Loop Engineering

Loop Engineering is a proposed approach to building software whereby an AI agent is used in a validation/feedback loop to essentially validate whether the task is complete yet or not.

The term is believed to have been coined by Addy Osmani, Peter Steinberger and Boris Cherny in June 2026. It is believed that this way of working emerged once agentic AI coding tools crossed a threshold:

> Coding agents had become reliable enough to finish a non-trivial task unattended; scheduling primitives had appeared in the major harnesses; and the cost of a single agent run had dropped far enough that running one repeatedly, on a timer, stopped looking wasteful.

Separate Generator and Evaluator Agents

The IEEE working note suggests that it is very important to separate the generator from the evaluator in the loop. The agents that generate code are typically very keen to praise themselves even when a human can plainly see that quality is mediocre. The context for the coding agent is stuffed full of reasons why the code was written in a particular way. So the agent looks at its own reasoning and sees a chain of self persuasion.

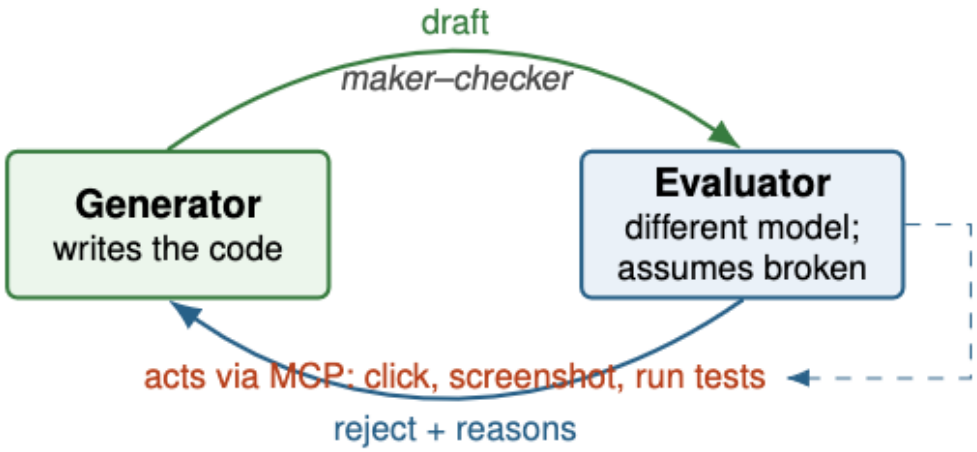


Fig. 3. Generator and evaluator as separate agents. The evaluator carries none of the generator's self-persuasion, defaults to doubt, and judges behavior by acting rather than just reading.

The key is to build an individual independent skeptic rather than trying to train the generator to be self critical. This is a similar architecture to the GAN model.

The evaluator should act, not just read. This implies that the model should do something like run tests, ideally end-to-end tests, that exercise the complete product. But they will also need to give enough feedback and granularity that the agent would know what to do with them afterwards.

References

- Addy Osmani Loop Engineering
- Loop-Engineering-IEEE.pdf

Revision #6
Created 2026-09-29 10:15:07 UTC by James
Updated 2026-09-29 12:30:34 UTC by James