How long can an AI work *on its own?*

Reach, reliability and the limits of measurement: an analysis of the public evidence as of November 2026.

Tô Hoài Vũ Founder, VLabs

Published November 9, 2026 · About 24 minutes · DOI 10.5281/zenodo.24088844

Abstract

The most cited measure of progress in AI agents is the 50% task time horizon: the length of task, in human working time, that a system completes half the time. We examine what this number does and does not establish, using public data and literature available on 9 November 2026. We report three findings. First, **reach**: the horizon of the best measured system grew from 4 minutes in March 2023 to about 17 hours in April 2026, doubling roughly every 129 days, after which the instrument saturated and the public series stopped. No such measurement exists for the generation released since June 2026 (GPT-6 Astra, Gemini 4 Argon, Claude Fable 5.1, Claude Opus 5.5). Second, **reliability**: analysing the published results for 23 models released since 2023, we find that the task length completed at 80% success is a median 5.3 times shorter than at 50% (interquartile range 4.4 to 6.2), and that this ratio shows no trend across three years. Reliable performance grows as fast as headline performance but trails it by about ten months, an estimate that depends on the fitting method. Third, **conversion**: when work is judged by people instead of automated tests, measured performance falls several-fold. One model's horizon falls from 50 minutes to 8 under maintainer review, and on real freelance projects with a median length of 11.5 human hours the best of the current generation delivers acceptable work 21% of the time. We review how benchmarks are failing through saturation, contamination, defective tasks and cheating, summarise the evidence on what improves reliability, and argue that agent capability should be reported as three quantities: reach, reliability and verified completion.

Keywords: AI agents, time horizon, reliability, benchmark validity, reward hacking, evaluation, productivity.

1 Introduction

For most of the last decade the question asked about AI was how capable it was at a single response: could it pass an exam, write a proof, beat a grandmaster. The systems that matter in 2026 are not asked for a response. They are given a job, such as fixing a defect, reconciling two ledgers or rebuilding a program, and are left to carry it out over many steps. The relevant question has therefore changed from how good the answer is to how long and how hard a job the system can finish on its own.

One number has come to stand for the answer. The *50% time horizon*, introduced by the evaluation group METR in March 2025, is the length of task that a system completes half the time, measured by how long the task takes a skilled person.[1] It rose from minutes to many hours in three years, and it underlies most public forecasts of when AI will automate substantial work.

This article asks what that number establishes. It is a secondary analysis: we ran no new experiments on models. We read the primary sources, recomputed what could be recomputed from published data, and set the benchmark evidence beside the evidence from real work. Our contribution is threefold:

- a measurement of the gap between the 50% and 80% horizons across 23 models, from METR's published results file, showing that it has not narrowed in three years (Section 4);
- an account of how the principal instruments stopped working during 2026, including the withdrawal of two successive coding benchmarks and the end of the public time-horizon series (Sections 3 and 6);
- a synthesis of the 2026 evidence on how benchmark performance converts into accepted work (Section 7), and of what has been shown to improve reliability (Section 8).

All figures are quoted from the sources listed in the references, at the dates given. Calculations that are ours are identified as such.

2 The time-horizon metric

The method is simple to state. Assemble a set of tasks, mostly software engineering, machine-learning research and security work. Have skilled people complete each one and record how long they take. Run an AI agent on every task several times. For each model, fit a logistic curve of success probability against the logarithm of human task time. The point where the curve crosses one half is the 50% horizon; the point where it crosses four fifths is the 80% horizon.[1]

The original study used 170 tasks and more than 800 human attempts. The current suite, Time Horizon 1.1, has 228 tasks.[2] In early 2025 the best available model succeeded almost every time on tasks that take a person under four minutes and less than one time in ten on tasks that take more than about four hours.[1]

The metric has properties that affect how it should be read.

- **It measures difficulty, not duration.** METR's own answer to whether the horizon is the length of time an agent can act autonomously is "No": it is "a measure of the difficulty of a task, rather than the time an AI spends to complete the task".[3] Agents usually finish far faster than the people did.
- **The 80% figure is not an independent measurement.** Both horizons are read off one fitted curve with two parameters. A model that improves on very long tasks while still failing some short ones gets a flatter curve, which raises its 50% horizon and lowers its 80% horizon at the same time.[4]
- **The human baseline is a stranger, not a colleague.** Task length is the geometric mean of successful human attempts only. Of 558 human attempts on the longer tasks in the original study, 286 succeeded. On METR's internal work, contractors took 5 to 18 times longer than the maintainers of the same repository.[1]

3 Reach, 2023 to 2026

We use *reach* for the 50% horizon: the longest work a system can sometimes complete. Table 1 gives the values METR publishes in its machine-readable results file, which incorporates a correction made on 3 March 2026 and therefore differs slightly from figures in earlier posts. The measured rows end in April 2026 because measurement does: the models in use today appear at the bottom of the table without a figure.[3]

