

Benchmarking LLM-Based Requirement Quality Evaluation and a Knowledge-Based Prototype for Pushing the Boundary of Trustworthiness

Jannatul Shefa¹, Alejandro Salado², Paul Wach², Joe Gregory², Tom McDermott³, & Taylan G. Topcu¹

¹Socio-technical Systems Engineering Lab, Grado Department of Industrial & Systems Engineering, Virginia Tech

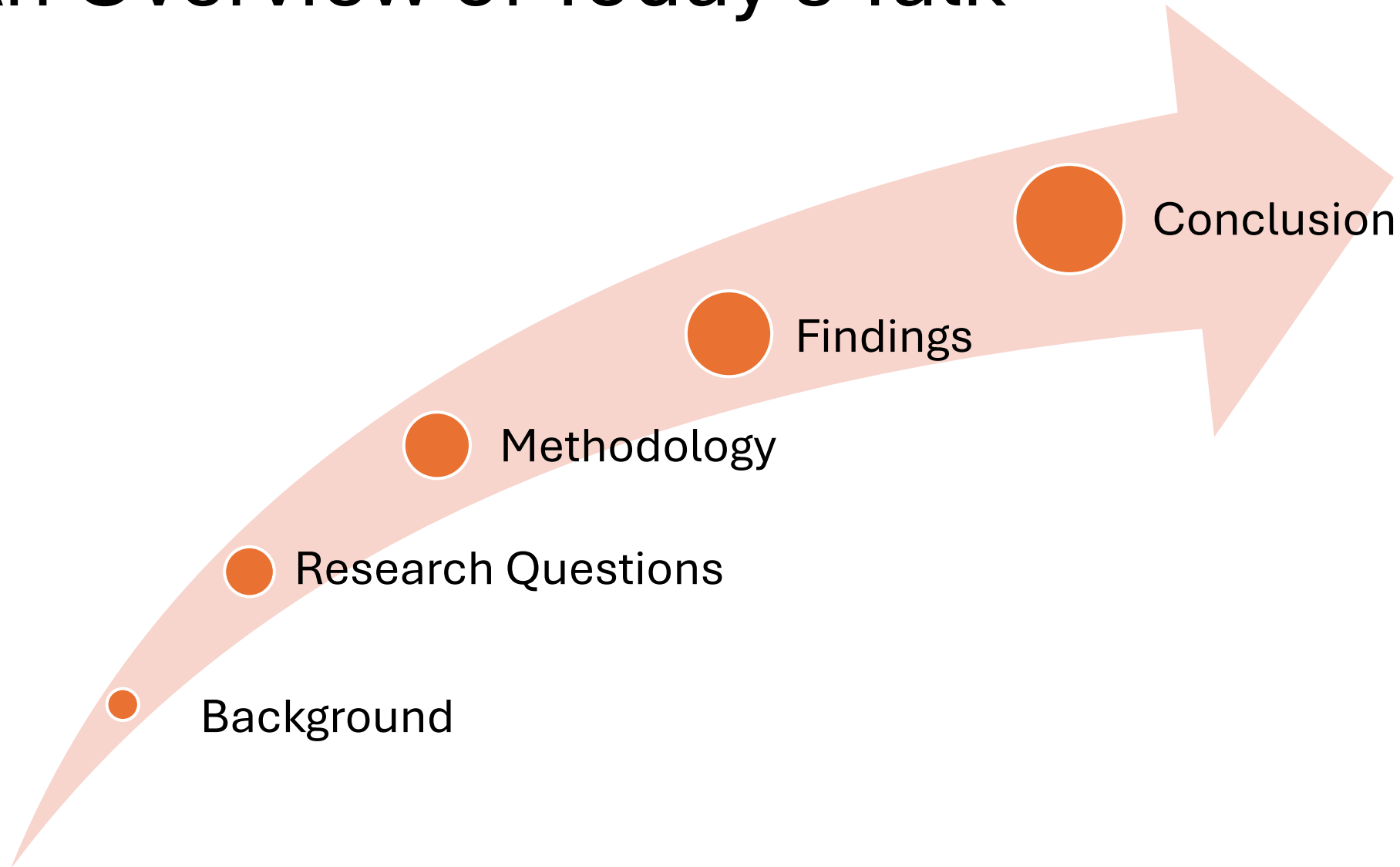
²Systems and Industrial Engineering, The University of Arizona

³Systems Engineering Research Center, Stevens Institute of Technology

Supported by the U.S. National Nuclear Security Administration (NNSA) under SERC Grants WRT 2406 and 2516

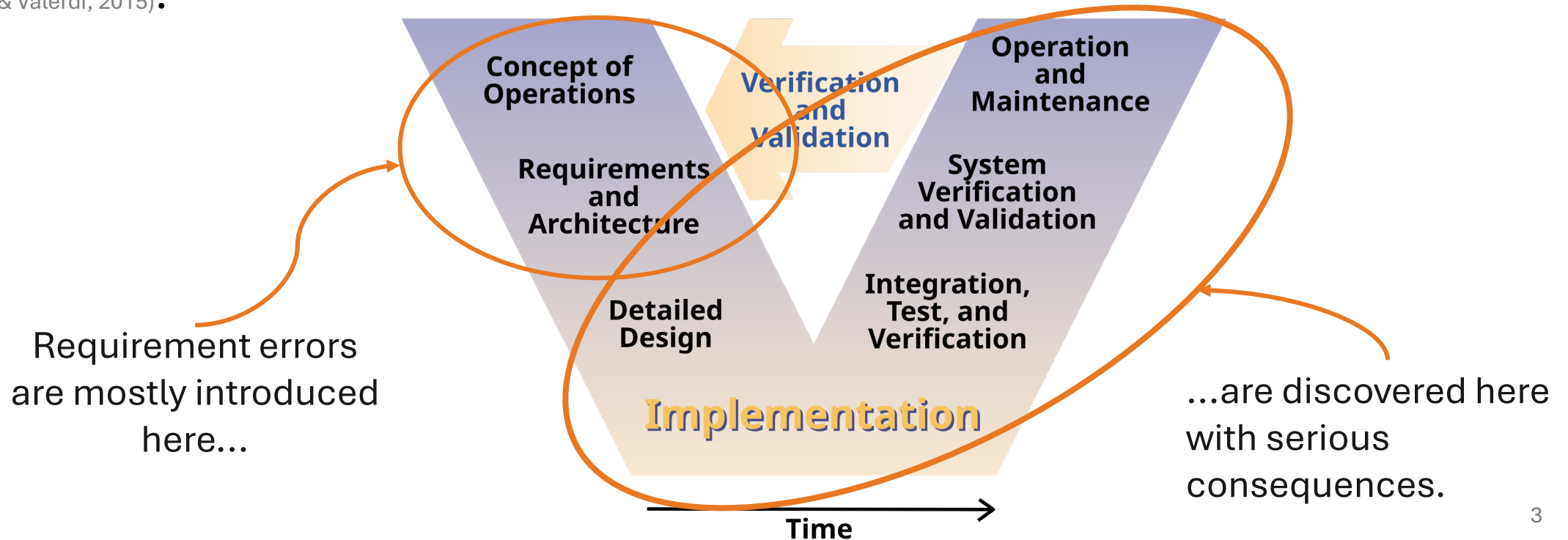


An Overview of Today's Talk



Role of Requirements in SE

- **Requirements** translate the ill-defined problem into a well-defined one that is amenable for the engineering effort (Bahill and Madni, 2017).
- Poor requirements often propagate through the **lifecycle** and are major cause of **design rework, schedule delays, and cost overruns** (Boehm & Papaccio, 1988; Peña & Valerdi, 2015).



The Research Gap

- AI for requirements engineering has evolved from rule-based NLP in the 1980s–90s (Fabbrini et al., 2001) to supervised/unsupervised ML in the 2000s (Kurtanović & Maalej, 2017), transformer-based models in the 2010s (Berquand et al., 2021), and LLMs and agentic AI in the 2020s (Massoudi & Fuge, 2025).

Research GAP: Limited understanding of the performance, error behavior, and variability of AI-assisted Requirements Engineering

systems bounded by physics are more complex and rework is far more costly (Hennig et al., 2022).

evidence on the extent of variability of LLM outputs (Ofsa & Topcu, 2025).

errors) but these are poorly documented (Topcu et.al., 2025).

Research Questions

RQ1: How well does an off-the-shelf LLM perform in evaluating requirement quality?

RQ2: How do error rates vary across INCOSE requirement quality criteria?

RQ3: How does LLM performance evolve over generations?

RQ4: To what extent does infusing expert knowledge improve requirement quality evaluation performance?

Methodology

- Adapted from a NASA Astrobeer challenge [Szajnfarber et al., 2023]
- Implemented in an automated n8n workflow.
- Performance was evaluated in 7 metrics:
 - true positives, true negatives, false positives (Type 1 Error), false negatives (Type 2 error), accuracy, precision, and F1 score.

Findings

Research Questions

RQ1: How well does an **off-the-shelf LLM perform** in evaluating requirement quality?

