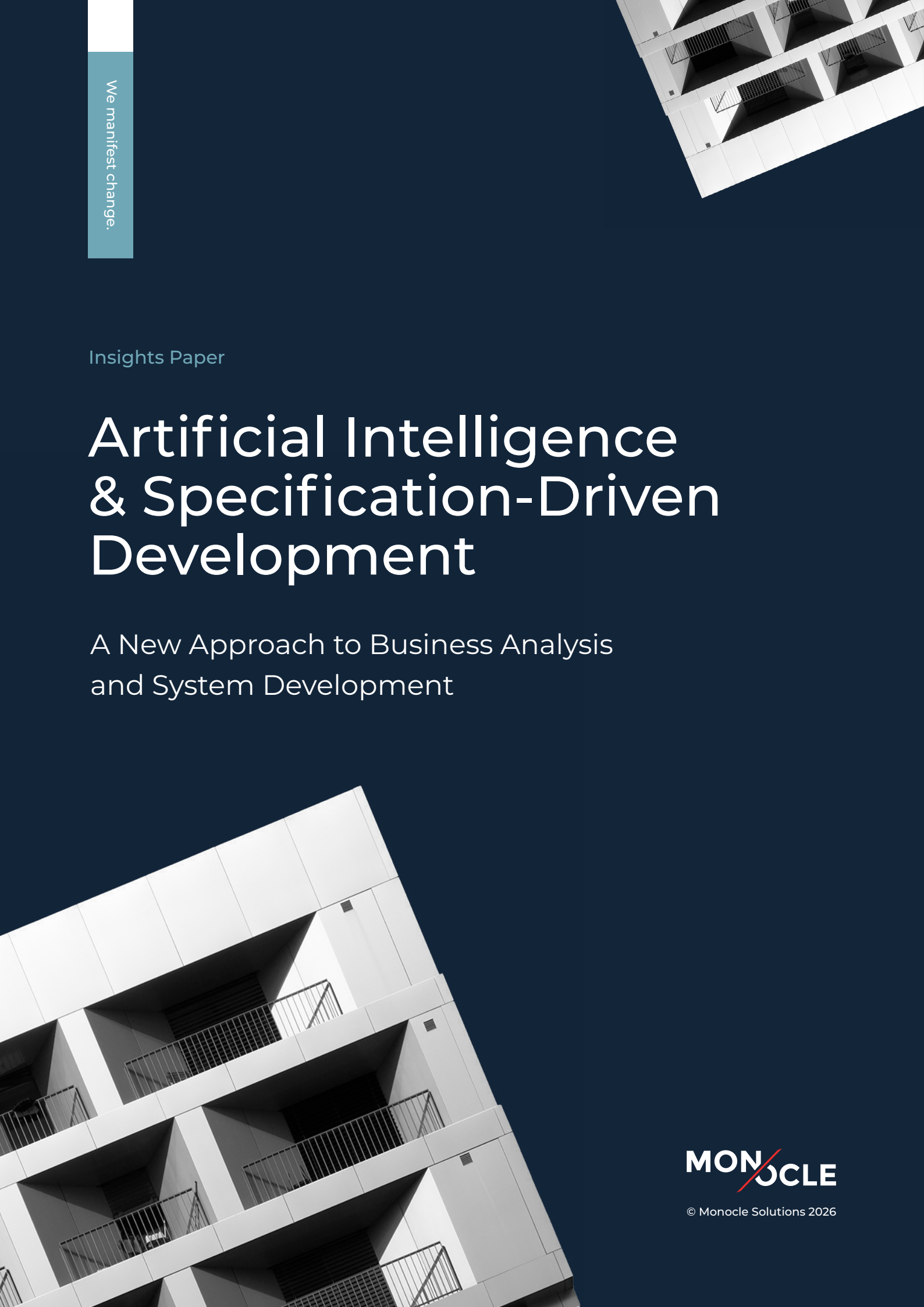




We manifest change.

Insights Paper

Artificial Intelligence & Specification-Driven Development

A New Approach to Business Analysis
and System Development

MONOCLE

© Monocle Solutions 2026

Software development has traditionally relied on human centred business requirements specifications (BRSs) to define business needs before development begins. However, this is prone to inefficiencies such as scope-creep, constant iterative adjustments and other inconsistencies that make it harder to keep requirements reliable, traceable, and connected throughout the software development life cycle (SDLC).