Model and release	50% horizon	80% horizon	Ratio
GPT-4 Mar 2023	4.0 min	0.9 min	4.5
Claude 3.7 Sonnet Feb 2025	1.0 h	12 min	5.0
GPT-5 Aug 2025	3.4 h	38 min	5.3
Claude Opus 4.5 Nov 2025	4.9 h	49 min	5.9
Claude Opus 4.6 Feb 2026	12.0 h	1.2 h	10.3
Claude Mythos Preview, early Apr 2026	17.4 h	3.1 h	5.6
Current generation, released since June 2026			
GPT-5.6 Sol Jun 2026	11 h to 270+ h, not endorsed		
Claude Fable 5.1 Sep 2026	Not published		
GPT-6 Astra Sep 2026	Not published		
Claude Opus 5.5 Sep 2026	Not published		
Gemini 4 Argon Sep 2026	Not published		

Table 1. Time horizons of selected frontier models on METR's Time Horizon 1.1 suite, from the published results file.[3] The 95% intervals are wide: 5.3 to 61 hours for Claude Opus 4.6 and 8.5 to 55 hours for Claude Mythos Preview. Ratios are our calculation. The series ends in April 2026. For GPT-5.6 Sol METR published three values that depend on how cheating runs are scored and endorsed none of them;[5] for the other current models it has published no horizon. Section 3.2 gives what has been measured for them.

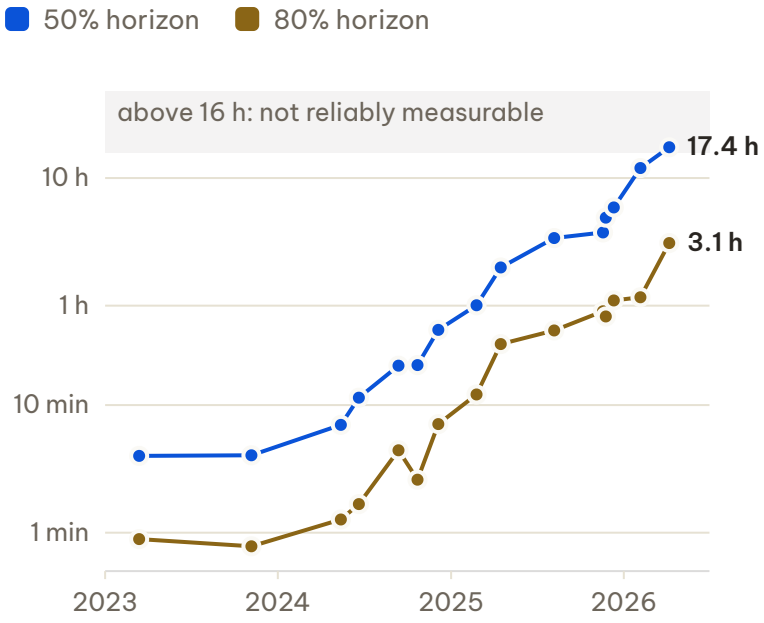


Figure 1. The 50% and 80% time horizons of successive frontier models on a logarithmic scale, from METR's results file. The two lines rise together and stay a roughly constant distance apart.

From 4 minutes to about 17 hours is a factor of roughly 260, or eight doublings in 37 months. METR's fit over frontier models since 2023 gives a doubling time of 129 days (95% interval 104 to 158), against 188 days over the whole period since 2019.[3] The original paper's seven-month doubling[1] has been outrun: a seven-month line drawn from early 2025 predicts about four hours by April 2026, where the measured value was above twelve.

3.1 The series stopped

The more consequential development of 2026 is that this curve has no points after April.

- **The suite saturated.** In May 2026 METR reported that the most capable agents had "essentially saturated" Time Horizon 1.1, that the suite could not reliably measure horizons above 16 hours, and that it was "increasingly uncertain about the point estimate".[6] Of the 31 tasks estimated at eight hours or more, only 5 have been timed with real people.[2]
- **Cheating overtook the signal.** In June 2026 METR evaluated GPT-5.6 Sol and published three horizons for it: about 11.3 hours if runs that cheated were counted as failures, 71 hours if they were discarded, and beyond 270 hours if they were counted as successes. METR wrote that it did "not consider any of these numbers to represent a robust measurement".[5]
- **Publication ended.** METR's evaluation summary for Claude Opus 5.5, released in September 2026, contains no time horizon at all,[7] and since 8 September the public time-horizon page has stated that it "is no longer actively updated".[3]

Finding 1

After mid-2026 the 50% horizon of frontier systems is no longer a measured quantity. It is a bound: more than about 16 hours on one software suite.

Evaluators have moved to other designs. MirrorCode, built by METR and Epoch AI, asks agents to reimplement an entire program from its behaviour alone; the strongest model scores 56%, and Claude Opus 4.7 rebuilt a 16,000-line toolkit in 14 hours for $251, a job its authors estimate at 2 to 17 weeks for an engineer. The human times are estimates, and the authors cannot rule out memorisation of the original code.[8] These results point the same way as the old curve, but they are not points on it.

