

Healthy AI First, Episode 4: How Much Can Run at Once

Sources and evidence for this episode

At twenty to eleven, three of Jonas's four agents reach for him at once, and Miriam loses track of her own quarterly plan across three AI chats. The episode explains attention residue, debunks the myth of the "23 minutes", uses Little's law to show why more agents with the same review capacity mostly mean longer waits, and proposes a resumption point for every switch plus a limit for the team. The tool tip: parallel agents in separate Git worktrees.

All links were checked on 17 October 2026.

What we mention in this episode

What Stays Behind

- **Claim:** Leaving a task unfinished leaves attention residue and lowers performance on the next task; finishing under some time pressure makes letting go easier.
Source: Leroy, S. (2009): Why is it so hard to do my work? The challenge of attention residue when switching between work tasks. *Organizational Behavior and Human Decision Processes* 109(2), 168-181.
Link: <https://doi.org/10.1016/j.obhdp.2009.04.002>
Strength and limits: Two lab experiments; a solid mechanism, but no fixed time penalty and no optimal number of agents can be derived.
- **Claim:** Expected time pressure on returning increases the residue; a short plan for returning reduces it and improves focus on the new task.
Source: Leroy, S. & Glomb, T. M. (2018): Tasks Interrupted: How Anticipating Time Pressure on Resumption of an Interrupted Task Causes Attention Residue and Low Performance on Interrupting Tasks and How a "Ready-to-Resume" Plan Mitigates the Effects. *Organization Science* 29(3), 380-397.
Link: <https://doi.org/10.1287/orsc.2017.1184>
Strength and limits: Four studies; applying it to agent work (resumption point) is a proposal of the series.

The 23 Minutes

- **Claim:** Knowledge workers stayed with one topic for a little over ten minutes on average; interrupted tasks were mostly resumed the same day, after almost half an hour and more than two other activities in between.
Source: Mark, G., Gonzalez, V. M. & Harris, J. (2005): No Task Left Behind? Examining the Nature of Fragmented Work. CHI 2005.
Link: <https://doi.org/10.1145/1054972.1055017> (PDF: <https://ics.uci.edu/~gmark/CHI2005.pdf>)
Strength and limits: Observation of 24 people over three days each; describes work patterns, not a concentration penalty.
- **Claim:** The popular "23 minutes" come from an interview and refer to the time until returning to the interrupted task, not the time until concentration is back.
Source: Gallup Business Journal (2006): Too Many Interruptions at Work? Interview with Gloria

Mark.

Link: <https://news.gallup.com/businessjournal/23146/too-many-interruptions-work.aspx>

Strength and limits: Interview, not a study of its own; the study itself reports 25 minutes 26 seconds.

- **Claim:** Participants answered emails with and without interruptions. In the interrupted conditions, active task time was shorter after time spent on interruptions was subtracted; the study found no statistically significant difference in errors or politeness. After about twenty minutes, however, they reported more stress, frustration and time pressure.

Source: Mark, G., Gudith, D. & Klocke, U. (2008): The Cost of Interrupted Work: More Speed and Stress. CHI 2008, 107-110.

Link: <https://doi.org/10.1145/1357054.1357072>

Strength and limits: Lab experiment with 48 participants, mostly German university students; a short email task.

Small Portions

- **Claim:** DORA, the research program at Google Cloud, considers any batch too big that takes longer than a week to complete and check, and names huge AI-generated pull requests as a typical trap.

Source: DORA: Working in small batches.

Link: <https://dora.dev/capabilities/working-in-small-batches/>

Strength and limits: Research-based practice guidance, not a causally derived limit. The review block of 15 to 30 minutes is a pilot rule of the series.

The Queue at the Pass

- **Claim:** In a stable system, on average, work in progress equals throughput times time in the system; if more arrives than can be reviewed, the queue grows every day.

Source: Little, J. D. C. (1961): A Proof for the Queuing Formula: $L = \lambda W$. Operations Research 9(3), 383-387; Little, J. D. C. (2011): Little's Law as Viewed on Its 50th Anniversary. Operations Research 59(3), 536-549.

Link: <https://doi.org/10.1287/opre.9.3.383> · <https://doi.org/10.1287/opre.1110.0940>

Strength and limits: A mathematical theorem under assumptions (long-run averages), not an experiment; the worked example in the episode is invented.

- **Claim:** When tasks arrive unevenly or take different amounts of time, long queues can form when reviewers are nearly fully busy, not only when they are overloaded.

Source: Kingman, J. F. C. (1961): The single server queue in heavy traffic. Mathematical Proceedings of the Cambridge Philosophical Society 57(4), 902-904.

Link: <https://doi.org/10.1017/S0305004100036094> (overview: https://en.wikipedia.org/wiki/Kingman%27s_formula)

Strength and limits: Queueing theory; an approximation, used in the script without numbers.

