower alternative format effort estimates. As empirical evidence, we observed that alternative format effort estimates increase while transitioning from low to medium. Yet, alternative format effort estimates remain lower than the traditional effort estimates. Moreover, high dispersion (larger interquartile ranges) in traditional format effort estimates compared to alternative format effort estimates' spread indicates a significantly strong anchoring effect. High anchoring index values (AI > 1) also verify the existence of high anchoring effects.

Our study is the first "Machine Psychology" study within the context of software effort estimation. Systematic replication of such studies conducted with (human) software professionals using LLMs can give SE researchers clues on which changes in format and content deteriorate and which ones improve LLMs' effort estimation performance. Such clues are crucial for forming prompt engineering guidelines for estimating software effort. Our study is a first step towards such a direction. Our findings also indicate the importance of providing anchors that can lead to more accurate effort estimations (e.g., using in-context learning in combination with NLP and search-based optimization techniques).

## 9 Data Availability

Our experimental materials, analysis scripts, and generated requirements specs are available at Zenodo (https://zenodo.org/records/14283482).

## References


Mohammed Alhamed and Tim Storer. 2022. Evaluation of Context-Aware Language Models and Experts for Effort Estimation of Software Maintenance Issues. In *IEEE International Conference on Software Maintenance and Evolution, ICSME 2022*. IEEE, Cyprus, 129–138. https://doi.org/10.1109/ICSME55016.2022.00020

Asad Ali and Carmine Gravino. 2019. A systematic literature review of software effort prediction using machine learning methods. *J. Softw. Evol. Process.* 31, 10 (2019), 1–25. https://doi.org/10.1002/SMR.2211

Jorge Aranda and Steve Easterbrook. 2005. Anchoring and adjustment in software estimation. In *Proceedings of the 10th European Software Engineering Conference Held Jointly with 13th ACM SIGSOFT International Symposium on Foundations of Software Engineering (ESEC/FSE-13)*. ACM, Lisbon Portugal, 346–355. https://doi.org/10.1145/1081706.1081761

Marcel Binz and Eric Schulz. 2023. Using cognitive psychology to understand GPT-3. *PNAS* 120, 6 (2023), 1–10.

Morakot Choetkiertikul, Hoa Khanh Dam, Truyen Tran, Trang Pham, Aditya Ghose, and Tim Menzies. 2019. A Deep Learning Model for Estimating Story Points. *IEEE Trans. Software Eng.* 45, 7 (2019), 637–656. https://doi.org/10.1109/TSE.2018.2792473

Ishita Dasgupta, Andrew K. Lampinen, Stephanie C. Y. Chan, Hannah R. Sheahan, Antonia Creswell, Dharshan Kumaran, James L. McClelland, and Felix Hill. 2022. Language models show human-like content effects on reasoning tasks. *ArXiv* arXiv:2207.07051 (2022), 1–94.

Jessica Echterhoff, Yao Liu, Abeer Alessa, Julian McAuley, and Zexue He. 2024. Cognitive Bias in High-Stakes Decision-Making with LLMs. *ArXiv* abs/2403.00811 (2024), 1–14.

Nicholas Epley and Thomas Gilovich. 2006. The anchoring-and-adjustment heuristic: Why the adjustments are insufficient. *Psychological science* 17, 4 (2006), 311–318. https://doi.org/10.1111/j.1467-9280.2006.01704.x

Eliane Maria De Bortoli Fávero, Dalcimar Casanova, and Andrey Ricardo Pimentel. 2022. SE3M: A model for software effort estimation using pre-trained embedding models. *Inf. Softw. Technol.* 147 (2022), 106886. https://doi.org/10.1016/J.INFSOF.2022.106886

G. Gigerenzer. 2008. *Rationality of Mortals*. Oxford University Press, New York. https://doi.org/10.1080/17439760.2011.614828

G. Gigerenzer and H. Brighton. 2012. Homo heuristicus: why biased minds make better inferences. *Top Cognitive Science* 19, 4 (2012), 6–16. https://doi.org/10.1111/j.1756-8765.2008.01006.x

Lucas Gren and Richard Berntsson Svensson. 2021. Is it possible to disregard obsolete requirements? a family of experiments in software effort estimation. *Requirements Engineering* 26, 3 (2021), 459–480.