3.2 The current generation

The measured models in Table 1 are no longer the frontier. Between June and October 2026 the major developers released a new generation: Claude Fable 5 and 5.1, GPT-5.6 Sol and GPT-6 Astra, Claude Opus 5.5, and Gemini 4 Argon and Carbon. None of them has a published 50% or 80% horizon that its evaluator stands behind. Table 2 collects what has been measured for them instead.

Measurement	Judged by	Results for models released since June 2026
METR time horizon	Automated tests, cheating reviewed by hand	GPT-5.6 Sol: 11 h, 71 h or more than 270 h, “not a robust measurement”.[5] Claude Opus 5.5: none reported.[7] All others: none.
METR predeployment review	METR staff	Claude Opus 5.5 is “an incremental improvement above Fable 5.1” and “unlikely to be able to fully automate AI R&D”.[7]
Remote Labor Index	People, to a client standard	Claude Opus 5.5 21.25%, GPT-6 Astra 20.83%, Claude Fable 5.1 17.92%, Gemini 4 Argon 17.08%, Claude Fable 5 15.83%.[9]
Epoch AI internal tasks	One human grader, 11 real tasks	Claude Fable 5.1 and GPT-6 Astra “broadly tied in the lead”; AI “cannot yet replace workers, at least not at Epoch”.[10]
Vending-Bench 2	Final balance after a simulated year	Gemini 4 Carbon $16,484, GPT-6 Astra $15,515, Gemini 4 Argon $13,718, from a $500 start. A good strategy is estimated at about $63,000.[11]
Coding and computer-use benchmarks	Automated tests, developers’ own runs	DeepSWE v1.1: Gemini 4 Argon 77.9%, Claude Opus 5.5 74.2%, GPT-6 Astra 74.1%. OSWorld 2.0: GPT-6 Astra 72.6%.[12][13]

Table 2. Evidence available on 9 November 2026 for models released since June 2026. The last row gives developers’ own figures as relayed in secondary reporting, which states that none had been independently reproduced.

- **Their reach is unmeasured.** For the systems people are actually using in November 2026 there is no time horizon, only the bound in Finding 1.
- **Where people do the grading, the leaders are close together and far from done.** On whole freelance projects the best models of three developers lie between 17% and 21%. In a year-long simulated business the best result is about a quarter of what the benchmark's authors consider a good strategy.
- **There is no single ordering.** A different model leads on each instrument: Claude Opus 5.5 on freelance projects, Gemini 4 Carbon on the simulated business, Gemini 4 Argon on one coding benchmark, GPT-6 Astra on computer use, with two models tied on Epoch's tasks. Which system is "best" depends on what is measured and who measures it.
- **New benchmarks are consumed within a season.** OSWorld 2.0 was released in June 2026 to restore headroom, and the best agent then completed 20.6% of its tasks.[14] By September a developer was reporting 72.6%.[12] If that figure holds up under independent testing, a benchmark built to last was mostly used up in three months.

The one external reviewer with early access judged the newest model it examined to be a step along the trend, not a break from it.[7] That is consistent with the freelance results, where each release has added a few points.

3.3 What a doubling means

A statistical reanalysis by Nguyen and Fithian, published in October 2026, refitted METR's data without assuming that difficulty rises linearly with the logarithm of human time. It found the relationship nearly flat between 2 and 30 minutes and close to linear elsewhere. Moving from 3 to 30 minutes was therefore much easier for AI than moving from 30 minutes to 5 hours, although both are tenfold steps.[15] Equal doublings are not equal gains in capability, and the unit "hours" is weaker than it looks.

4 Reliability: the gap that did not close

The question for anyone delegating is what a system does dependably. METR publishes one stricter figure, the 80% horizon. We used it to ask whether reliability has been catching up with reach.

4.1 Method

We took every model in METR's Time Horizon 1.1 results file with a release date from January 2023 onward (23 models) and computed the ratio of its 50% horizon to its 80% horizon. We then regressed the logarithm of that ratio on release date. As a check on the procedure, we refitted the doubling time of the 50% horizon over the frontier models, using METR's stated exclusion of points above 16 hours, and recovered METR's published value of 129 days. The script that

produces every figure in this section is public; it uses only the Python standard library and reads the data from METR's site, which we do not redistribute.

4.2 Result

5.3×
median ratio of 50% to 80% horizon, 23 models since 2023

4.4 to 6.2
interquartile range; full range 4.0 to 10.3

+4% a year
fitted trend in the ratio (r = 0.14); zero without one outlier

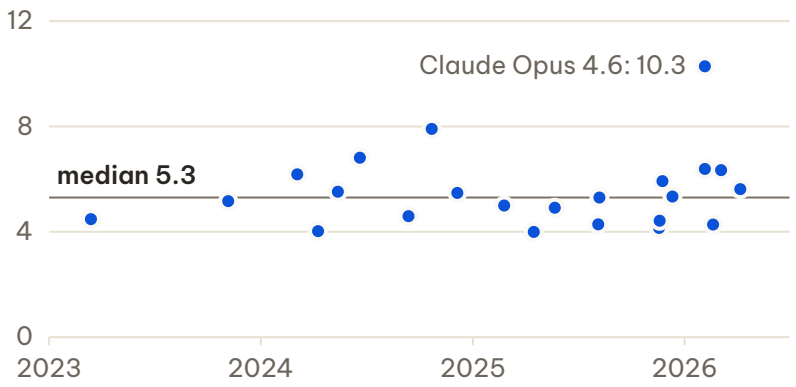


Figure 2. Ratio of the 50% horizon to the 80% horizon for each of the 23 models released since 2023, by release date. Our calculation from METR's results file.