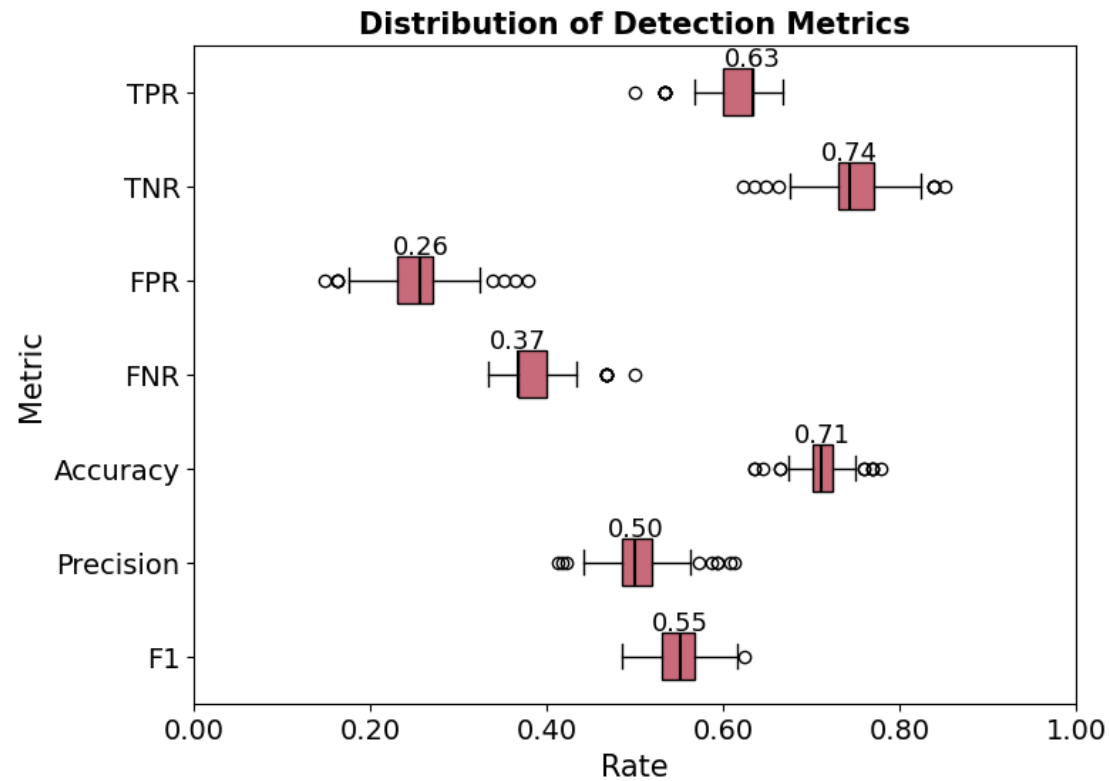
RQ2: How do **error rates vary** across INCOSE requirement quality criteria?

RQ3: How does LLM performance **evolve over generations**?

RQ4: To what extent does **infusing expert knowledge** improve requirement quality evaluation performance?

Can We Trust Off-the-shelf LLMs?

Summary Stats: Claude Opus 5 Baseline



Research Questions

RQ1: How well does an off-the-shelf LLM perform in evaluating requirement quality?

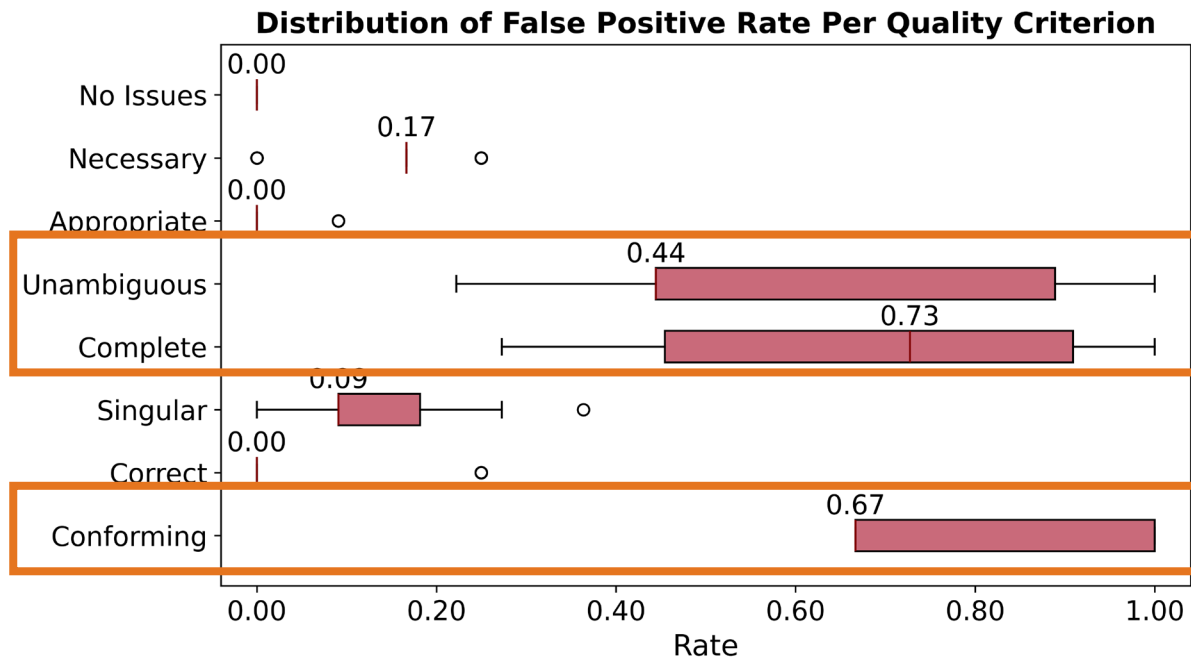
RQ2: How do error rates vary across INCOSE requirement quality criteria?

RQ3: How does LLM performance evolve over generations?

RQ4: To what extent does infusing expert knowledge improve requirement quality evaluation performance?

Can We Trust Off-the-shelf LLMs?

Criteria Specific Error Rates: Claude Opus 5 Baseline

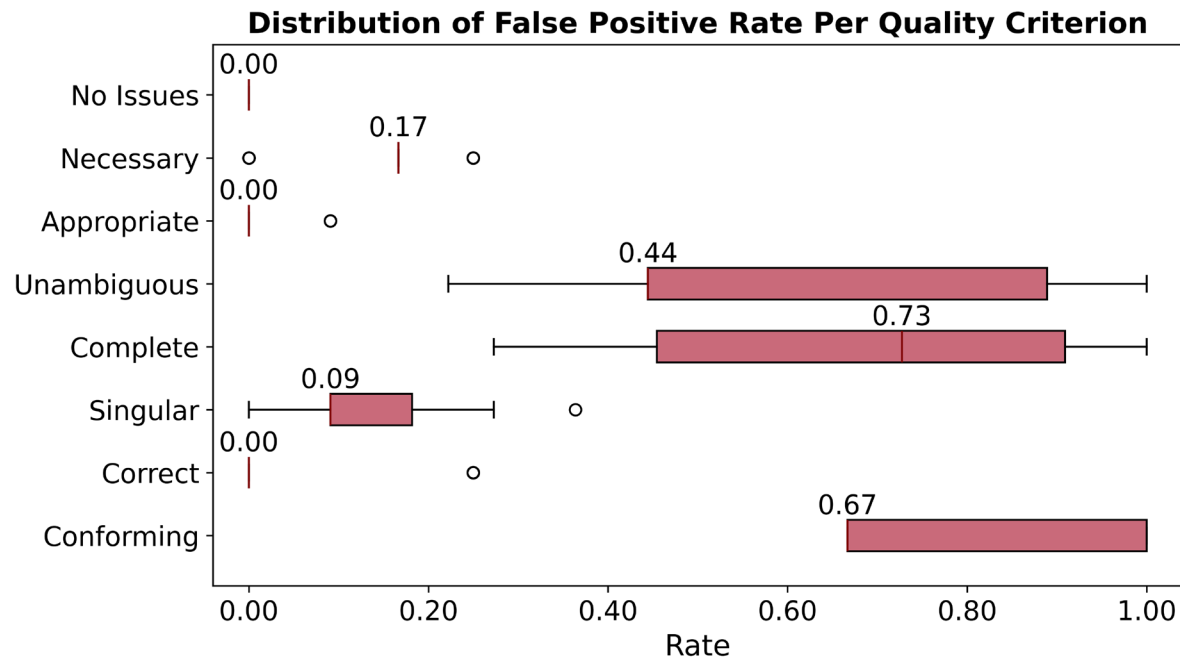


Type 1 Errors or False Alarms

High false alarms in Ambiguity, completeness, and even conformity.

