

The Bottleneck Moved

This piece refines the treatment of risk-spotting in The Team That No Longer Fits, which placed it among the irreducibly human evaluation activities. The two are reconciled by the degree of externalisation: risk-spotting is mechanical where the relevant knowledge has been externalised into tests, conventions, and recorded history, and human where it remains tacit.

1. The bottleneck moved

When AI tooling accelerated code production, it did not make engineering faster in proportion. It moved the constraint. The work of producing code, which used to consume most of an engineer's time, can now be done in a fraction of it. But the work of reviewing that code has not accelerated at all, because review depends on human attention, and human attention reads at the same speed it always did. The result is that review has become the bottleneck, and most teams have not noticed, because they are still treating review as a step rather than as the constraint.

The symptom is familiar. Changes arrive for review larger and more frequently than they used to. Reviewers cannot keep up. A large change sits in a queue, or worse, it is approved without being properly examined, because reviewing it thoroughly would take longer than the team can afford. The common responses are to demand smaller changes or to speed up the integration pipeline. Neither addresses the problem, because the problem is not the size of the change or the speed of the pipeline. It is that the review process was designed for a rate of production that no longer exists, and it has not been redesigned.

This piece argues that the single-reviewer model, in which one person reads an entire change and approves it, is obsolete for AI-accelerated work. It argues that review is the current bottleneck precisely because it is the step that has not yet been addressed with tooling, and that the response is not to ask reviewers to read more or faster, but to redesign review around AI tooling the same way production was. Continuous integration remains sound. Small batches and fast pipelines remain worth having. What has to change is the human review step, which is the part of the process that scales with human attention and therefore became the constraint the moment production stopped scaling that way.

2. Why review became the constraint

The practices of modern code review were calibrated, implicitly, to a rate of production. An engineer produced a bounded amount of code in a day because producing code by hand took time. A reviewer could read that day's output and form a judgement about it within a reasonable fraction of their own day. The single-reviewer model worked because the thing being reviewed was naturally limited in size by the speed at which it could be produced. Review capacity and production capacity were roughly matched, not by design, but because both were performed by humans working at human speed.

AI tooling broke the match. Production now happens faster than a human can produce by hand, and often faster than a human can read. The reviewer's capacity did not change, because reviewing is reading, and reading is the one part of the process AI acceleration did not touch. The same reviewer who could keep pace with one engineer's hand-written output now faces several times the volume, in larger units, arriving faster. The single-reviewer model has not failed because reviewers became worse. It has failed because the thing it was calibrated against has multiplied.

This is why review is the bottleneck and not, say, the pipeline. The integration pipeline runs on compute, and compute scales. When the pipeline is too slow, you provision more of it, and the constraint eases. Review runs on human attention, and human attention does not scale by provisioning more of it, because reading a change is not parallelisable across the hours of a single reviewer's day. The pipeline can be made to keep pace with production by spending money. Review cannot, as long as it is modelled as one human reading everything. The constraint is structural, not a matter of effort or resourcing within the existing model.

3. Why this is not a ceiling

The natural conclusion is pessimistic: if review depends on human attention and human attention does not scale, then review is a permanent ceiling on how fast engineering can go, and the gains from accelerated production are capped by the rate at which humans can check the output.

This conclusion repeats the mistake that was made about production before AI. Production, too, was once assumed to be irreducibly human. Writing code was something only a person could do, line by line, and the rate of engineering was assumed to be bounded by how fast people could write. AI tooling falsified that assumption by showing that large parts of production were mechanical enough to be supported or performed by tooling, leaving the human to direct and judge rather than to type. The bottleneck moved because the task turned out to be more decomposable than it looked.

Review is decomposable in the same way. What a reviewer does is not a single indivisible act of judgement. It is a bundle of distinct activities, some of which require human judgement and many of which do not. Checking that a change adheres to conventions does not require judgement; it requires pattern-matching against a known standard. Verifying that test coverage maps to the lines that changed does not require judgement; it requires comparison. Identifying the regions of a large diff most likely to contain problems does not require judgement where the knowledge of where problems tend to occur has been externalised, into tests, into conventions, into the recorded history of past failures; there it requires analysis of the change against that record, which tooling can perform. Where that knowledge remains tacit, held only by experienced engineers, it stays a matter of human judgement. Summarising what a change actually does, so that a human reviewer can orient quickly, does not require judgement; it requires comprehension of the diff, which tooling can now produce.