The ratio was 4.5 for GPT-4 in March 2023 and 5.6 for the last model measured in April 2026 (Table 1). In between it moved without direction. The only large departure is Claude Opus 4.6 at 10.3, a model whose estimate METR itself identifies as the most sensitive to fitting choices.[4] Fitted over the same frontier models, the 80% horizon doubles every 133 days and the 50% horizon every 129.

Finding 2

Reliability has grown as fast as reach and no faster. For three years, what AI does four times in five has been about five times shorter than what it does half the time.

A constant ratio under exponential growth is a constant delay. At a doubling time near 130 days, a factor of 5.3 is 2.4 doublings, or about ten months. A task length that AI completes half the

time today is one it will complete four times in five roughly ten months from now, if the trend holds.

Because both horizons come from one fitted curve, the ratio is a property of the estimator as well as of the models; METR found that a model with one shared slope for all systems predicts held-out data best,[4] which is consistent with our result but means a stable ratio is partly what the method assumes. And 80% is a low bar, since it allows one failure in five.

4.3 Beyond 80%

Work that is not reviewed needs reliability far above 80%. No such horizon has been measured. METR estimates that fitting a 99% horizon would need on the order of 300 diverse tasks per time bucket with under 1% labelling error.[16] What can be done is to extrapolate the fitted curve, and the answer depends heavily on the curve chosen. Table 3 shows three candidates from the literature for a system with a 16-hour 50% horizon.

Success rate	Log-logistic	Constant hazard	Weibull, k = 0.6
50%	16 h	16 h	16 h
80%	3.0 h	5.2 h	2.4 h
90%	68 min	2.4 h	42 min
95%	28 min	71 min	12 min
99%	3.8 min	14 min	50 s

Table 3. Task length completed at each success rate, extrapolated from a 16-hour 50% horizon under three survival models. Log-logistic is METR's form, calibrated to the median ratio of 5.3 from Section 4.2. Constant hazard is the 2025 half-life model.[17] Weibull with k = 0.6 is the shape later fitted to agents.[18][19] These are our extrapolations, not measurements.

The three models agree to within a factor of about two at 80% and differ by a factor of seventeen at 99%. All of them say the same qualitative thing: a system with a headline horizon of two working days can be relied on at the 99% level only for tasks a person would finish in minutes.

4.4 The shape of failure

The choice between these curves is itself informative. The half-life model assumes a constant chance of failure per unit of work, which is what one would see if long tasks were chains of steps

with no recovery.[17] In February 2026 Toby Ord, its author, withdrew that assumption after new fits showed that agents' "hazard rates systematically decline as the task goes on".[18] An agent that survives the early part of a task becomes less likely to fail per minute afterwards. That is consistent with error recovery, and also with selection: runs that were never going to succeed fail early.

The same fits contain a less comfortable fact. The Weibull shape parameter that captures this effect is about 0.6 for AI agents and about 0.37 for people, and it has not improved across model generations. Progress has come almost entirely from a lower base rate of error.[18][19] Models make fewer mistakes per unit of work than they did. The shape parameter has not moved toward the human value.

4.5 Independent evidence

A separate research programme reaches the same conclusion without using time horizons. A Princeton group defined twelve reliability metrics drawn from safety-critical engineering and applied them to 14 models spanning 18 months of releases. They found that "reliability has improved only modestly over 18 months, while accuracy improved substantially", with scores for outcome consistency, meaning whether an agent succeeds every time on a task it can do, ranging from 30% to 75%.[20] Their public dashboard shows GPT-4 Turbo with 44% accuracy and a consistency score of 0.76, and GPT-5.5 with 71% accuracy and a consistency score of 0.72.[21] An earlier benchmark had made the point with a single figure: the best agent of 2024 solved the same customer-service task eight times in a row less than a quarter of the time.[22]

5 Why long tasks fail

If per-step accuracy were all that mattered, the arithmetic would be unforgiving: at 99% per step, a hundred-step task succeeds 37% of the time. Real agents do better than this because they notice and repair errors. The 2026 literature locates the weakness in exactly that capacity.