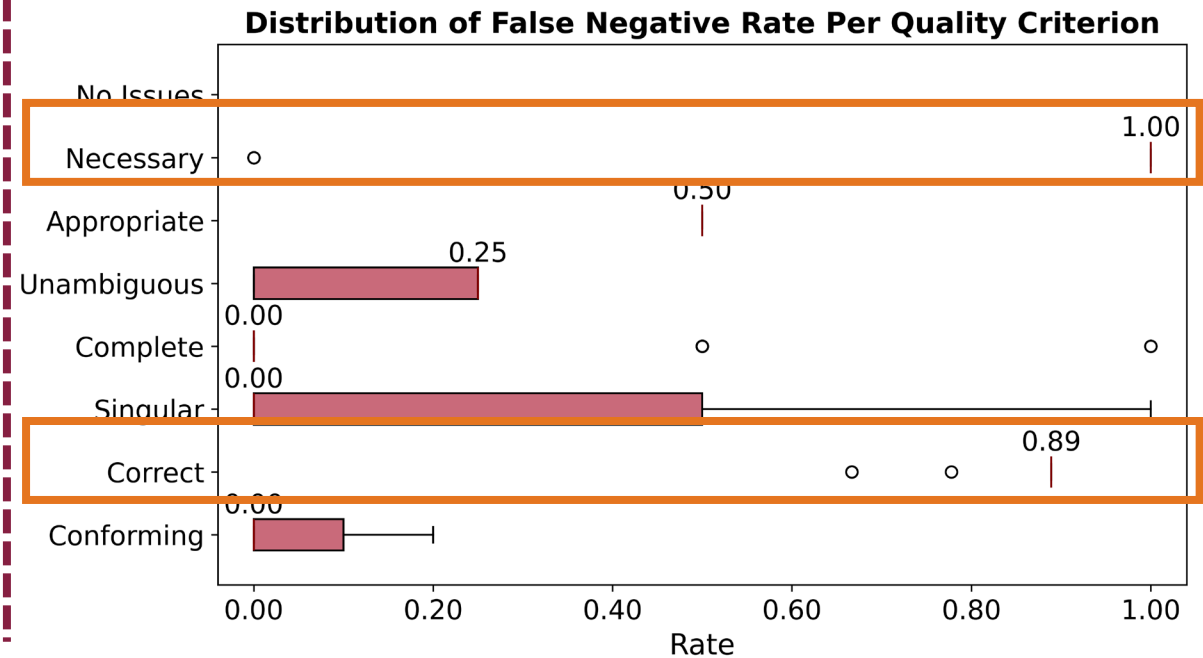
Can We Trust Off-the-shelf LLMs?

Criteria Specific Error Rates: Claude Opus 5 Baseline



Type 1 Errors or False Alarms

High false alarms in Ambiguity, completeness, and even conformity.



Type 2 Errors or Misses

Extremely high miss rate in categories that require SE judgment and contextual reasoning.

Research Questions

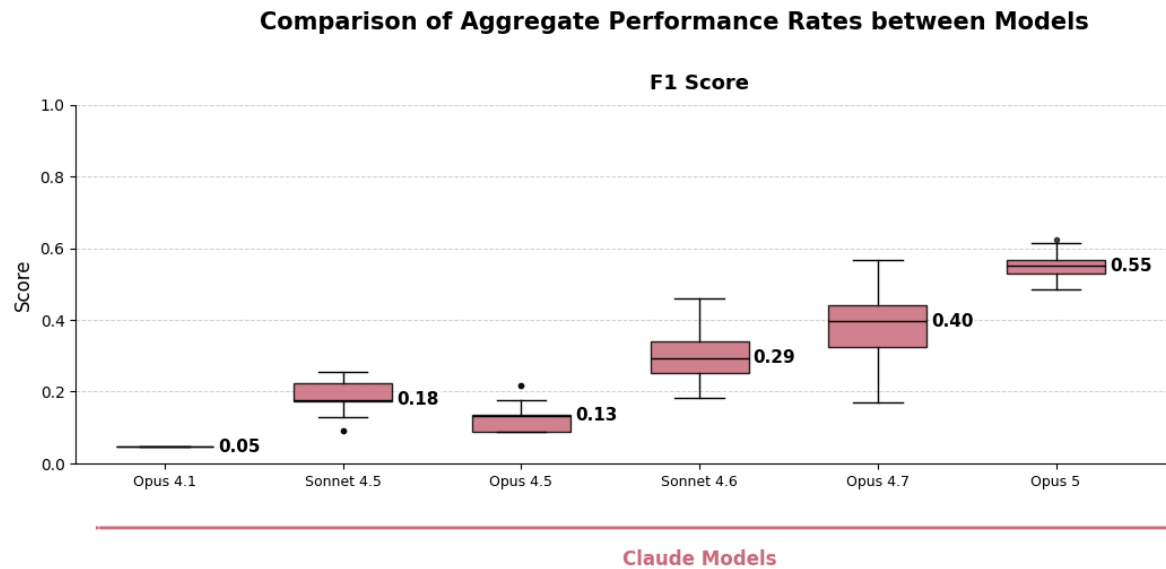
RQ1: How well does an off-the-shelf LLM perform in evaluating requirement quality?

RQ2: How do error rates vary across INCOSE requirement quality criteria?

RQ3: How does LLM performance evolve over generations?

RQ4: To what extent does infusing expert knowledge improve requirement quality evaluation performance?

Generational Trends of models



Is there hope?

Research Questions

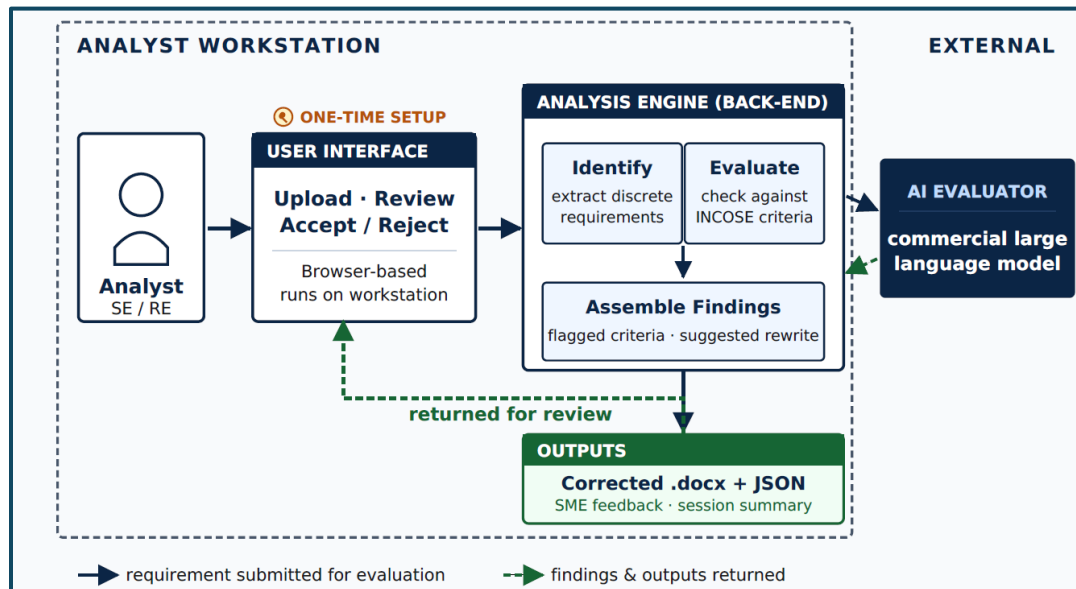
RQ1: How well does an **off-the-shelf LLM perform** in evaluating requirement quality?

RQ2: How do **error rates vary** across INCOSE requirement quality criteria?

RQ3: How does LLM performance **evolve over generations**?

RQ4: To what extent does **infusing expert knowledge** improve requirement quality evaluation performance?

Prototype Beta 1



- **What is it composed of?**

- A React based frontend and FastAPI based Backend.

