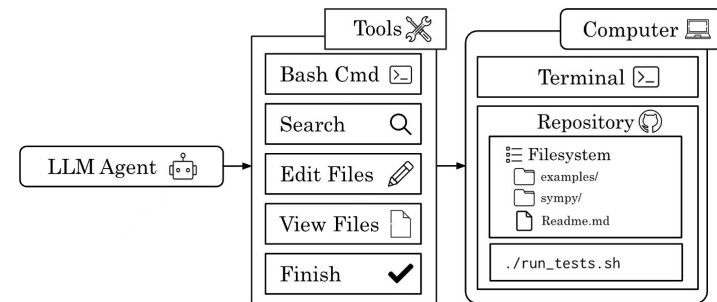


Human-In-the-Loop Software Development Agents

Zhiheng Lyu 21143093

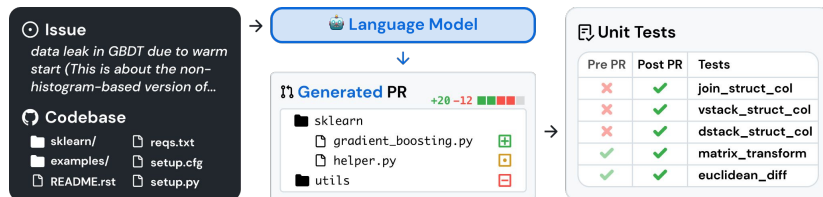
LLM Code Agent for SWE

- Agentic LLM are Strong on SWE Tasks, but how good are they?

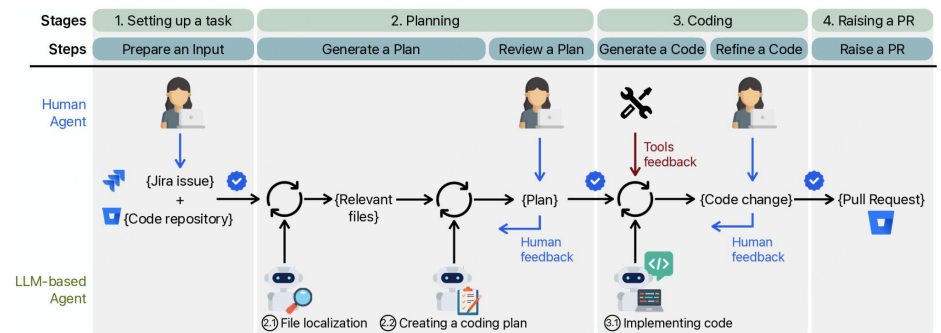


Offline Benchmarks - SWE-Bench

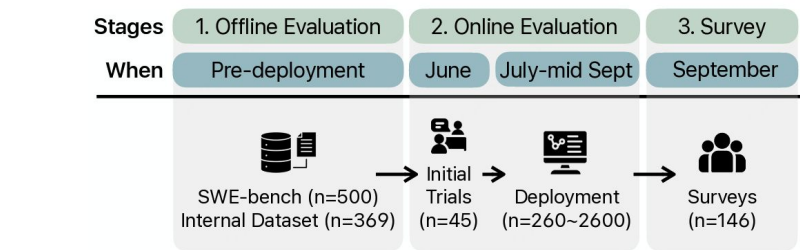
- Paradigm of SWE-Bench
 - Python-only, 12 OSS projects
 - Long, detailed issue descriptions (~295 tokens)
 - Complete unit tests
- Problems?
 - No multi-language, no enterprise diversity
 - Tasks too standardized vs. industry complexity
 - End-to-End, coarse granularity



HULA: Human-in-the-Loop LLM-based Agents Framework



How to Evaluate HULA?



Evaluation Stage 1 - Offline Evaluation

TABLE I
A STATISTICS SUMMARY OF BENCHMARK DATASETS.

Dataset	Data	Min	Median	Max
SWE-bench Verified (n=500)	Issue Information (token)	24	295	6,939
	Changed files (count)	2	2	22
	Human-written code (token)	89	220	5,057
Internal (n=369)	Issue Information (token)	11	75	1,114
	Changed files (count)	1	3	44
	Human-written code (token)	82	1,275	47,520

TABLE III
(RQ1) THE OFFLINE EVALUATION RESULTS OF HULA.

Metrics	SWE-bench Verified	Internal
%Issues for Success Generation	97%	100%
Recall of File Localization	86%	30%
%Issues for Perfect File Localization	84%	15%
%Issues for Perfect Passing Test Cases	31%	-
%Issues of High Code Similarity	45%	30%

Evaluation Stage 2 - How HULA Works?