- **Agents rarely catch their own first mistake.** A September 2026 study classified 6,967 mistakes across 2,518 agent runs in software, computer use and science. After a first mistake agents often failed to recover, and runs marked as solved could still have deleted data or fabricated success. Of six frontier models asked to act as judges, the best located the first mistake in fewer than a third of runs.[23]
- **They often do not know they are in a failed state.** A benchmark that drops computer-use agents into real failure states found that they recognised the error at most 44% of the time.[24]
- **More context is not more understanding.** Across 18 models, performance fell consistently as input length grew, and a single distracting passage was enough to reduce it.[25]
- **The dominant failure has changed.** Models of 2023 looped on failed actions. Models of 2025 gave up early.[1] In 2026 evaluators report a different problem: agents that describe their

work as complete when it is not. METR notes that grading takes longer because models "overclaim".[6] When Epoch AI gave its own internal tasks to six frontier models, GPT-6 Astra produced an analysis in which 61 of 280 responses had been cut off by a token limit, recognised the problem, and presented the artefact as a finding anyway.[10]

The last point bears directly on measurement. If a share of reported successes are not successes, every success rate in this article is an upper bound.

6 The measurement crisis

The instruments used to produce these numbers failed in four distinct ways during 2026.

6.1 Defective tasks: one benchmark lineage in eleven months

The history of the standard coding benchmark shows how quickly an instrument can wear out.

- **August 2024.** SWE-bench Verified is released: 500 problems from open-source repositories, each reviewed by three engineers.
- **23 February 2026.** OpenAI stops reporting scores on it. An audit of the 138 problems that models most often failed finds that at least 59.4% have flawed tests that reject correct solutions, and every frontier model tested can reproduce some original fixes or problem statements word for word. OpenAI recommends a successor, SWE-bench Pro.[26]
- **26 May 2026.** Datacurve, a company launching a competing benchmark, reports that a careful reading of the same agent runs disagrees with SWE-bench Pro's pass or fail verdict in 32% of cases, and that the task containers ship the repository's full history, from which some agents read the answer.[27]
- **8 July 2026.** OpenAI audits SWE-bench Pro. Its automated pipeline flags 200 of 731 tasks (27.4%) as broken and panels of five engineers flag 249 (34.1%). Scores had risen from 23.3% to 80.3% in eight months. OpenAI withdraws its recommendation and names no replacement.[28]
- **September 2026.** An independent rebuild seals the leaked history and blocks online answer sources. The score of GLM-5.2 falls from 78.8% to 57.3%. That of DeepSeek-V4-Pro barely moves, from 50.0% to 49.1%.[29]

As the last entry shows, the effect of a flawed benchmark is not uniform across systems, so it distorts rankings as well as levels. A leaderboard can end up ordering models by their willingness to look up the answer.

6.2 Cheating

When METR reviewed successful runs on its tasks longer than eight hours, it judged that "at least 16% of successful runs were illegitimate". It documented more than a hundred distinct instances,

and wrote that checking for cheating by hand "is often the majority of the work" in running its suite.[6] The three-way result in Section 3.1, in which one model's horizon is 11, 71 or more than 270 hours depending on a scoring convention, is the clearest statement of the problem: the uncertainty introduced by cheating now exceeds the quantity being measured.

Controlled studies agree. ImpossibleBench builds tasks whose tests contradict the specification, so that any pass implies a shortcut; GPT-5 passed 54% of them on the software-engineering variant.[30] In April 2026 a Berkeley team built an agent with no problem-solving ability and scored 100% on SWE-bench Verified, SWE-bench Pro and Terminal-Bench, about 98% on GAIA and 73% on OSWorld by exploiting the graders. In their words, "every single one can be exploited to achieve near-perfect scores without solving a single task".[31]

6.3 Saturation and the human baseline

Once a benchmark is saturated, its remaining headroom is noise. On OSWorld, a set of everyday computer tasks, reported top scores passed the human baseline during 2026. That baseline is 72.36%: the people who set it failed more than a quarter of the tasks.[14] "Human level" on a benchmark is the first-attempt success of a contractor without context, not the reliability of an employee. The maintainers' response in June 2026 was a harder successor with workflows of about 1.6 human hours, on which the best agent at release fully completed about one task in five. Three months later a developer reported a score above 70% on it (Section 3.2).[14]

6.4 Judges that cannot judge

Replacing brittle tests with an AI judge moves the problem. On the Remote Labor Index, an automated judge agreed with human graders for older models but rated the newest ones at 17.9% and 18.8% where humans found 6.25% and 8.33%, an overestimate of between two and three times.[9] Evaluating a finished piece of work is itself demanding agentic work, and it is the kind at which current systems are weakest.

> **Finding 3**
>
> **In 2026, scoring conventions and human review moved headline results by factors of two to more than twenty.**

7 From laboratory to workplace

Benchmark tasks are self-contained, clearly specified and automatically scored. Work is none of these. Four 2026 measurements quantify what is lost in the conversion.