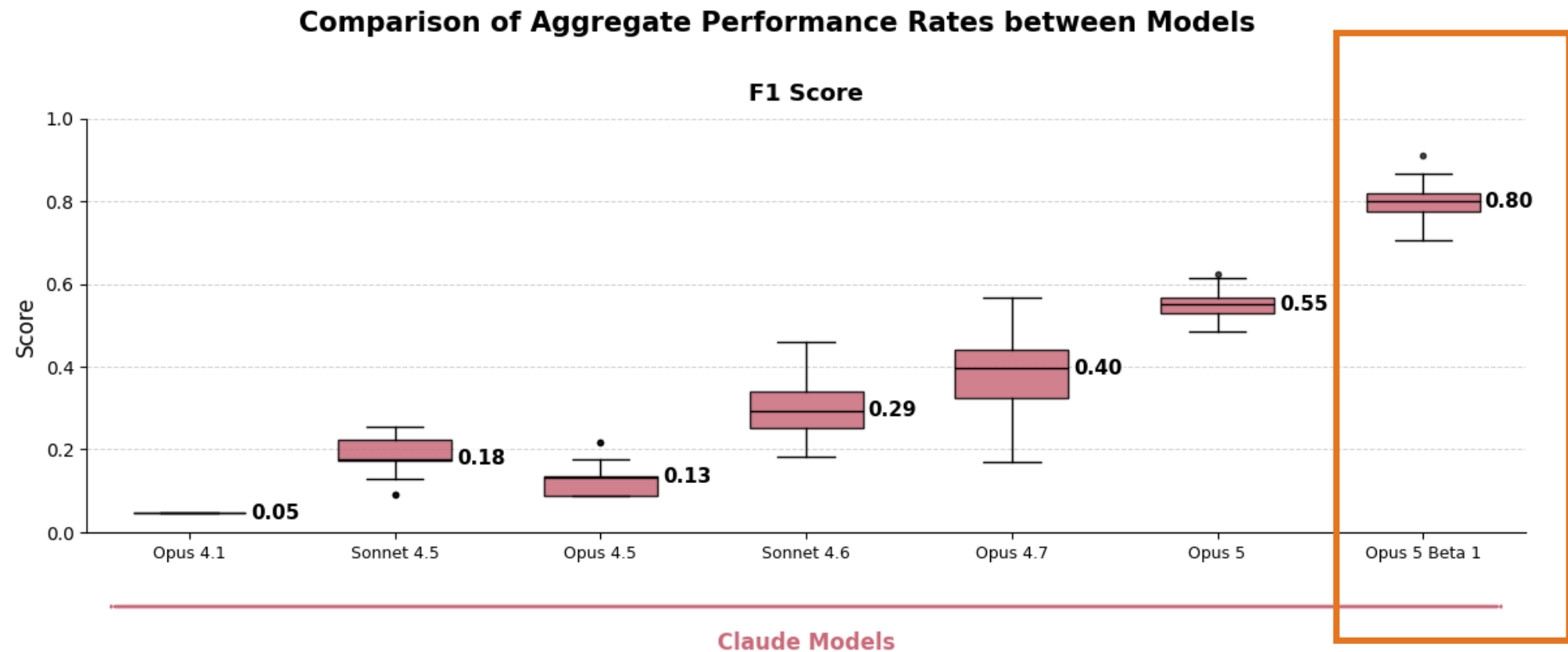
- **Frontend Functionality:**

- receives requirements, system context document (e.g., ConOps), and API key for desired LLM.
- Communicates with back-end, provides recommendations, and captures SME judgements (accept, modify, reject).

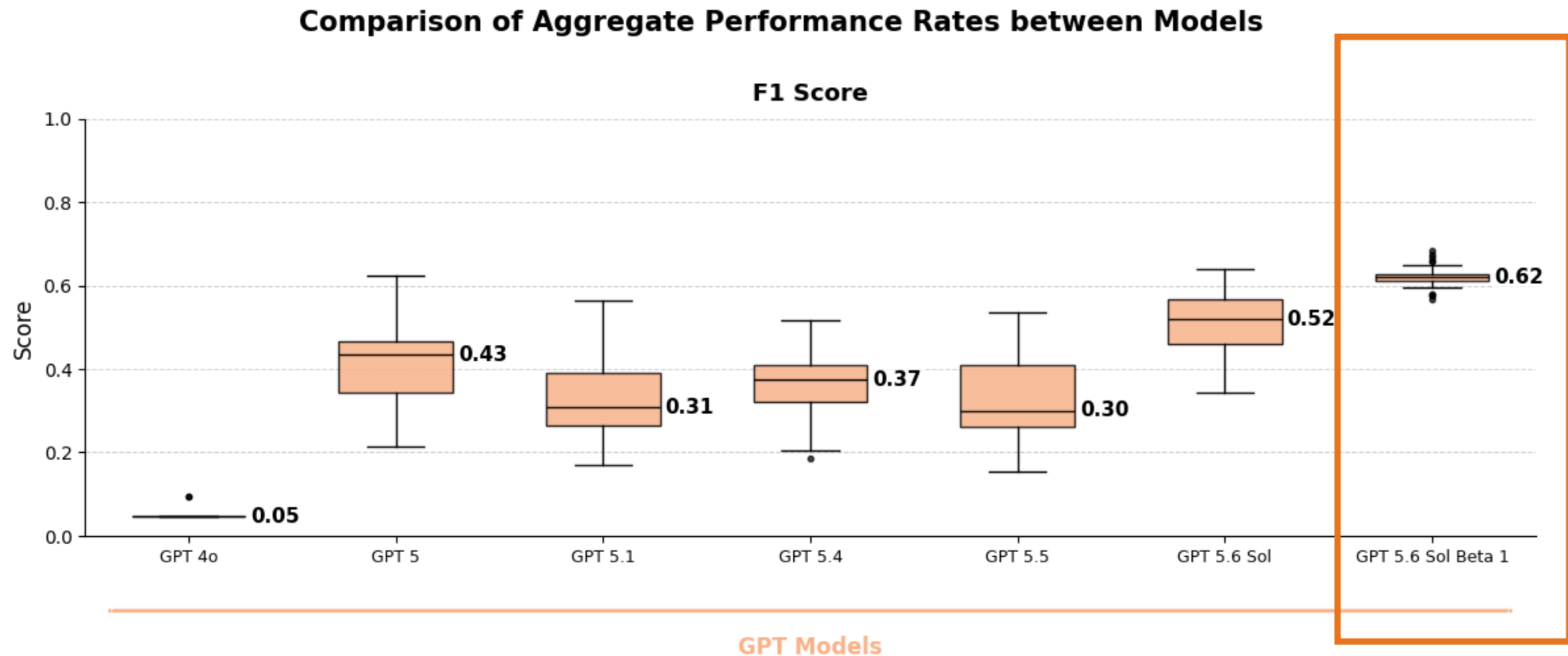
- **Backend Functionality:**

- SE Knowledge Stack and Analysis Engine.
- API that facilitates LLM communication.
- Long-term memory model and Data Export Module.

Generational Claude Models vs Beta 1



Generational GPT Models vs Beta 1



Key Takeaways

- The role of human reviewers is not replaceable.
- A single run can be misleading.
- Performance varies significantly across quality criteria.
- We discussed our prototype and how effectively it can be integrated.

Future Works

- Compare against commercial tools
- Infusing Expert feedback back into the prototype
- Agentic AI approach for criteria-specific intervention
- Set-level requirements quality assessment

Acknowledgment

This work was supported by: The U.S. National Nuclear Security Administration (NNSA) under SERC Grants WRT 2406 and 2516.



Taylan Topcu
Virginia Tech



Paul Wach
University of Arizona



Alejandro Salado
University of Arizona



Joe Gregory
University of Arizona



Tom McDermott
SERC



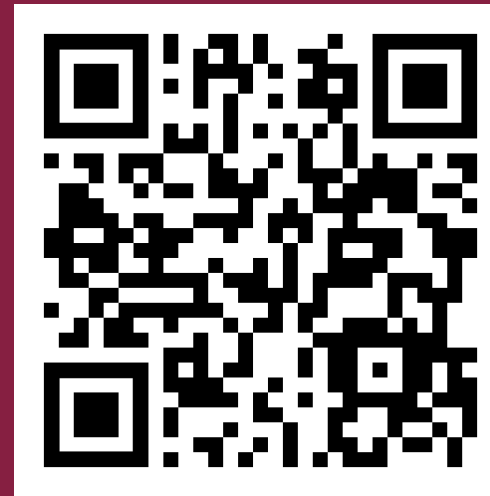
Thank You!

Scan to read more!

Contact us!