Thilo Hagendorff. 2023. Machine Psychology: Investigating Emergent Capabilities and Behavior in Large Language Models Using Psychological Methods. *ArXiv* abs/2303.13988 (2023), 1–17.

T. Hagendorff, S. Fabi, and Kosinski. 2023. Human-like intuitive behavior and reasoning biases emerged in large language models but disappeared in ChatGPT. *Nat Comput Sci* 3 (2023), 833—-838. https://doi.org/10.1038/s43588-023-00527-x

Rachel M. Harrison. 2024. A Comparison of Large Language Model and Human Performance on Random Number Generation Tasks. *ArXiv* arXiv:2408.09656 (2024), 1–5.

Erik Jones and Jacob Steinhardt. 2024. Capturing failures of large language models via human cognitive biases. In *Proceedings of the 36th International Conference on Neural Information Processing Systems (NIPS '22)*. ACM, New Orleans, 1–15.

M. Jørgensen. 2004. A review of studies on expert estimation of software development effort. *Journal of Systems and Software* 70, 1 (2004), 37–60. https://doi.org/10.1016/S0164-1212(02)00156-5

Magne Jørgensen. 2010. Identification of more risks can lead to increased over-optimism of and over-confidence in software development effort estimates. *Inf. Softw. Technol.* 52, 5 (2010), 506–516. https://doi.org/10.1016/j.infsof.2009.12.002

Magne Jørgensen and Stein Grimstad. 2008. Avoiding Irrelevant and Misleading Information When Estimating Development Effort. *IEEE Softw.* 25, 3 (2008), 78–83. https://doi.org/10.1109/MS.2008.57

Magne Jorgensen and Stein Grimstad. 2011. The Impact of Irrelevant and Misleading Information on Software Development Effort Estimates: A Randomized Controlled Field Experiment. *IEEE Transactions on Software Engineering* 37, 5 (2011), 695–707. https://doi.org/10.1109/TSE.2010.78

M. Jørgensen and T. Gruschke. 2005. Industrial Use of Fromal Software Cost Estimation Models: Expert Estimation in Disguise. In *Proceedings of the International Conference on Evaluation and Assessment in Software Engineering*. ACM, United Kingdom, 426–427.

Magne Jørgensen and Torleif Halkjelsvik. 2010. The effects of request formats on judgment-based effort estimation. *J. Syst. Softw.* 83 (2010), 29–36. https://doi.org/10.1016/j.jss.2009.03.076

Magne Jørgensen, Karl Halvor Teigen, and Kjetil MoløKken. 2004. Better sure than safe? Over-confidence in judgement-based software development effort prediction intervals. *J. Syst. Softw.* 70, 1–2 (2004), 79–93. https://doi.org/10.1016/S0164-1212(02)00160-7

Magne Jørgensen. 2006. The effects of the format of software project bidding processes. *International Journal of Project Management* 24, 6 (2006), 522–528. https://doi.org/10.1016/j.ijproman.2006.04.001

D.. Kahneman. 2011. *Thinking, fast and slow*. Farrar, Girar and Stroux, United States. https://doi.org/10.1007/s00362-013-0533-y

Daniel Kahneman and Amos Tversky. 1979. Prospect Theory: An Analysis of Decision under Risk. *Econometrica* 47, 2 (1979), 263–291.

Erik Løhre and Magne Jørgensen. 2016. Numerical anchors and their strong effects on software development effort estimates. *J. Syst. Softw.* 116, C (2016), 49–56. https://doi.org/10.1016/j.jss.2015.03.015

Olivia Macmillan-Scott and Mirco Musolesi. 2024. (Ir)rationality and cognitive biases in large language models. *Royal Society Open Science* 11 (2024), 19 pages. https://doi.org/doi.org/10.1098/rsos.240255

Viljan Mahnič and Tomaž Hovelja. 2012. On using planning poker for estimating user stories. *Journal of Systems and Software* 85, 9 (2012), 2086–2095. https://doi.org/10.1016/j.jss.2012.04.005