What does require human judgement is a smaller set: whether the change is the right change, whether it fits the architecture it is joining, whether it introduces the kind of subtle correctness problem that no automated check can express. These are the parts of review that depend on substrate

and experience, and they are not yet tooling-addressable. But they are a fraction of what currently consumes a reviewer's attention. The reviewer spends most of their time on the mechanical surface and has less left for the judgement that only they can provide. Review is slow and incomplete not because the judgement is slow, but because the judgement is buried under mechanical work that should not require a human at all.

4. What AI-supported review does

If review is decomposable, then the response to the review bottleneck is the same as the response to the production bottleneck: build tooling that handles the mechanical part and let the human spend their attention on the judgement part. This does not remove the human from review. It does the opposite. It clears the mechanical work out of the way so that the scarce resource, human judgement, lands where it actually matters.

The functions such tooling serves are reasonably clear, even where the specific tools are still maturing.

Tooling can summarise a change before a human reads it, producing an account of what the change does, which parts of the system it touches, and how its pieces relate. A reviewer who begins with an accurate summary orients in minutes rather than reconstructing the structure of the change from the diff alone.

Tooling can surface the parts of a change that most warrant human attention. Not all of a large change is equally risky. The portions that touch sensitive logic, that deviate from established patterns, that lack corresponding tests, or that resemble past changes which caused problems, are where human judgement should be spent. Tooling that flags these directs the reviewer's scarce attention rather than leaving them to spread it evenly across a change that is not evenly risky.

Tooling can verify the mechanical properties that currently consume review time. Convention adherence, test coverage mapping to the change, the presence of documentation for new interfaces, the absence of known anti-patterns. These are checks a human currently performs by reading, and they are checks tooling can perform more reliably and instantly, leaving the human to assume they are handled rather than to perform them.

Tooling can support the human side of review rather than only the machine side. A reviewer working through a large change can use tooling to ask questions of the change, to trace how a value flows through it, to understand why a particular decision was made, without having to reconstruct all of it manually. The review becomes an interrogation supported by tooling rather than an unaided read.

With these functions in place, the human review process itself can be redesigned around them. Review can be layered, with different reviewers taking the concerns they understand best, each supported by tooling that has already surfaced where to look. Review can be structural, focused on the shape and fit of the change rather than its every line, because the line-level mechanical checks are handled by tooling. And the author can walk reviewers through a change supported by a

tooling-generated account of it, so that the walk starts from a shared and accurate understanding rather than from the author's recollection. The human work that remains is the work that needs humans. The rest is handed to the tooling that can do it.

5. The pattern

What has happened with review is what happened with production, and it is worth naming the pattern directly, because it is likely to happen again.

A task that was assumed to be irreducibly human turns out, under pressure, to be decomposable. Most of it is mechanical and can be supported or performed by tooling; a smaller part requires judgement and remains human. When tooling addresses the mechanical part, the task ceases to be the bottleneck, and the constraint moves to whatever is next. Production was the first such task. Review is the current one. The constraint will move again, to whatever depends most heavily on unaugmented human attention once review has been addressed, and the response then will be the same as the response now: identify the mechanical part, build the tooling that handles it, and reserve human judgement for the part that genuinely requires it.

The mistake at each stage is to treat the current bottleneck as permanent and to respond by demanding more human effort against it. Asking reviewers to read more and faster is the review-stage version of asking engineers to type more and faster. It misunderstands the nature of the constraint. The constraint is not that humans are not trying hard enough. It is that human attention is being spent on work that does not require it, and the response is to move that work to tooling so that human attention is freed for the work that does.

6. Closing

Review is the bottleneck that AI-accelerated production has exposed. The single-reviewer model, in which one human reads an entire change and approves it, was calibrated to a rate of production that no longer exists, and it does not scale to the volume and size of change that tooling now generates. The common responses, smaller changes and faster pipelines, do not address this, because the constraint is the human review step, which scales with attention rather than with compute.

This is not an argument against continuous integration, which remains sound, nor against fast pipelines, which remain worth building. It is an argument that review is the next part of the engineering process to be redesigned around tooling, the same way production already was. Review decomposes into mechanical work that tooling can handle and judgement work that requires a human. Building the tooling that handles the mechanical part is what lets human judgement be spent where it matters, and it is what turns review from a ceiling on throughput into a step that keeps pace with the production it was always meant to check.

The work is not to ask reviewers to read more. It is to build the workflows that let them read less, and judge more.