Jannatul Shefa: shefa@vt.edu

Dr. Taylan Topcu: ttopcu@vt.edu

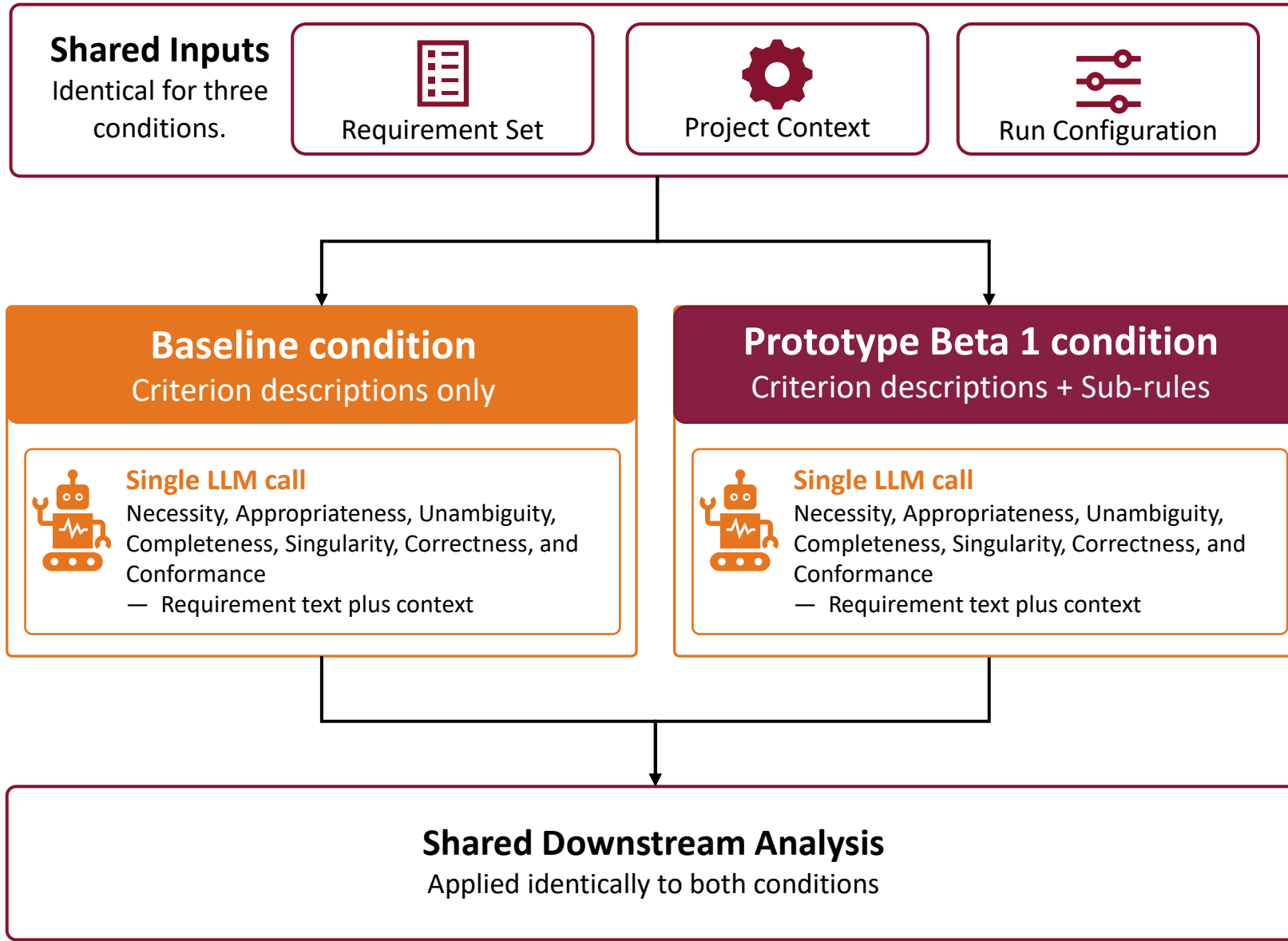


References

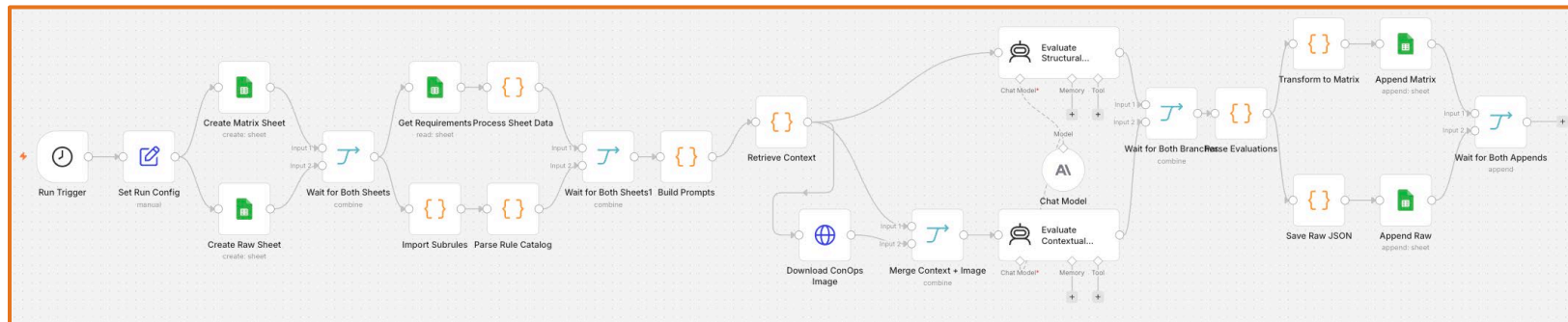
- Bahill, A. T., & Madni, A. M. (2017). Discovering System Requirements. In A. T. Bahill & A. M. Madni, *Tradeoff Decisions in System Design* (pp. 373–457). Springer International Publishing. https://doi.org/10.1007/978-3-319-43712-5_4
- Boehm, B. W., & Papaccio, P. N. (1988). Understanding and controlling software costs. *IEEE Transactions on Software Engineering*, 14(10), Article 10.
- Peña, M., & Valerdi, R. (2015). Characterizing the Impact of Requirements Volatility on Systems Engineering Effort. *Systems Engineering*, 18(1), 59–70. <https://doi.org/10.1111/sys.21288>
- Kurtanović, Z., & Maalej, W. (2017, September). Automatically classifying functional and non-functional requirements using supervised machine learning. In *2017 IEEE 25th international requirements engineering conference (RE)* (pp. 490-495). IEEE.
- Fabbri, F., Fusani, M., Gnesi, S., & Lami, G. (2001). *The linguistic approach to the natural language requirements quality: Benefit of the use of an automatic tool*. 97–105.
- Berquand, A., Darm, P., & Riccardi, A. (2021). Spacetransformers: Language modeling for space systems. *IEEE Access*, 9, 133111–133122.
- Massoudi, S., & Fuge, M. (2025). Agentic large language models for conceptual systems engineering and design. *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 89237, V03BT03A045. <https://asmedigitalcollection.asme.org/IDETC-CIE/proceedings-abstract/IDETC-CIE2025/89237/1226016>
- Hennig, A., Topcu, T. G., & Szajnfarter, Z. (2021). So You Think Your System Is Complex?: Why and How Existing Complexity Measures Rarely Agree. *Journal of Mechanical Design*, 144(4), Article 4. <https://doi.org/10.1115/1.4052701>
- Ofsa, M., & Topcu, T. G. (2025). *An empirical exploration of chatgpt's ability to support problem formulation tasks for mission engineering and a documentation of its performance variability*. 381–395.
- Topcu, T. G., Husain, M., Ofsa, M., & Wach, P. (2025). Trust at Your Own Peril: A Mixed Methods Exploration of the Ability of Large Language Models to Generate Expert-Like Systems Engineering Artifacts and a Characterization of Failure Modes. *Systems Engineering*, sys.21810. <https://doi.org/10.1002/sys.21810>
- Szajnfarter, Z., Hennig, A., Mukherjee, S., Rader, S., & Crusan, J. (2023). Linking solver characteristics, solving processes and solution attributes: A data explainer for an open innovation generated robotic design dataset. *Data in Brief*, 50, 109547.