Rahul Mohanani, Iflaah Salman, Burak Turhan, Pilar Rodríguez, and Paul Ralph. 2020. Cognitive Biases in Software Engineering: A Systematic Mapping Study. *IEEE Transactions on Software Engineering* 46, 12 (2020), 1318–1339. https://doi.org/10.1109/TSE.2018.2877759

K. Moløkken and M. Jørgensen. 2003. A review of software surveys on software effort estimation. In *2003 International Symposium on Empirical Software Engineering, 2003. ISESE 2003. Proceedings.* ACM, Rome, Italy, 223–230. https://doi.org/10.1109/ISESE.2003.1237981

Rory V. O'Connor and Marion Lepmets. 2015. Exploring the use of the cynefin framework to inform software development approach decisions *(ICSSP '15)*. Association for Computing Machinery, New York, NY, USA, 97–101. https://doi.org/10.1145/2785592.2785608

Andrew Parry, Debasis Ganguly, and Manish Chandra. 2024. "In-Context Learning" or: How I learned to stop worrying and love "Applied Information Retrieval". In *Proceedings of the 47th ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '24)*. 14–25. https://doi.org/10.1145/3626772.3657842

Max Rehkopf. 2020. User stories with examples and a template. https://www.atlassian.com/agile/project-management/user-stories. Accessed: 2024-08-21.

Federica Sarro, Alessio Petrozziello, and Mark Harman. 2016. Multi-objective software effort estimation. In *Proceedings of the 38th International Conference on Software Engineering (ICSE '16)*. ACM, Austin, TX, USA, 619–630. https://doi.org/10.1145/2884781.2884830

David J. Snowden and Mary E. Boone. 2007. A Leader's Framework for Decision Making. *Harvard Business Review* 85, 11 (2007), 68–76.

K. B. Stanovich. 2008. *What intelligence tests miss: The psychology of rational thought.* Yale University Press, New Heaven.

J. Sutherland. 2014. *The art of doing work twice the time.* Crown Business New York, United States.

Alaina N. Talboy and Elizabeth Fuller. 2023. Challenging the appearance of machine intelligence: Cognitive bias in LLMs and Best Practices for Adoption. *ArXiv* arXiv:2304.01358 (2023), 1–11.

Vali Tawosi, Afnan Al-Subaihin, Rebecca Moussa, and Federica Sarro. 2022a. A Versatile Dataset of Agile Open Source Software Projects. In *2022 IEEE/ACM 19th International Conference on Mining Software Repositories (MSR)*. ACM/IEEE, Pittsburg, PA, USA, 707–711. https://doi.org/10.1145/3524842.3528029

Vali Tawosi, Salwa Alamir, and Xiaomo Liu. 2023. Search-Based Optimisation of LLM Learning Shots for Story Point Estimation. In *Proceedings of Search-Based Software Engineering: 15th International Symposium, SSBSE 2023*. Springer, San Francisco, CA, USA, 123–129. https://doi.org/10.1007/978-3-031-48796-5_9

Vali Tawosi, Rebecca Moussa, and Federica Sarro. 2022b. On the Relationship Between Story Points and Development Effort in Agile Open-Source Software. In *Proceedings of the 16th ACM / IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM '22)*. 183–194. https://doi.org/10.1145/3544902.3546238

A. Tverskey and D. Kahneman. 1978. Judgment under Uncertainty: Heuristics and Biases. *Science* 185, 4157 (1978), 1124–1131.

Jianfeng Wen, Shixian Li, Zhiyong Lin, Yong Hu, and Changqin Huang. 2012. Systematic literature review of machine learning based software development effort estimation models. *Inf. Softw. Technol.* 54, 1 (2012), 41–59. https://doi.org/10.1016/J.INFSOF.2011.09.002

Y. Zong and X. Guo. 2022. An Experimental Study on Anchoring Effect of Consumers' Price Judgment Based on Consumers' Experiencing Scenes. *Frontiers in psychology* 13 (2022), 1–15. https://doi.org/10.3389/fpsyg.2022.794135

Received 2024-09-13; accepted 2025-01-14