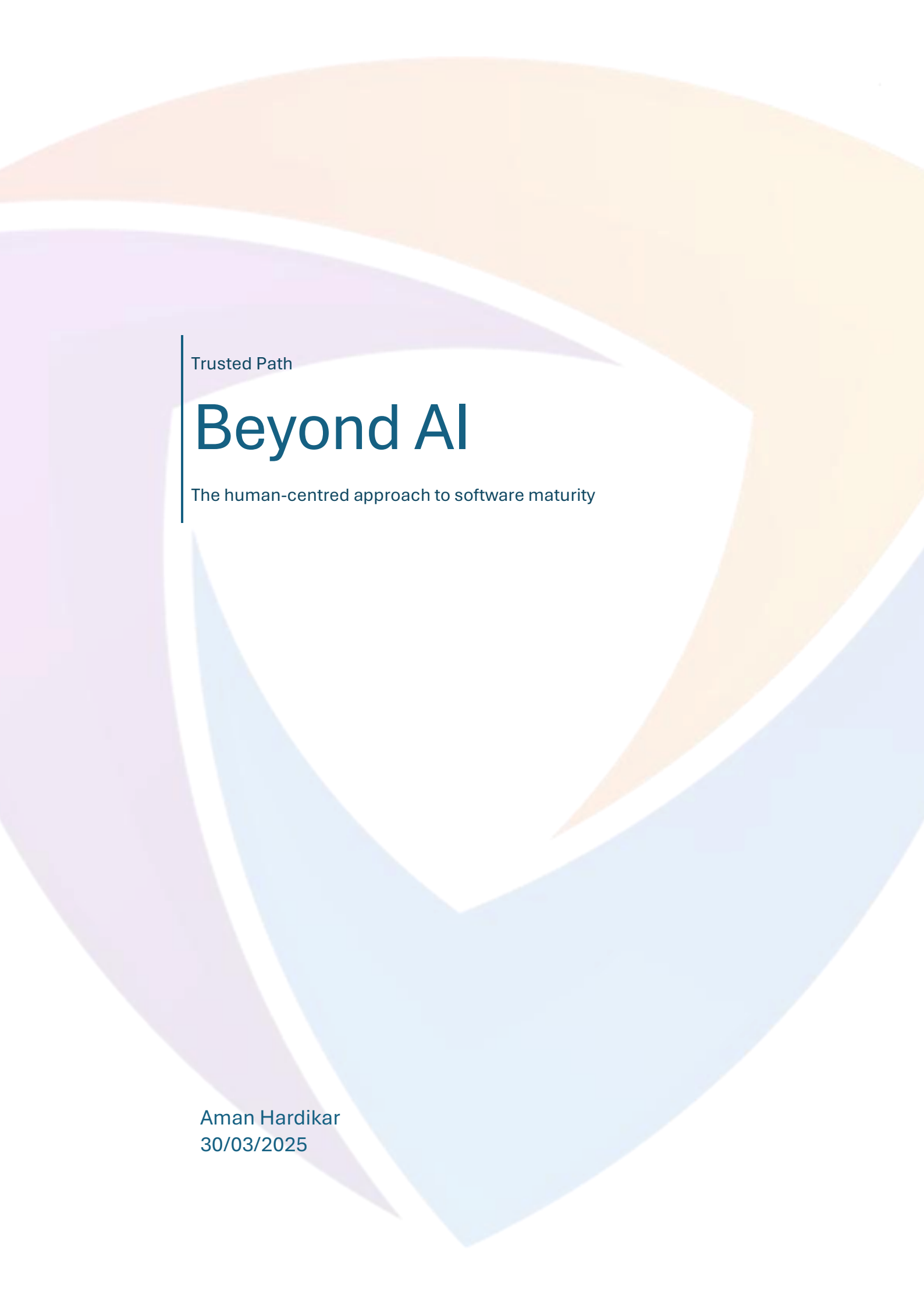




Trusted Path

Beyond AI

The human-centred approach to software maturity

Aman Hardikar
30/03/2025



IN BRIEF

AI coding assistants and AI security tools are now part of everyday engineering work. They are fast, and for some jobs they are very good. The question I keep hearing from engineering and security leaders in regulated firms is whether these tools can carry software security maturity on their own.

My answer is no, and the reason isn't that the models are weak. Maturity belongs to an organisation: how it decides which risks it will accept, how it builds those decisions into design, and how it learns when something goes wrong. A model can help with each of those steps. It can't own them.