Evaluation Stage 2 - Usage Flow

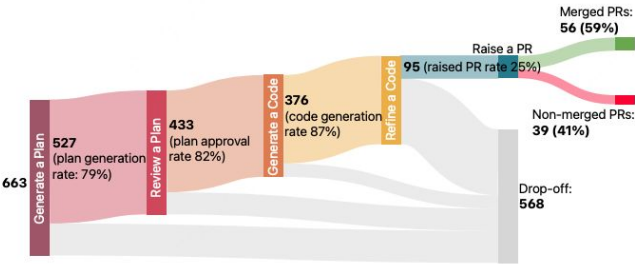
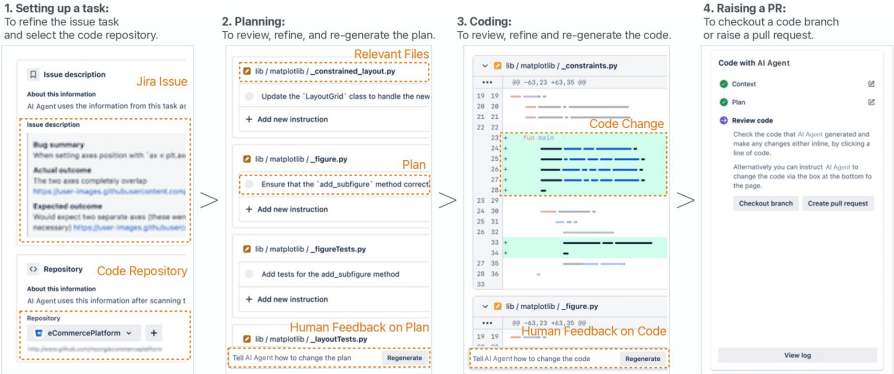


Fig. 4. (RQ2) The Online Evaluation Results of HULA.

Evaluation Stage 3 - Survey (n=109)

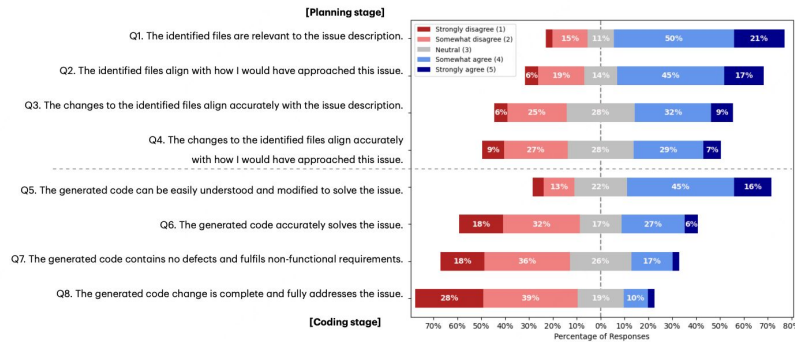


Fig. 6. The Survey Responses on the Satisfaction of the Generated Plans and Code by HULA (n=109).

Conclusion

HULA: First human-in-the-loop LLM agent framework deployed in Atlassian JIRA.

Key findings:

- Works well for planning and simple tasks.
- Still challenges in code quality and complex issues.

Takeaways:

- Human-AI collaboration > full automation (at least for now).
- Benchmark–real world gap → future benchmarks must evolve.

Future: Improve context, richer evaluation metrics, continuous learning from feedback.

Review: Positive Point

- **Real-world deployment inside Atlassian JIRA** → beyond lab studies
- **Multi-stage evaluation:** offline, online, survey → rigorous triangulation
- **Human-in-the-loop design:** pragmatic, reduces context switching & workload
- **Strong adoption metrics:** 80%+ plan approval, 50+ merged PRs

Review: Negative Point

- **Lack of comparison with IDE-native tools** (e.g., Cursor, Copilot) → unclear if HULA is the best collaboration paradigm
- **Limited technical novelty** → mainly orchestration of GPT-4 into workflow
- **Insufficient task-level analysis** → no breakdown of which issue types (fix, refactor, feature) succeed or fail more often, nor difficulty profiling
- **Weakness in agentic scaffold** → pipeline design still fragile, may not generalize well

Rating & Future Work

Rating: 3.5 / 5 – solid deployment study with valuable insights, but limited novelty and code quality concerns remain

Future Work:

- Improve context retrieval (docs, history, embeddings)
- Richer evaluation metrics (readability, maintainability, defect density)
- Continuous learning from developer feedback
- **Agentic Arena:** benchmark where developers compare different agents head-to-head

Discussion Points & QA

- **Future Paradigm:** Human-in-the-Loop (Cursor) v.s. Fully Autonomous (Claude Code like, End2End)?
- **Efficiency vs quality:** would companies accept “AI writes 70%, human edits 30%”?
- **Benchmark vs reality:** how should we design next-gen datasets?

The End.

Thanks for your listening.