- **Claim:** Once a column's WIP limit is reached, nothing new is added; existing work gets finished first.

Source: DORA: Work in process limits.

Link: <https://dora.dev/capabilities/wip-limits/>

Strength and limits: Kanban practice; no empirically optimal number of parallel agents.

Separate Working Copies

- Claim:** A Git worktree is a separate working copy on its own branch; parallel agents then do not overwrite each other's files, but each copy only sees its own state.
Source: Git documentation git-worktree; Claude Code: Run parallel sessions with worktrees.
Link: <https://git-scm.com/docs/git-worktree> · <https://code.claude.com/docs/en/worktrees>
Strength and limits: Official documentation.
- Claim:** When two branches change the same part differently, Git may report a conflict; changes that are incompatible in behavior (semantic conflicts) can still merge cleanly. So review the changes together and run appropriate tests on an integrated version; tests do not catch every conflict either. So merge early and often; DORA says at least once a day.
Source: Fowler, M.: Semantic Conflict; DORA: Trunk-based development; Git documentation git-merge.
Link: <https://martinfowler.com/bliki/SemanticConflict.html> · <https://dora.dev/capabilities/trunk-based-development/> · <https://git-scm.com/docs/git-merge>
Strength and limits: Practitioner knowledge and documentation.
- Claim:** The instruction file that the agent reads at every start is context, not a lock.
Source: Claude Code: How Claude remembers your project; OpenAI Codex: Custom instructions with AGENTS.md.
Link: <https://code.claude.com/docs/en/memory> · <https://learn.chatgpt.com/docs/agent-configuration/agents-md>
Strength and limits: Vendor documentation.

The Resumption Point

- Claim:** A short plan for returning reduces attention residue (see Leroy and Glomb above); five to six lines are a budget to try.
Source: Leroy & Glomb (2018), see above; line budget from the basis of the series, chapter 7.
Strength and limits: The budget is explicitly a proposal to test, not a research result.
- Claim:** Offloading thinking to tools relieves us, but we often remember offloaded information less well.
Source: Risko, E. F. & Gilbert, S. J. (2016): Cognitive Offloading. Trends in Cognitive Sciences 20(9), 676-688.
Link: <https://doi.org/10.1016/j.tics.2016.07.002>
Strength and limits: Research review; effects depend on task and use.

A Limit for the Team

- Claim:** A visible limit for waiting reviews on the board and a trial with a fixed end.
Source: Derived from DORA Work in process limits (see above) and Little's law.
Strength and limits: Proposal of the series; the example number "three" is an example, not a source.

Tool tip: separate Git worktrees for parallel agents

- Claude Code with `claude --worktree <name>` (short `-w`), a second terminal with a different name:
<https://code.claude.com/docs/en/worktrees> · <https://code.claude.com/docs/en/common-workflows>
- Git basics: <https://git-scm.com/docs/git-worktree>

- Rules such as "never change field names without asking" in CLAUDE.md or AGENTS.md (context, not a lock; real locks via hooks): <https://code.claude.com/docs/en/memory> · <https://code.claude.com/docs/en/hooks> · <https://learn.chatgpt.com/docs/agent-configuration/agents-md>

Further reading

- Parnin, C. & Rugaber, S. (2011): Resumption Strategies for Interrupted Programming Tasks. Software Quality Journal 19(1), 5-34. About 10,000 programming sessions from 86 programmers. <https://doi.org/10.1007/s11219-010-9104-9>
- Trafton, J. G., Altmann, E. M., Brock, D. P. & Mintz, F. E. (2003): Preparing to resume an interrupted task: effects of prospective goal encoding and retrospective rehearsal. International Journal of Human-Computer Studies 58(5), 583-603. [https://doi.org/10.1016/S1071-5819\(03\)00023-5](https://doi.org/10.1016/S1071-5819(03)00023-5)
- Google Engineering Practices: Small CLs. About 100 lines usually reasonable, about 1,000 usually too big, no fixed limit. <https://google.github.io/engineering-practices/review/developer/small-cls.html>
- Sadowski, C., Söderberg, E., Church, L., Sipko, M. & Bacchelli, A. (2018): Modern Code Review: A Case Study at Google. ICSE-SEIP 2018. <https://doi.org/10.1145/3183519.3183525>
- SmartBear: Best Practices for Code Review (vendor claim, 200 to 400 lines at a time; only partly supported). <https://smartbear.com/learn/code-review/best-practices-for-peer-code-review/>
- Basis of the series: Healthy AI First Software Development, version 2 (1 October 2026), chapters 5 and 7.

Note

Healthy AI First is AI-researched and AI-generated, narrated in Björn's voice, recreated with AI, and reviewed by him.

Jonas, Daniel and Miriam are fictional characters used as illustrations; their scenes and the worked example for Little's law are invented.