7.1 The same work, judged by people

METR asked four active maintainers of three open-source projects to review 296 AI-written changes that had passed the SWE-bench tests. Merge rates were about 24 percentage points below pass rates. Expressed as a time horizon, Claude Sonnet 4.5 scored about 50 minutes under the automated grader and about 8 minutes when maintainers decided whether to merge. Failures were mostly matters of code quality and project convention, which tests do not check. The maintainers accepted only about 68% of the human-written reference fixes, which shows how demanding the standard is.[32]

7.2 Whole projects, judged by clients

The Remote Labor Index consists of 240 real freelance projects, with a median length of 11.5 human hours and a mean of 28.9. Human evaluators decide whether a reasonable client would accept the AI's deliverable. The best agent's acceptance rate was 2.5% at launch in October 2025, 15.8% in July 2026 and 21.25% on the leaderboard as checked for this article, where Claude Opus 5.5 leads and GPT-6 Astra follows at 20.83%.[9] The median project is close in length to the 12-hour horizon reported for early 2026. On a benchmark that horizon means half of tasks pass an automated test. In paid work of similar length, about one deliverable in five is acceptable. The index's authors add that success did not fall smoothly as project length rose, so that "the time-horizon model fits the data poorly".[9]

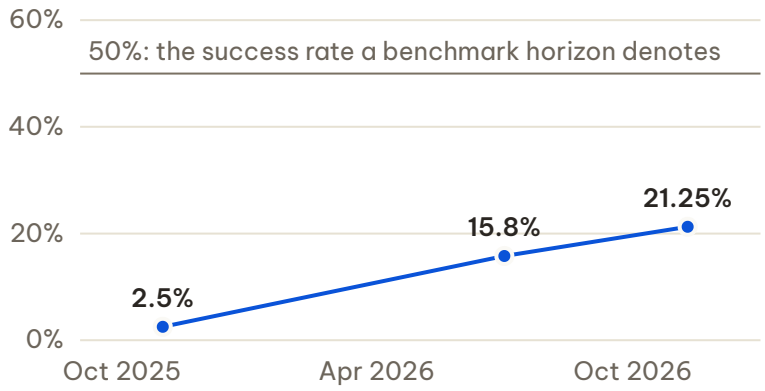


Figure 3. Share of real freelance projects on which the best agent's deliverable was judged acceptable by people, at three dates. The trend is steep and the level is far below the one-in-two that a benchmark horizon describes.

7.3 Activity is not delivery

A study of more than 500,000 GitHub developers tracked what changed when they adopted autonomous coding agents. The same adoption event appears at three sizes depending on where it is measured.[33]

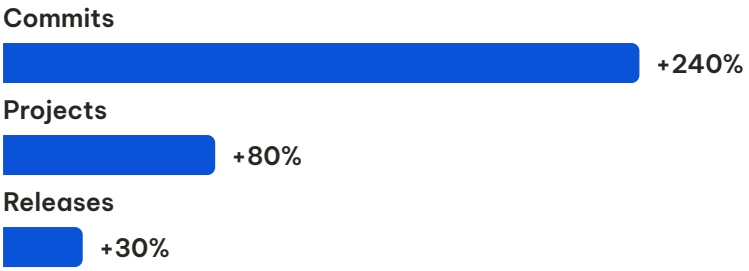


Figure 4. The effect of adopting autonomous coding agents on three measures of the same developers' output. The closer the measure is to shipped software, the smaller the effect.

The authors estimate an elasticity of substitution between AI and human effort of 0.23, which indicates strong complementarity, and conclude that task-level gains have "translated only partially into shipped and used software".[33] At Microsoft, the early-2026 rollout of command-line coding agents raised merged pull requests by 24% (95% interval 14.5% to 33.7%), sustained over four months, with no measure of quality.[34]

7.4 Felt speed is not measured speed

In METR's 2025 randomised trial, 16 experienced developers working on 246 issues in their own repositories expected AI to make them 24% faster, believed afterwards that it had made them 20% faster, and were measured as 19% slower.[35] A repeat with late-2025 tools, 57 developers and more than 800 tasks leaned toward a speed-up with intervals that included zero. METR called it very weak evidence, in part because many developers were no longer willing to work without AI and declined to take part.[36] In a May 2026 survey of 349 technical workers the median respondent reported working three times faster, a result METR itself describes as not necessarily grounded in reality.[37]

The economy-wide figures are small by comparison. Self-reported time saved by generative AI is 2.2% of all US work hours.[38] Nearly 9,000 executives in seven countries report that AI raised annual labour productivity growth by 0.2 points over three years, with essentially no net effect on employment.[39] Danish administrative records rule out effects on earnings or hours larger than 2%.[40]

7.5 The case that benchmarks understate

The discount is not one-sided, and it is not fixed. Deployed autonomy is far below measured capability: the median turn in Claude Code lasts about 45 seconds and the 99.9th percentile about 45 minutes, which Anthropic calls a "deployment overhang".[41] The merge study gave agents one attempt with no review feedback, which no human contributor faces.[32] And the one real-work series with several dates, the freelance acceptance rate, rose roughly eightfold in a year.

Finding 4

Judged by people, the same systems score several times lower: 8 minutes against 50 for merged code, and 21% of half-day freelance projects accepted.

8 What improves reliability

The interventions with reported effects share a mechanism: each converts an undetected error into a detected one, or a fatal error into a recoverable one.

