Kozuchi Agent

A Language-Agnostic Open-Weight Agent for Software Repair

Mehdi Bahrami, Kosaku Kimura, Satoshi Munakata, Satoshi Nakashima, Yu Ishikawa, Kosuke Maeda, Nao Soma, Kenichi Kobayashi, Keisuke Miyazaki, Keizo Kato, Shigeki Fukuta, Tatsuo Kumano, Nobutaka Imamura, Kevin Musgrave, Shahbaz Abdul Khader, Kwun Ho Ngan, Joe Townsend, Fayas Asharindavida, Matthieu Parizy, Akira Sakai, Yuma Ichikawa, Yang Zhao, Michiaki Takizawa, Taku Fukui, Hiroki Ohtsuji, Wei-Peng Chen, Hiromichi Kobashi

mbahrami@fujitsu.com

27B open-weight model • no fine-tuning • no learned verifier • hidden tests never used for selection

Why the harness matters?

- Long repair horizons lose evidence and repeat work.
- Tool-call syntax drifts across model families.
- Inference, testing, and grading span heterogeneous clusters.
- Full benchmark reruns are too expensive for rapid iteration.

THESIS The next performance gain came from the harness—not the weights.

Results at a glance

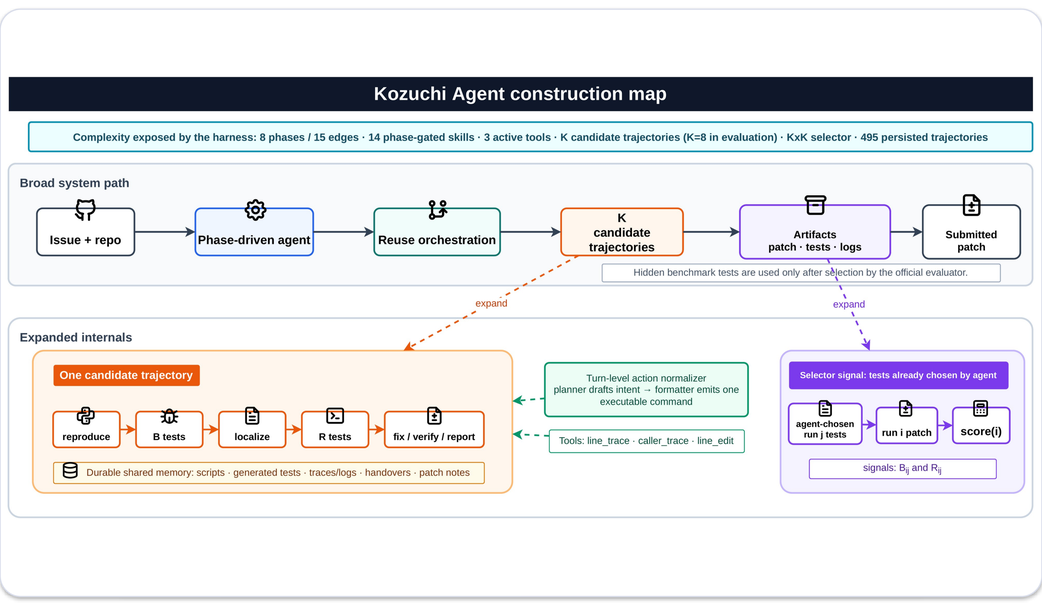
74.80%
SWE-bench Verified
374 / 500 resolved

#1
open-weight agent
rank 12 / 135 overall

27B
parameters
smallest model on Pareto front

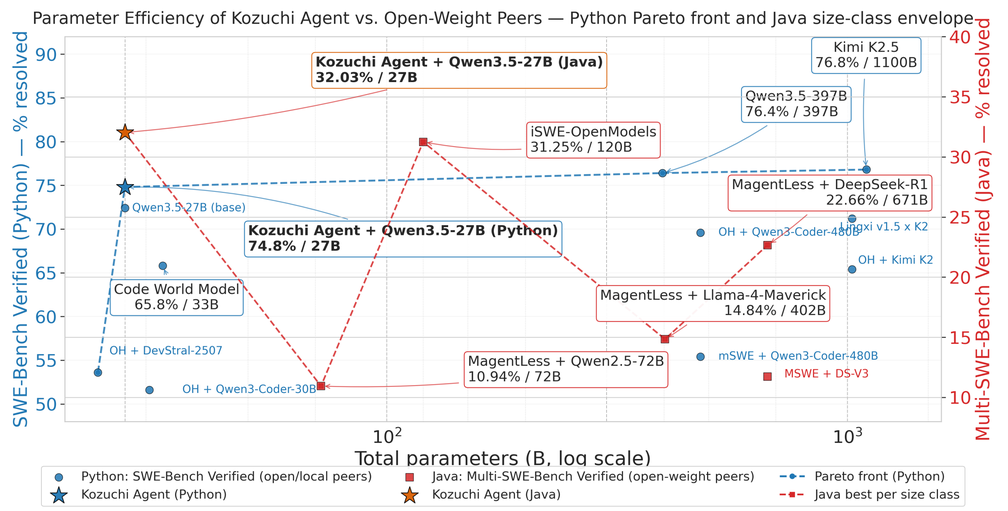
32.03%
Multi-SWE-bench Java
41 / 128 • #1 open-weight