Extras

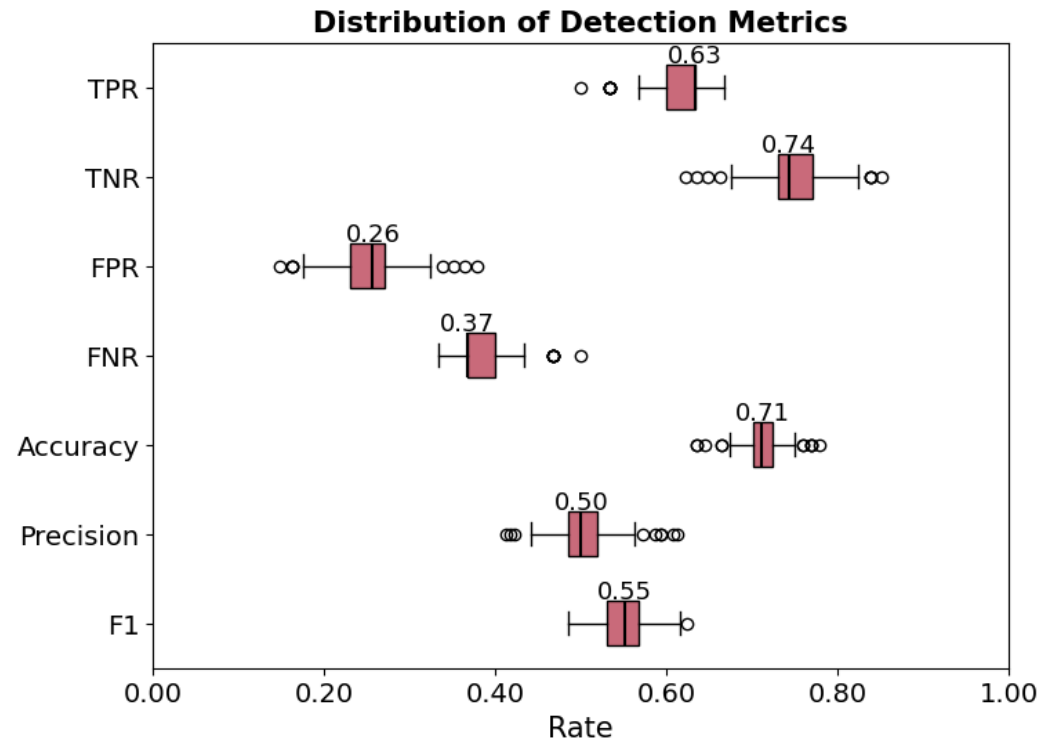
Methodology



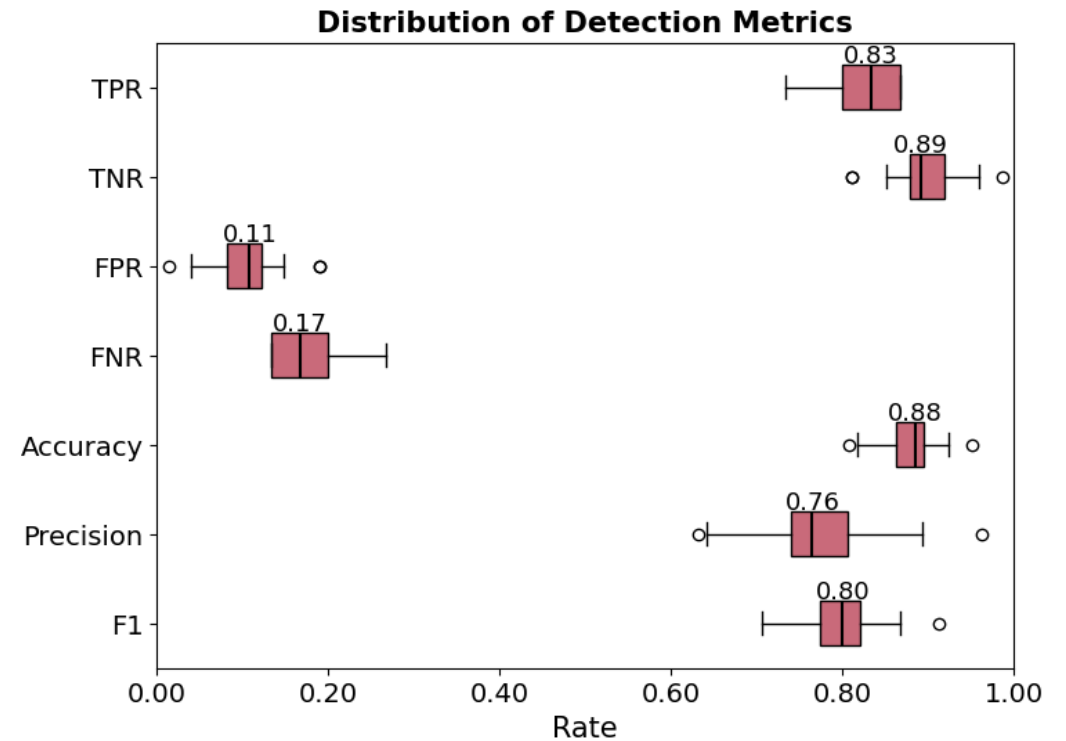
Methodology



Summary Stats (Baseline VS v1)

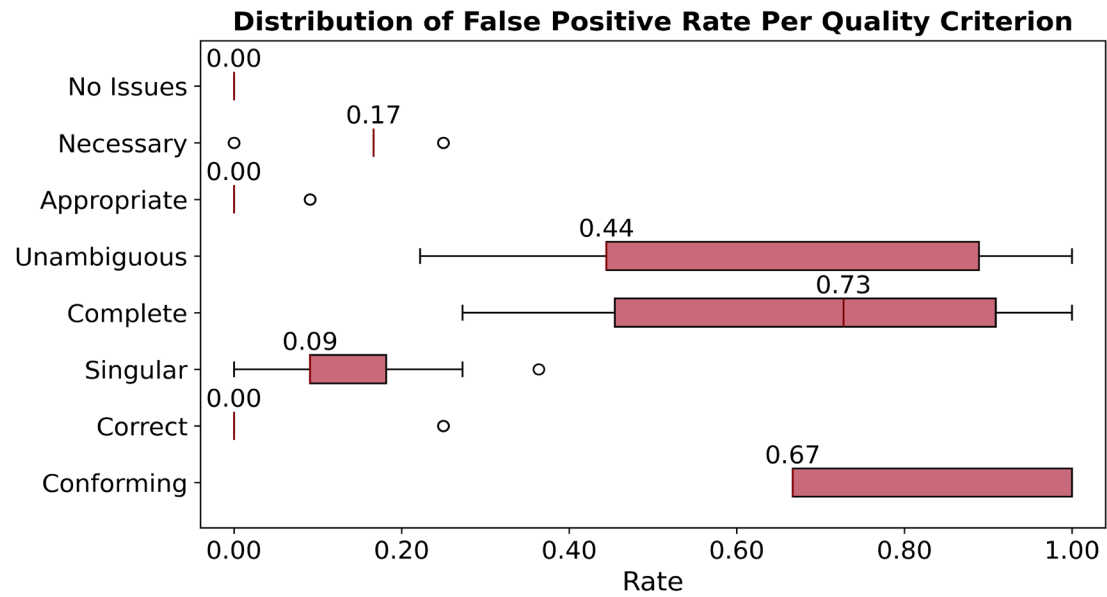


Opus 5 Baseline

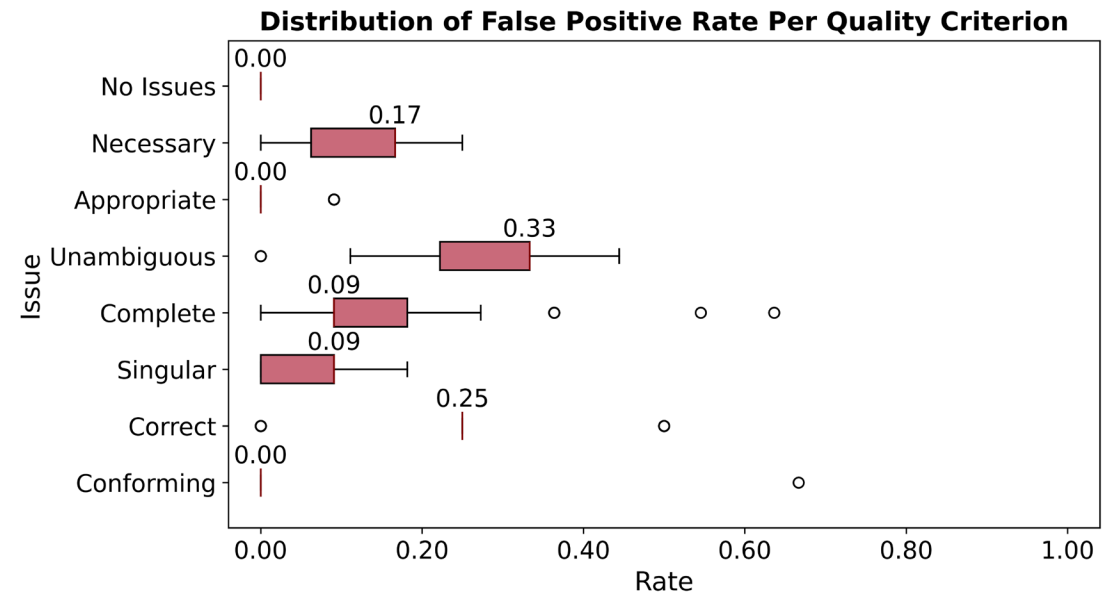


Prototype V1

False Positive Rates (Baseline VS v1)