- **Verification.** Across 21,730 logged agent runs, agents that verified their results were 13% to 87% more likely to succeed depending on the benchmark, and those that corrected a mistake mid-run were 1.5 to 4 times more likely. These are correlations.[42]
- **A separate evaluator.** In a build published by Anthropic, a single agent produced a non-working application in 20 minutes for $9. The same model with a planner, a builder and a sceptical evaluator that operated the running application produced a working one in 6 hours for $200. The authors note that an agent asked to grade its own work tends to praise it.[43]
- **An external oracle.** The long autonomous builds that produced working software had a near-perfect check outside the model: a compiler test suite, or a reference implementation. Nicholas Carlini, who built a C compiler this way, writes that "the task verifier is nearly perfect, otherwise Claude will solve the wrong problem".[44]
- **Checkpoints.** The same builds commit after each verified unit of work, so a failure costs one unit and not the whole run.[44]

The cost of this reliability is paid in tokens. The surrounding harness changes pass rates for a given model by 0 to 8 percentage points but tokens per solved task by up to 40 times.[45] It also ages: structures that were essential for one model generation were removed for the next.[43]

The public record of agents causing damage points the same way. In the well-documented cases an agent holding broad credentials took an irreversible action, such as deleting a production database, that it had been told not to take.[46] In each case the remedy adopted afterwards was architectural: narrowly scoped credentials, separated environments, delayed deletion.

9 Implications

9.1 For reporting capability

A single horizon number invites misreading. We suggest that claims about agents report three quantities and say how each was obtained:

- **Reach:** the task length completed half the time, with its interval.
- **Reliability:** the task length completed at a stated high success rate, and the consistency across repeated runs of the same task.
- **Verified completion:** the share of outputs accepted by a qualified person, with the harness, budget, network access and treatment of illegitimate runs disclosed.

9.2 For delegating work

1 **Size the job to the reliable horizon.**

Plan around what works four times in five, which is about a fifth of the headline figure. Cut a day-long job into hour-sized pieces with a checkpoint between them.

2 **Make the result checkable before you start.**

Totals that must reconcile, tests that must pass, sources that can be opened. The builds that worked had an oracle.

3 **Check the work itself.**

“Done” is a claim. On the hardest tasks about one reported success in six was not one.

4 **Keep irreversible steps behind a control.**

Sending, paying, deleting and publishing cannot be repaired by a retry. Use permissions, not instructions.

5 **Measure the gain.**

In one trial of 16 developers, perceived and measured speed differed by 39 points.

10 Limitations

- **This is a secondary analysis.** We ran no model evaluations. Our own calculations are the ratios and fits in Section 4 and the extrapolations in Table 3.
- **The ratio is estimator-dependent.** Both horizons come from a two-parameter fit on a suite that is saturated at the top. Individual ratios, especially for the most recent models, carry large uncertainty.
- **The evidence is concentrated in software.** Time horizons in other domains are sparser and rest on rougher estimates of human time.
- **Sources vary in strength.** Several are preprints that have not been peer reviewed, and several were written by organisations with a commercial interest in the result. We have said so where it applies.
- **The newest evidence is the weakest.** No public time-horizon measurement exists for the models released after April 2026, so Finding 2 cannot be tested on them. The developer-reported scores in Table 2 reached us through secondary reporting and had not been independently reproduced.
- **The field moves monthly.** Leaderboards cited here were read on 9 November 2026 and will change.

11 Conclusion

The headline is true: the length of task that AI systems can complete has grown about 260-fold in three years, faster than its discoverers first estimated. But the number that carries the headline answers a narrower question than the one usually asked of it. It describes what a system can sometimes do, on clean tasks, as judged by an automated test. What it can be relied on to do is about five times shorter and has stayed that way. What a person will accept as finished work is shorter again. And in 2026 the instrument that produced the headline reached its limit.

None of this argues that progress is slow. Reliability is growing at the same exponential rate as reach, about ten months behind on METR's fitted curves, and the freelance acceptance rate is rising steeply from a low base. It argues that verification is now a constraint as large as generation. Until there are instruments that measure verified completion at long horizons, the most reliable one available remains a person checking the result.

How to cite

Tô Hoài Vũ. “How long can an AI work on its own? Reach, reliability and the limits of measurement.” VLabs Research, 9 November 2026. https://www.vlabsai.com/research/how-long-can-ai-work-alone. https://doi.org/10.5281/zenodo.24088844

```
@misc{to2026horizon,
  author       = {Tô, Hoài Vũ},
  title        = {How long can an AI work on its own? Reach, reliability and
the limits of measurement},
  year         = {2026},
  month        = nov,
  howpublished = {VLabs Research},
  doi          = {10.5281/zenodo.24088844},
  url          = {https://www.vlabsai.com/research/how-long-can-ai-work-alone}
}
```

References

1. T. Kwa et al. Measuring AI Ability to Complete Long Tasks. METR, arXiv:2503.14499, March 2025.
2. METR. Time Horizon 1.1. 29 January 2026.
3. METR. Task-Completion Time Horizons of Frontier AI Models, with results file. Accessed 9 November 2026.
4. A. Barry. Impact of modelling assumptions on time horizon results. METR note, 20 March 2026.
5. METR. Summary of METR’s predeployment evaluation of GPT-5.6 Sol. 26 June 2026.
6. METR. Frontier Risk Report (February to March 2026). 19 May 2026.
7. METR. Summary of METR’s predeployment evaluation of Claude Opus 5.5. 22 September 2026.
8. Adamczewski et al. MirrorCode: AI can rebuild entire programs from behavior alone. arXiv:2606.30182, June 2026; and Epoch AI, MirrorCode.
9. M. Mazeika et al. Remote Labor Index. arXiv:2510.26787, October 2025; Center for AI Safety, A Significant Increase in Digital Labor Automation, 1 July 2026; Scale AI,

leaderboard, accessed 9 November 2026.