One auditable harness, end-to-end



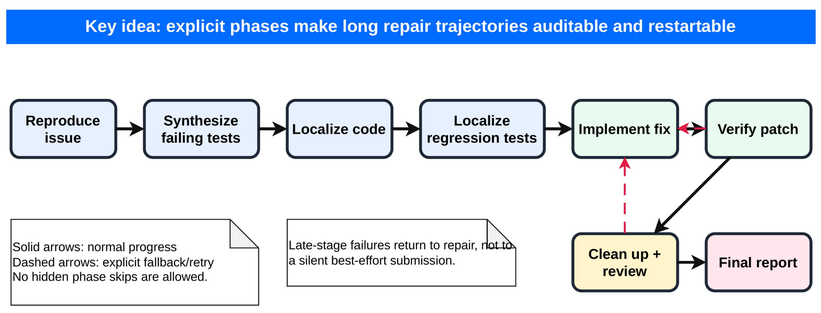
Issue + repository → K auditable candidate trajectories → cross-agent testing → submitted patch

Parameter-efficient on Python and Java



The same 27B configuration sits on the Python open/local Pareto frontier and leads its Java size class.

Method: explicit phases + cross-agent tests



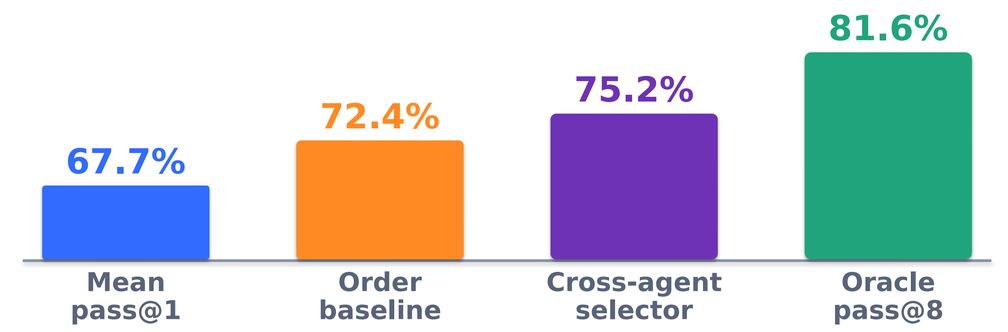
K = 8 independent runs

Each run contributes one patch plus bug-revealing and regression tests. Tests from run j are executed against patch i, producing a K×K evidence matrix.

$$\text{score}(i) = \frac{1}{K}\sum_{j}[0.3\,B_{ij} + 0.7\,R_{ij}]$$

NO HIDDEN-TEST ACCESS

Where the gain comes from



+7.5 pp vs. mean pass@1 • +2.8 pp vs. order baseline • 92.2% of oracle union captured

Auditable by design

- **8 phases / 15 edges**
- **Filesystem-backed state**
- **One action per turn**
- **3 deterministic SE tools**

5 operator touch-points → 1 CI push
6/9 stages reusable
495 persisted trajectories

Reliability and transfer

494 / 495
produced patches apply cleanly

91.3%
of unresolved cases are clean-applying WRONG_FIX patches (model limitation)

Same scaffold, two languages

Per-phase message share remains within ±5 pp. Remaining Java failures are dominated by language/harness effects—not a change in the agent contract.

TAKEAWAYS

1. **Workflow is a capability multiplier.**
2. **Diversity needs executable evidence.**
3. **The remaining gap is semantic correctness + selection.**

SCAN FOR PAPER • ARTIFACT CODE • REPRODUCTION

Paper DOI: 10.1145/3832783.3834531

https://marscod.github.io/kozuchi-agent-artifact/