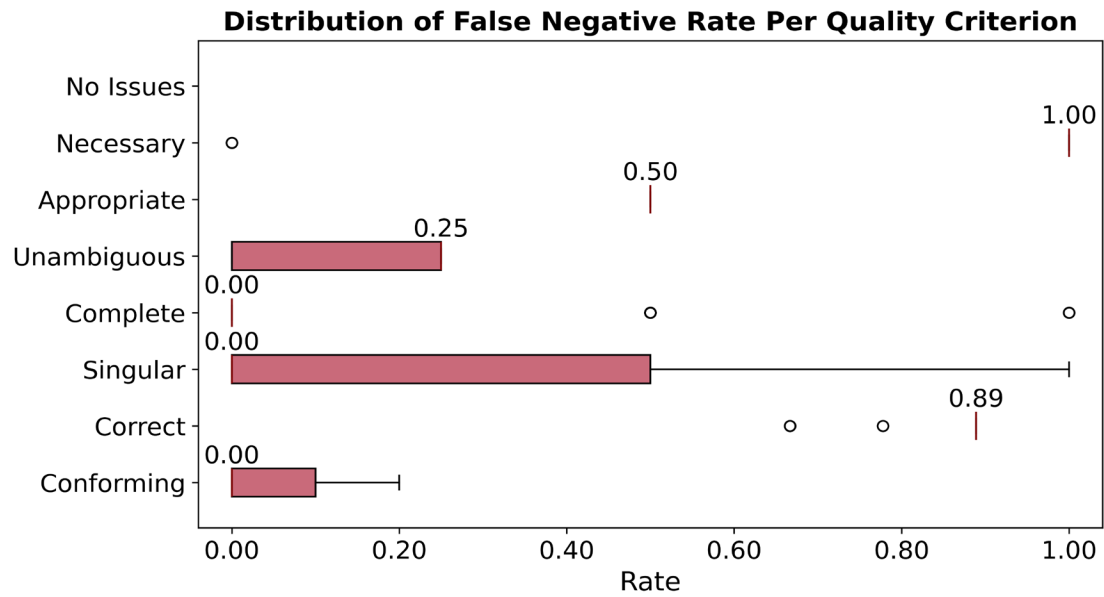


Opus 5 Baseline

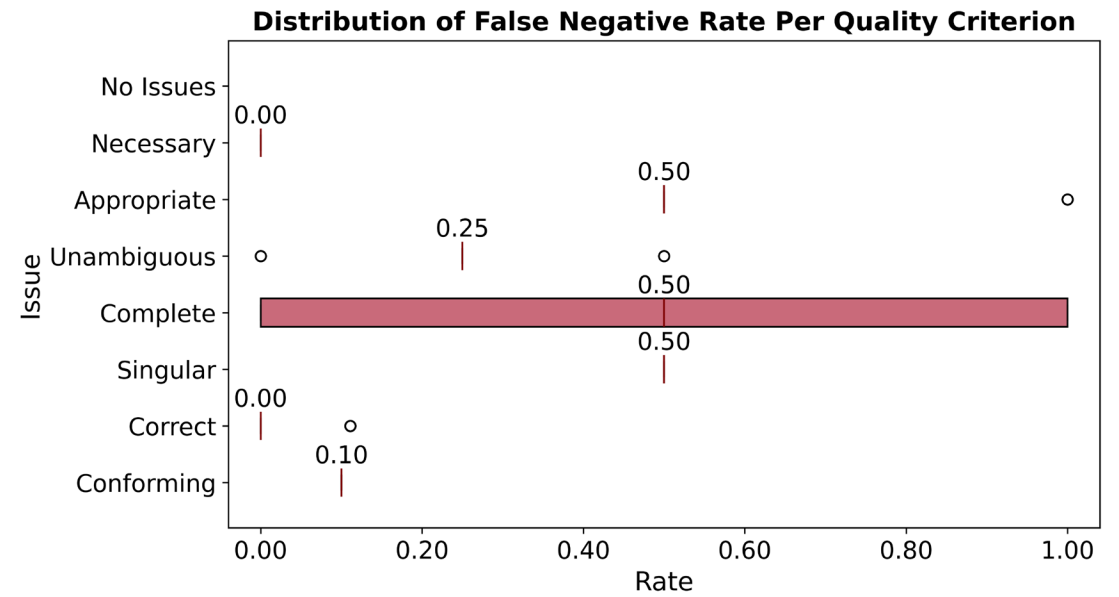


Prototype V1 (RQ)

False Negative Rates (Baseline VS v1)

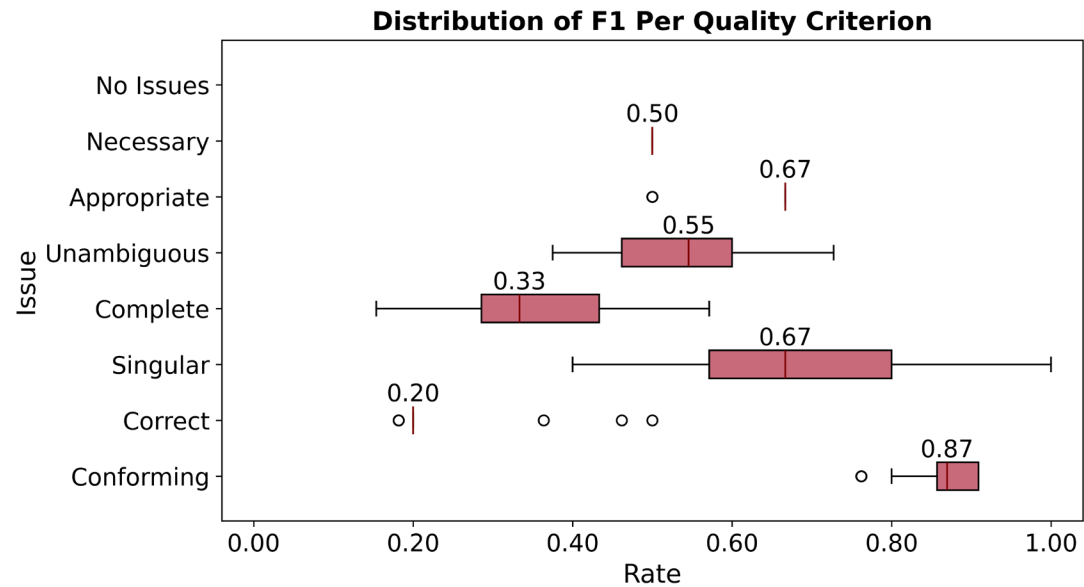


Opus 5 Baseline

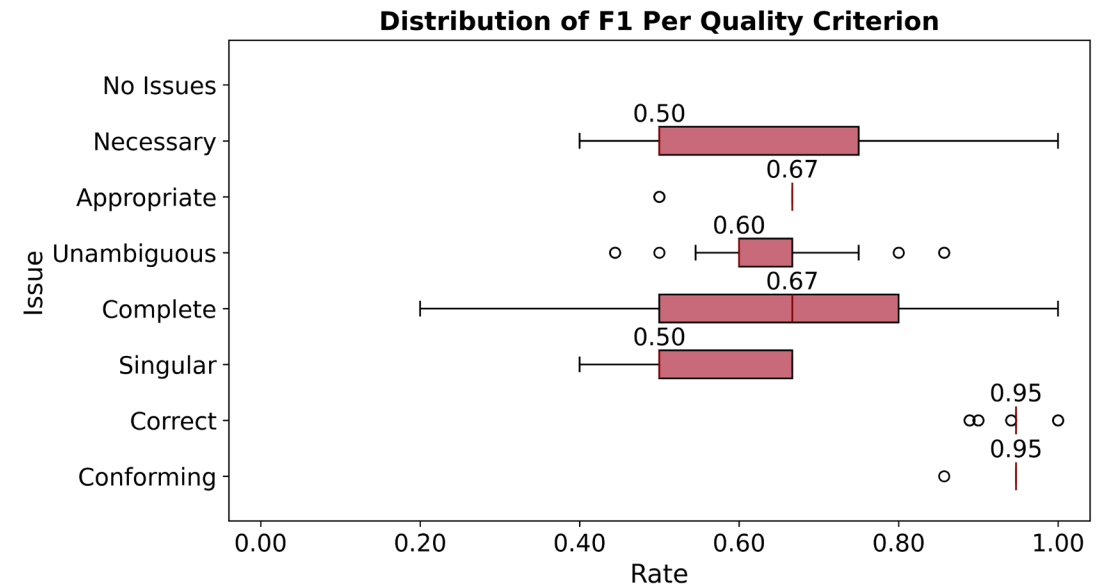


Prototype V1 (RQ1)

F1 (Baseline VS v1)

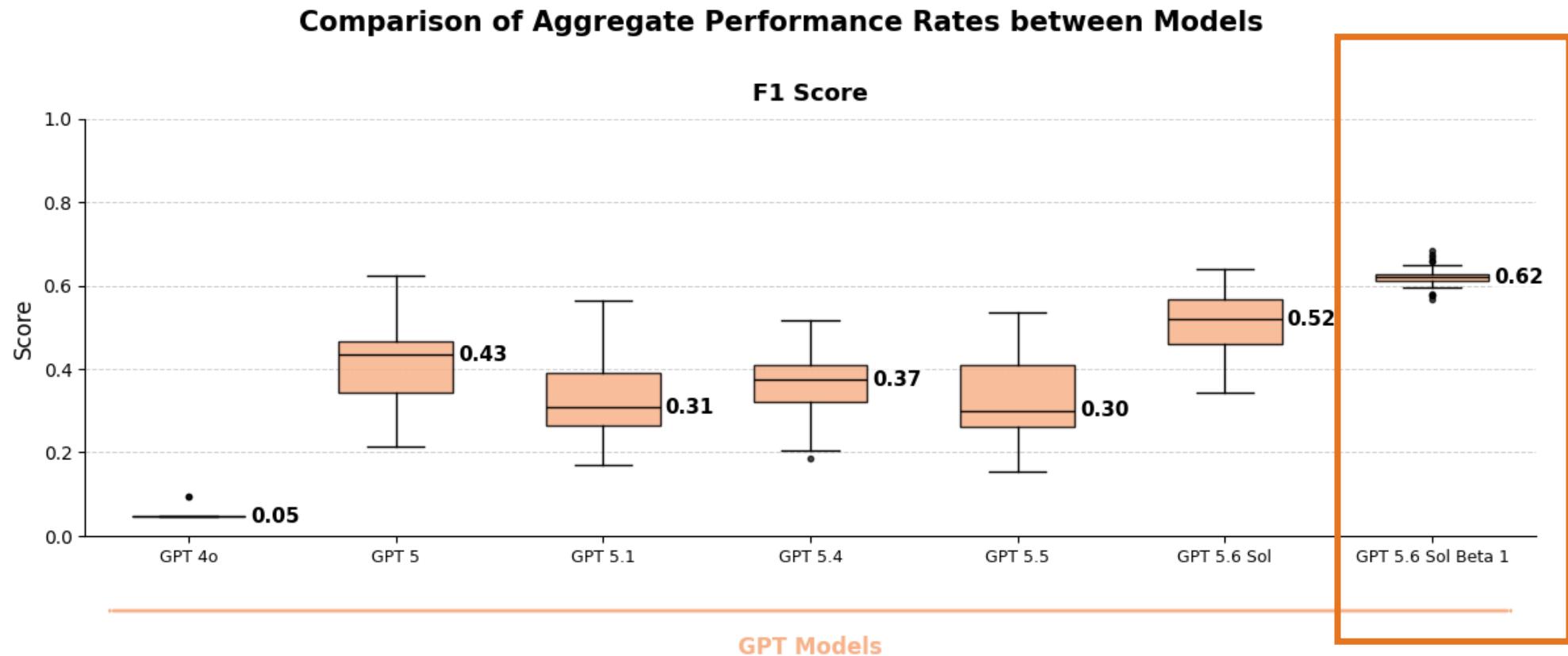


Opus 5 Baseline



Prototype V1

Bonus: Generational Trends GPT models



Metrics

- **True positive rate (TPR/Recall):** Among all real issues present, what fraction did the LLM correctly identify? (the more the better)

$$TPR = \frac{TP}{\text{Ground Truth Total Positive}}$$

- **True negative rate (TNR):** Among all non-issues, what fraction were correctly left unflagged? (the more the better)

$$TNR = \frac{TN}{\text{Ground Truth Total Negative}}$$

- **False positive rate (FPR):** Among all non-issues, what fraction were incorrectly flagged? (the less the better)

$$FPR = \frac{FP}{\text{Ground Truth Total Negative}}$$

- **False negative rate (FNR):** Among all real issues present, what fraction were missed? (the less the better)

$$FNR = \frac{FN}{\text{Ground Truth Total Positive}}$$

- **Accuracy:** Out of all predictions made, what fraction were correct? (the more the better)