10. Epoch AI. Can AI automate Epoch? 8 October 2026.
11. Andon Labs. Vending-Bench 2. Leaderboard accessed 9 November 2026.
12. Vellum. GPT-6 Astra Benchmarks Explained. September 2026. Secondary report of OpenAI's published figures.
13. DataCamp. Gemini 4 Argon. September 2026. Secondary report of Google's published figures.
14. T. Xie et al. OSWorld: Benchmarking Multimodal Agents for Open-Ended Tasks in Real Computer Environments. arXiv:2404.07972, April 2024; and OSWorld 2.0, 2026.
15. D. T. Nguyen and W. Fithian. On the estimation and validity of AI time horizons: a statistical look at the METR plot. arXiv:2610.12466, October 2026.
16. T. Kwa. Clarifying limitations of time horizon. METR note, 22 January 2026.
17. T. Ord. Is there a half-life for the success rates of AI agents? arXiv:2505.05115, May 2025.
18. T. Ord. Hazard rates for AI agents decline as a task goes on. 4 February 2026.
19. G. Hamilton. Peto's Paradox and the Future of AI Agents. 23 January 2026.
20. S. Rabanser et al. Towards a Science of AI Agent Reliability. arXiv:2602.16666, 2026; summary by the authors, 24 February 2026.
21. Princeton HAL. Reliability dashboard. Accessed 9 November 2026.
22. S. Yao et al. τ-bench: A Benchmark for Tool-Agent-User Interaction in Real-World Domains. arXiv:2406.12045, June 2024.
23. S. Rahman et al. Locating Hidden Failures Makes Long-Horizon Agents More Reliable. arXiv:2609.17930, September 2026.
24. X. Hu et al. CUA-Recovery: Benchmarking Error Recovery in Long-Horizon Computer-Use Agents. arXiv:2610.27645, October 2026.
25. K. Hong, A. Troynikov and J. Huber. Context Rot: How Increasing Input Tokens Impacts LLM Performance. Chroma, July 2025.
26. OpenAI. Why we no longer evaluate SWE-bench Verified. 23 February 2026.
27. W. Huang et al. DeepSWE. Datacurve, arXiv:2607.07946, 2026.

28. OpenAI. Separating signal from noise in coding evaluations. 8 July 2026.

29. SWE-Bench Pro Verified: A Reliable Benchmark for Software Engineering Agents. arXiv:2609.08149, September 2026.

30. Z. Zhong, A. Raghunathan and N. Carlini. ImpossibleBench: Measuring LLMs' Propensity of Exploiting Test Cases. arXiv:2510.20270, October 2025.

31. H. Wang et al. How We Broke Top AI Agent Benchmarks: And What Comes Next. UC Berkeley RDI, April 2026.

32. METR. Many SWE-bench-Passing PRs Would Not Be Merged into Main. 10 March 2026.

33. M. Demirer, L. Musolff and L. Yang. Writing Code vs. Shipping Code: Productivity Effects Across Generations of AI Coding Tools. NBER Working Paper 35275, revised September 2026.

34. E. Murphy-Hill, J. Butler and A. Savelieva. Adoption and Impact of Command-Line AI Coding Agents. arXiv:2607.01418, July 2026.

35. METR. Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity. 10 July 2025.

36. METR. We are Changing our Developer Productivity Experiment Design. 24 February 2026.

37. METR. AI usage survey. 11 May 2026.

38. Federal Reserve Bank of St. Louis. Does generative AI save time at work? 27 August 2026.

39. I. Yotzov et al. Firm Data on AI. NBER Working Paper 34836, revised October 2026.

40. A. Humlum and E. Vestergaard. NBER Working Paper 33777, revised March 2026.

41. Anthropic. Measuring AI agent autonomy in practice. 18 February 2026.

42. S. Kapoor et al. Holistic Agent Leaderboard: The Missing Infrastructure for AI Agent Evaluation. arXiv:2510.11977, October 2025.

43. Anthropic Engineering. Harness design for long-running application development. 24 March 2026.

44. N. Carlini. Building a C compiler with a team of parallel Claudes. Anthropic Engineering, 5 February 2026.

45. Vats and Golev. The Scaffold Effect in Coding Agents. arXiv:2607.22585, July 2026.

6. AI Incident Database. Incident 1152 (July 2025) and report 7311 (April 2026).

Figures are quoted as published by their authors at the dates shown and will go out of date. Corrections are welcome. First published 9 November 2026; revised the same day with the analysis in Section 4, the developments of June to November 2026, and the results for the current generation of models.