This article sets out eight places where AI-only approaches fall short, using published research rather than vendor claims. It then looks at where AI does earn its place, and finishes with practical steps for leaders who want the speed of AI without losing control of their security, privacy and resilience posture.

THE QUESTION

The pitch is simple. If a model can review code, suggest controls and draft a threat model in minutes, why spend months building a security programme?

It's a fair question. Large language models are good at finding patterns across huge amounts of text, and a lot of security work involves exactly that. They spot common weakness patterns, explain unfamiliar code and turn rough notes into readable documentation.

The evidence on AI-generated code is less comfortable. When researchers at New York University asked GitHub Copilot to complete 89 security-relevant scenarios, about 40% of the 1,689 programs it produced were vulnerable [1]. A Stanford user study went further. Developers with an AI assistant wrote less secure code than those without one on most of the tasks studied, and they were more likely to believe their code was secure [2].

That second finding is the one that worries me. Faster output combined with weaker scrutiny is the opposite of maturity.

WHERE AI-ONLY APPROACHES FALL SHORT

1. A tool is not a method

AI produces findings and suggestions. Maturity needs a method for deciding which findings matter, who owns them and how practice improves from one release to the next. Without



one, teams collect a growing pile of recommendations and have no reliable way to tell whether they are getting better.

That is why Trusted Path is built around the TRUST framework. It describes five levels of practice mapped to CMM levels 1 to 5: Tribal, where security lives in a few people's heads, then Repeatable, Uniform and Steered, up to Tuned, where practice is measured and adjusted on evidence. A model can help you assess a control. Deciding that your organisation needs to move from Repeatable to Uniform this year, and agreeing what that will cost, is a leadership call.

2. Context lives outside the prompt

A model only knows what you give it. It doesn't know that a legacy payments system can't take a library upgrade until the supplier certifies it, that your resilience team has already set an impact tolerance for a critical service, or why an architecture decision was made three years ago.

Regulation adds another layer. The same control can mean different things for a bank under the EU Digital Operational Resilience Act, which has applied since 17 January 2025 [6], and for a health provider handling patient records under UK GDPR. Getting that interpretation right depends on people who know the business, and on a record of decisions that a model can read but didn't make.

3. Your design data has to go somewhere

To review a design, a model needs the design: architecture diagrams, data flows, threat models and sometimes source code. In a regulated firm that material is among the most sensitive information the organisation holds.

Before any of it goes into an AI service, someone has to answer some plain questions:

- Where is the data processed and stored?
- Is it retained, and for how long?
- Can the provider use it to train future models?
- Who at the provider can see it?

Many enterprise AI services now give good answers, with contractual no-training terms and regional hosting. The point is that someone in your organisation has to ask, check the answers against your data protection obligations and record the decision. That is governance work, and it stays with you.

4. A first draft is not progress

AI is very good at first drafts: a starting threat model, a control list, a review checklist. Maturity is what happens after the first draft. You can see it when last quarter's design



review changes this quarter's defaults, when an incident leads to a better pattern in the architecture library, and when the same finding stops coming back.

That needs memory and feedback: what was recommended, what was built, whether it worked and what changed as a result. Most AI tools answer the question in front of them and keep no record of whether their last answer helped. Trusted Path is built to keep that record across the lifecycle, so each new project starts from what the organisation has already learned instead of a blank page.

5. Models are confidently wrong in new ways

In security work, hallucination is more than an annoyance. A study of 16 code-generating models found that 19.7% of the software packages they recommended did not exist [3]. More than half the time, a hallucinated package came back again when the same prompt was repeated. An attacker who registers one of those names on a public registry gets a ready-made route into your software supply chain.

Models also have a training cutoff, so they know least about the newest attack techniques, which are often the ones you most need to hear about. Threat intelligence teams and practitioner networks still tend to spot new attack patterns first.

6. AI tools are part of your attack surface

As teams connect models to IDEs, pipelines and ticketing systems, and give them permission to act, the tools themselves need securing. The OWASP Top 10 for LLM Applications, published in November 2024, ranks prompt injection first and acknowledges that no foolproof prevention is currently known [4]. It also lists excessive agency, where a model has more permissions or autonomy than its task needs.

The UK government's Code of Practice for the Cyber Security of AI, published on 31 January 2025, makes the same point from a governance angle [5]. Its principles include designing AI systems for security as well as function, and keeping humans responsible for them. You can't hand the job of securing AI to the AI.

