When is a faster program still correct?

Reference-output checks

Coral	PLENA*
Integer arithmetic	Floating point / quantized
Exact match	**Within tolerance**

Extra checks: 3 PLENA kernels · 13 input conditions

KernelCraft §3.2 · Appendix C.2 · Table 13

* Programmable Long-context Efficient Neural Accelerator (PLENA)

Close scores, different decisions

Hypothetical example

Kernel	Scores	Categorisation (A vs B)
Reference	0.5001 / 0.4999 A B	**Selects: A**
Candidate	0.4999 / 0.5001 A B	**Selects: B**

What must match before deployment?

Context: KernelCraft, Appendix C.2 · This is not a reported failure.

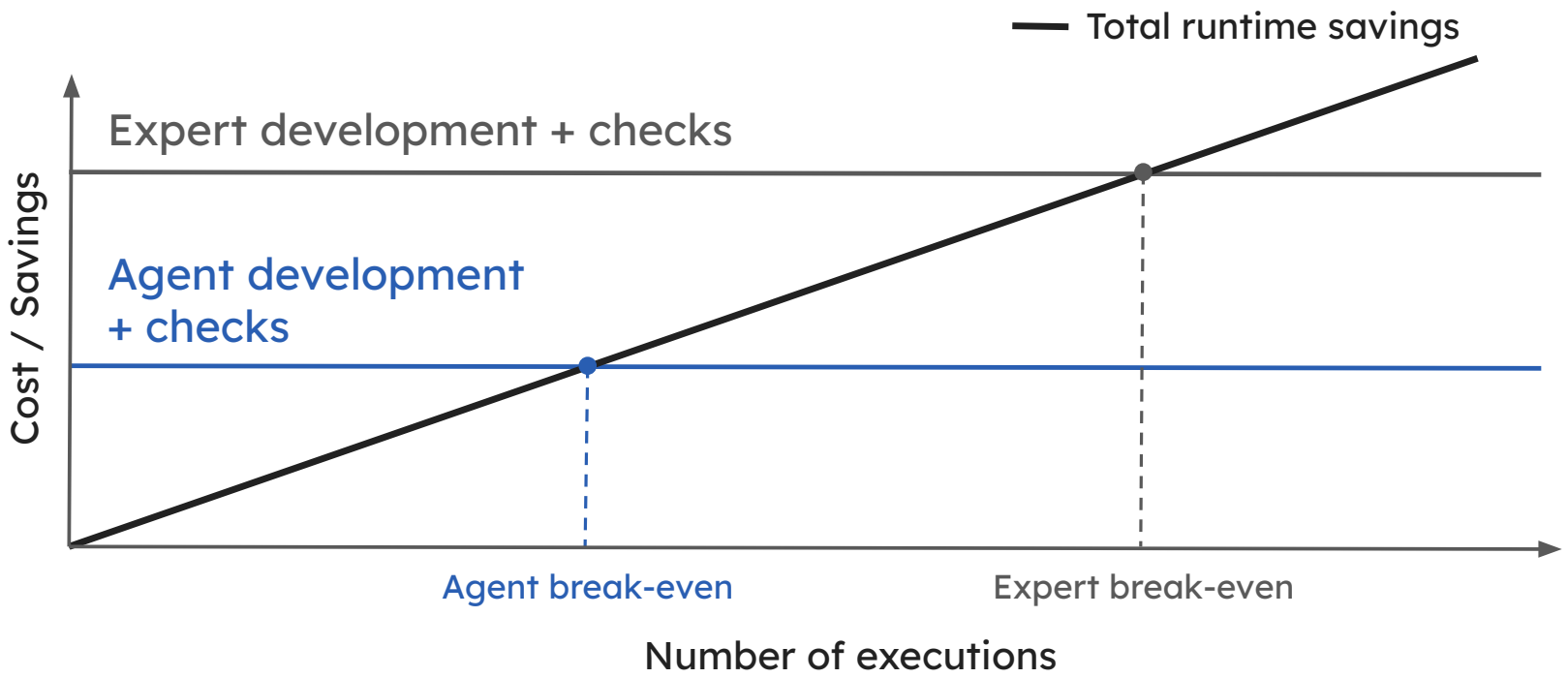
What should the agent change?

Kernel	Compiler	Hardware
Generate assembly for a workload	Improve a template 2.9% lower latency*	Propose instructions Human supplies specs

Where can AI actually help? And where do we still need to think for ourselves?

KernelCraft §4 · §5.1 (*configuration C2) · §5.5 & Appendix I.2

Does AI lower the bar for specialisation?



KernelCraft §1: reducing development effort · Table 11: runtime quality still matters