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

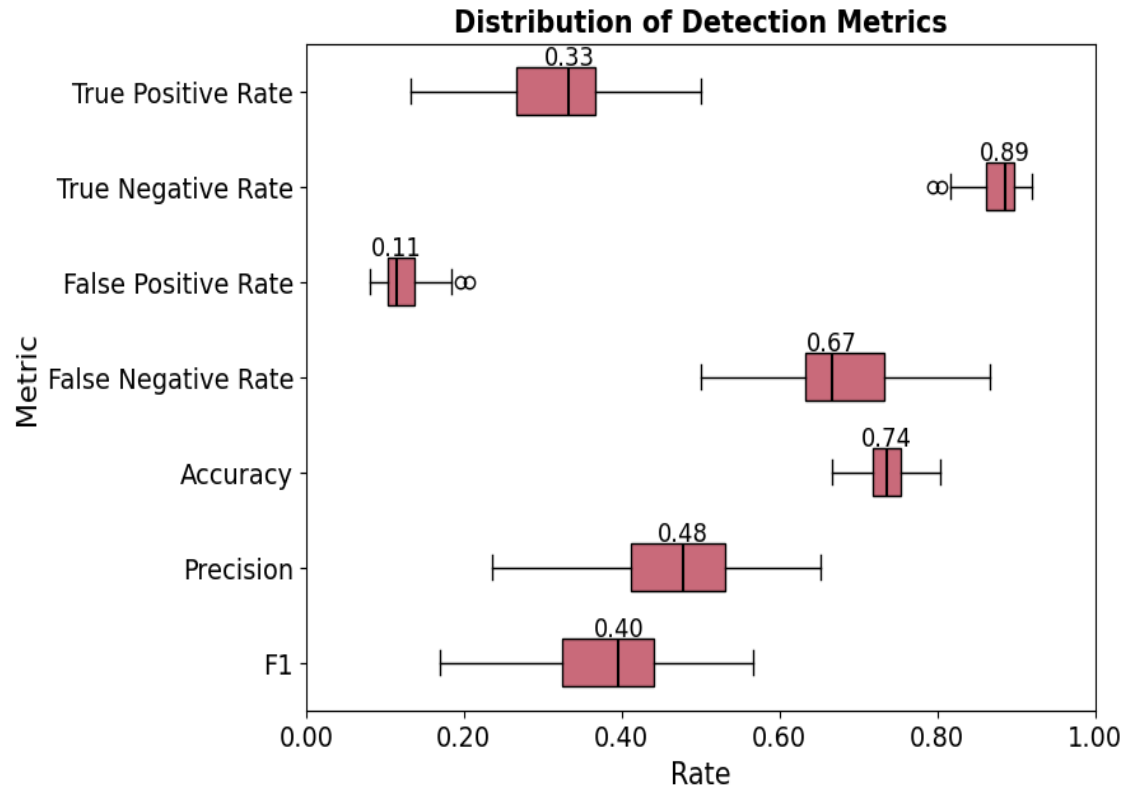
- **Precision:** Among all issues flagged by the LLM, what fraction were actual issues? (the more the better)

$$Precision = \frac{TP}{TP + FP}$$

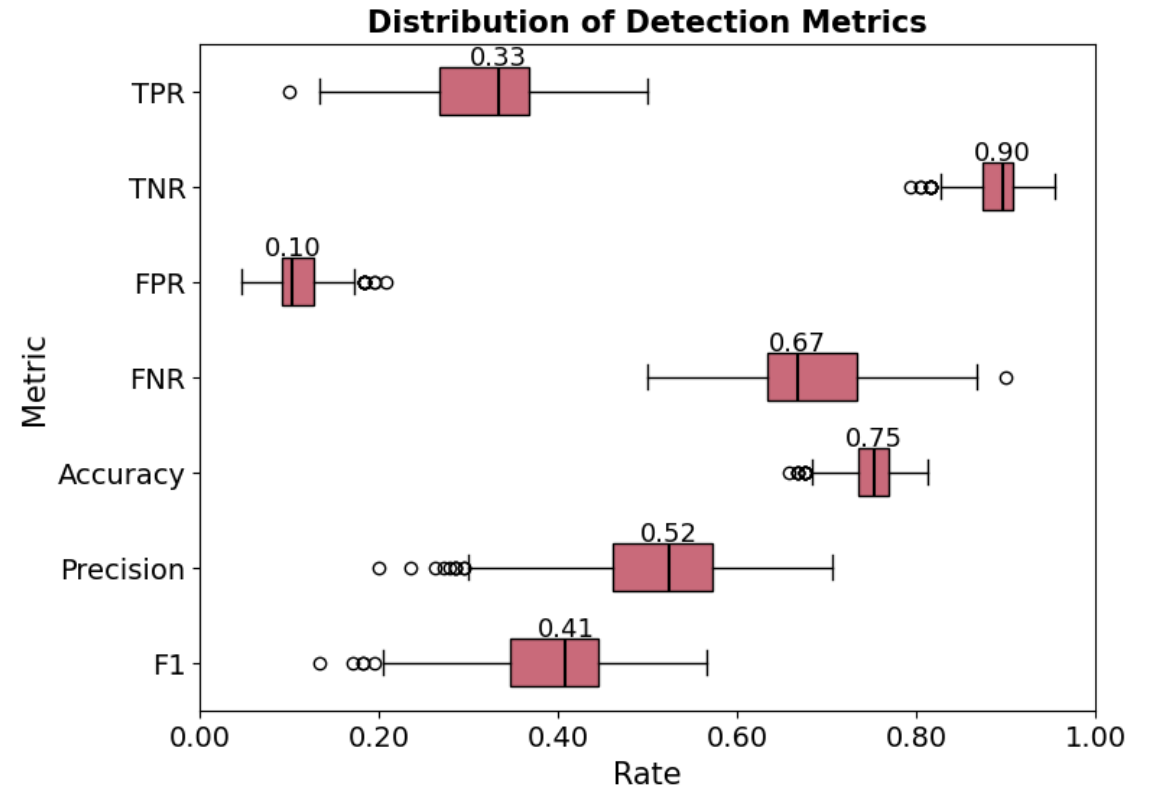
- **F1 Score:** Harmonic mean of Precision and Recall, balancing both in a single metric. (the more the better)

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

100 vs 500 Runs



Claude Opus 4.7 Baseline (100 runs)



Claude Opus 4.7 Baseline (500 runs)

Artificial Intelligence to the Rescue?

- **Requirements** have been an early AI4SE research target mostly since they are expressed in natural language

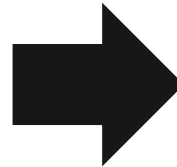
Rule-based NLPs [Fabbrini et al. 2001]

~1980s – 1990s

Ambiguity and vagueness detection

Useful for subtasks

Limited due to extensive fine-tuning



Supervised/Unsupervised ML

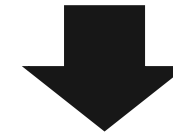
[Kurtanović and Maalej 2017; Zhao et al. 2021]

~2000s

Classification and Prediction based on human-labeled data

SVMs for boundaries, CNNs for patterns

Limited due to dependency on large labelled sets, cross domain extensibility



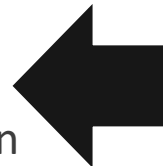
Transformer-based Models

[Vaswani et al. '17, Berquand et al. '21, Ray et al. '23]

~2010s

Transformers and transfer learning, pre-trained self-attention architectures, BERTs

Limitation token limits and context awareness, mostly classification tasks with high accuracy



LLMs and Agentic AI [Lewis et al. '20; Gao et al. '24, Arora et al. '24, Norheim et.al. '24, Massoudi and Fuge '25)

~2020s

GPTs, RAG, Agentic AI..

Context awareness and task specialization

Limited due to stochasticity and lack of interdisciplinary knowledge, fundamentally some form of extrapolation