7. Good answers need the expertise you were hoping to replace

The Stanford study had a second finding that gets less attention. Participants who trusted the assistant less and put more effort into their prompts ended up with more secure code [2]. That fits my own experience over twenty years in security: the people who get the most from AI security tooling already know enough to judge what it tells them.

AI can raise the output of an experienced engineer. It doesn't supply the experience. Answers also shift with phrasing and model version, so two teams asking the same question can get different advice, and consistency across a portfolio suffers. Structured questions, agreed standards and recorded evidence fix that. Cleverer prompting on its own does not.

8. Maturity is cultural, and culture is human



The hardest part of security maturity has never been tooling. It is getting developers, security, operations, compliance and leadership to agree on priorities and share ownership of the outcome. It is making it safe for an engineer to raise a problem early instead of hiding it. And it is changing what gets rewarded.

No model will chair the meeting where a product owner and a security architect agree to accept a risk, write down why and set a date to revisit it. Tools can make that meeting shorter and better informed. The decision, and the accountability for it, stay with people.

WHERE AI EARNS ITS PLACE

None of this is an argument against AI. It is an argument for putting AI where it is strong and keeping people where judgement is needed. AI adds real value when it is used for:

- reading design documents and diagrams and flagging patterns that deserve a closer look
- suggesting candidate controls for a reviewer to accept, change or reject
- explaining security and privacy requirements in plain language for engineers who aren't specialists
- drafting documentation and evidence from work that has already been done

In each case the model proposes and a named person decides. That is the rule we follow when we use AI inside Trusted Path.

HOW TRUSTED PATH APPROACHES IT

Trusted Path helps engineering teams build secure, private and resilient software from the start of a project, so they aren't sent back for reviews late in delivery, when changes cost the most.

At the plan and design stages, teams work through structured checks covering security, privacy and resilience. At build and deploy, the platform works with the tools teams already use and covers the gaps those tools leave. Dashboards show where each project and team sits against the TRUST framework, so leaders can see progress instead of guessing at it.

WHAT TO DO NEXT

If you lead engineering or security in a regulated organisation, start here:

- Require a named human reviewer for any AI output that changes a design, a control or a risk decision.



- Treat AI tools as part of your attack surface. Assess them against the OWASP Top 10 for LLM Applications and limit what they are allowed to do.
- Decide which design data may go to which AI service, write that decision down and check it against your data protection obligations.
- Check every AI-suggested dependency against an approved registry or allow-list before anything is installed.
- Measure maturity over time, not the number of findings. If the same issues keep coming back, the tool isn't the problem.

AI will keep improving, and on some of these points the gap will narrow. The gap on accountability won't, because it was never a technical gap. The organisations that get the most from AI will be the ones that already know what good looks like and can tell when the model has missed it.

ABOUT TRUSTED PATH

Trusted Path builds an SDLC maturity platform for regulated industries, including financial services, healthcare and critical national infrastructure. It treats security, privacy and resilience as engineering disciplines and measures progress against its TRUST framework.

Trusted Path was founded by Aman Hardikar Muppidi, who has more than 20 years' experience in cyber security across security architecture, cloud security, governance, risk and compliance, and application security.

To see how Trusted Path would work on one of your projects, ask us about a pilot.

REFERENCES

- [1] Pearce, H. et al. (2022) 'Asleep at the Keyboard? Assessing the Security of GitHub Copilot's Code Contributions', IEEE Symposium on Security and Privacy. arXiv:2108.09293.
- [2] Perry, N., Srivastava, M., Kumar, D. and Boneh, D. (2023) 'Do Users Write More Insecure Code with AI Assistants?', ACM Conference on Computer and Communications Security (CCS '23). doi:10.1145/3576915.3623157.
- [3] Spracklen, J. et al. (2024) 'We Have a Package for You! A Comprehensive Analysis of Package Hallucinations by Code Generating LLMs'. arXiv:2406.10279.
- [4] OWASP GenAI Security Project (2024) OWASP Top 10 for LLM Applications 2025, published 18 November 2024. genai.owasp.org.
- [5] Department for Science, Innovation and Technology (2025) Code of Practice for the Cyber Security of AI, published 31 January 2025.



[6] Regulation (EU) 2022/2554 on digital operational resilience for the financial sector (DORA), applicable from 17 January 2025.

© 2025 Trusted Path Ltd. Company No. 16189138. All rights reserved.