Specification-driven development (SDD) is an evolution of the traditional way of managing requirements across the typical business requirement, development and testing process. Instead of treating documentation as something that is created and then handed over to developers, SDD puts a structured specification at the centre of the development process. Combining this approach with the proven power of artificial intelligence through AI-coding tools presents a use case for striking productivity gains across technical design, development tasks, testing, and coding.

Converting AI Predictions into AI Proof

In Monocle's experience, a spec-driven development approach utilising AI coding tools has the potential to compress a typical requirements gathering and development process from 10 weeks to 3 days, allowing teams to work faster and shift their time to more high-value tasks.

WHAT IS SPECIFICATION-DRIVEN DEVELOPMENT

Specification or spec-driven development is an approach where detailed specifications are used to guide the software development process from start to finish. Instead of developers having to interpret BRSs on their own, business analysts, architects, and developers work together to create a clear specification. This specification explains what the system needs to do, how it should be designed, what the acceptance criteria are, and what needs to be done to implement it.

Combining this methodology with AI-assisted coding and development tools is where the real value lies. These tools enforce the spec-driven development approach to then create technical designs, development plans, and even production-ready code that can be largely driven through agentic AI. In fact, AWS who offer their own agentic AI-coding tool, Kiro, found they could complete projects 80% faster with 85% fewer resources.¹

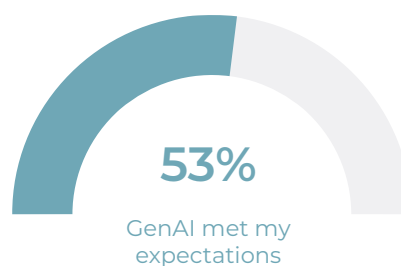
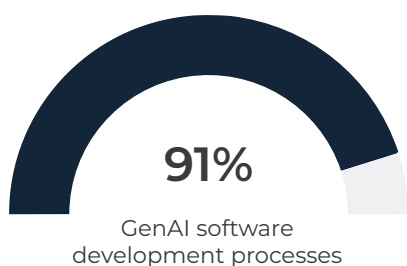
Traditional BRS vs Specification-Driven Development

Traditional BRS	Specification-Driven Development
Static business documents	Living specification
Multiple handovers between teams	Single source of truth
Developers interpret requirements	AI and developers use the same specification
Documentation often becomes outdated	Documentation evolves with implementation
Testing follows development	Testing derived directly from the specification
Limited traceability	End-to-end traceability
Higher risk of requirement drift	Improved governance and consistency

These efficiencies reduce the amount of repetitive and tedious work analysts and developers have to do while making sure the development stays aligned with the original business requirements.

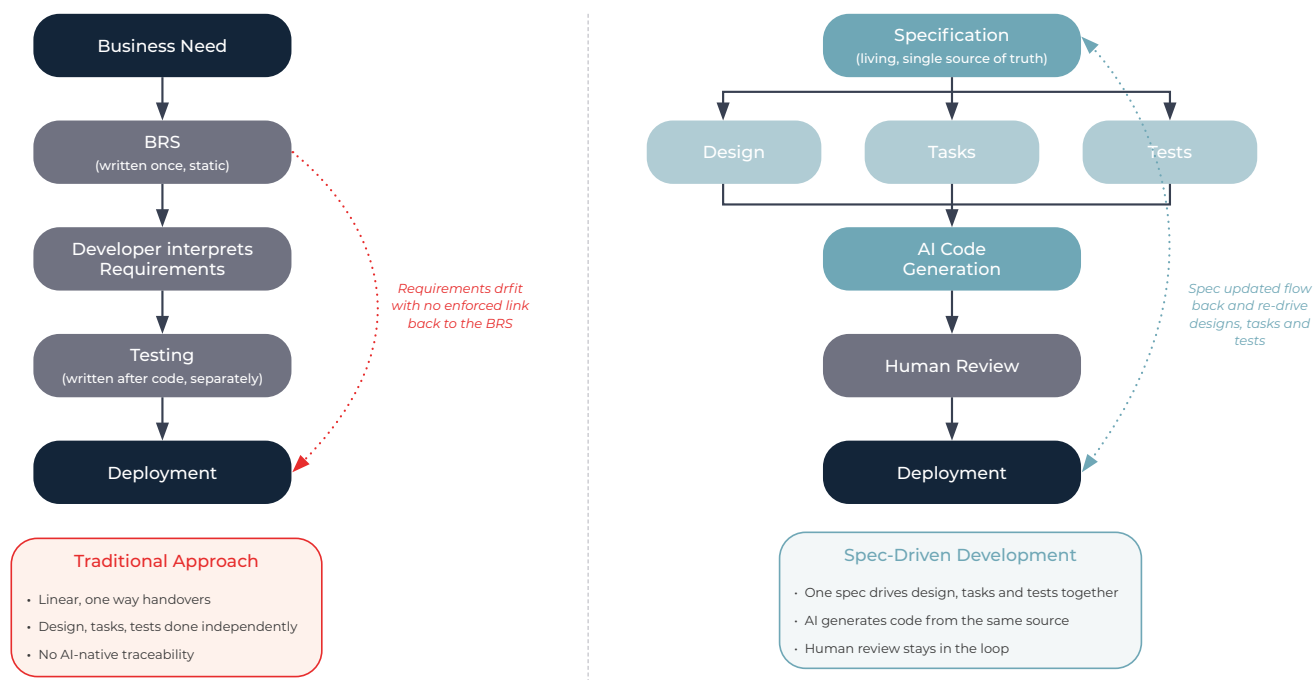
The traditional human centred approach to BRS writing now shifts as business analysts prompt SDD AI-tools in order to draft the various requirements. With these AI-tools given system and data context through access to relevant technical documentation and repositories, well-constructed prompts compress drafting cycles to mere days rather than weeks or months. The role of the business analyst then becomes centred around critical review and guidance of the AI output, which means business and system analysts will need to improve their quality judgement and domain knowledge to adequately understand and critically review AI outputs.

Two Sides of the Same Coin: Financial services practitioners are experiencing the benefit of generative AI but output has not yet met expectations.ⁱⁱ



TRADITIONAL VS SPEC-DRIVEN DEVELOPMENT

In the traditional approach, work moves through a one-way chain of handovers. The BRS is read once, and each team then interprets it in its own way, which is where requirements can start to drift. With SDD, the design, tasks, and tests all stay linked to the same specification. AI-generated code still goes through human review, and any changes are fed back into the specification, so it does not become outdated as soon as development starts.



RISKS AND LIMITATIONS

While spec-driven development combined with AI-coding tools has clear productivity benefits, there are significant hurdles to overcome to ensure real efficiencies as well as to manage various risks that arise when using generative AI.

- **More Effort Upfront:** SDD and the use of AI requirements and coding tools is dependent on sufficient granularity and accuracy of requirements which should be sourced across multiple stakeholders at the beginning of the process. Efficiency gains from AI are hindered by poor context, vague prompts and undefined requirements that are often a direct result of poor coordination upfront.
- **Comprehensive Context:** AI requirement and coding tools require up-to-date and detailed context regarding the systems they are drafting and developing for. Teams should ensure documentation repositories, that contain contextual information used by these tools, are well maintained to ensure reliable context for AI.
- **Detailed and Clear Prompting:** If the prompts to draft a specification are vague or inconsistent, AI tools may fill in the gaps on their own and produce code that looks right but is at best superficial and at worst detrimental to the solution. Upfront planning minimises uncertainty, but users need to be trained to be specific in their prompts and critical in their review.
- **Data and IP Exposure:** The repositories, schemas and requirements that give these tools useful context are often an organisation's most sensitive material. Teams should confirm where that context is processed and retained, what vendor terms permit, and whether inputs are used for model training.
- **Skills and Tooling Gaps:** Writing specifications that are clear enough for AI is not the same as writing a traditional BRS. Many business and system analysts as well as solution architects may still need training, particularly when prompting the AI and reviewing its output. This includes detecting omissions and hallucinations, separating genuine business requirements from implementation detail the model has inferred, and challenging output rather than simply accepting it.
- **Review Fatigue:** Generative AI can produce output extremely quickly and compress the SDLC significantly, but someone must still review it thoroughly and properly. If review becomes rushed or treated as a checkbox exercise, SDD loses much of its value. Review is rarely regarded as engaging work, and sustained exposure to it risks a decline in reviewer attentiveness and, in turn, review quality.

Review of "almost right" AI output, especially complex and materially important code, has become a significant headache for developers, which can make debugging and review more time-consuming.ⁱⁱⁱ It is therefore critical these obstacles are understood and addressed early as part of any AI-enabled SDD approach.

Users need to be trained to be **specific** in their prompts and **critical** in their review.

HOW MONOCLE CAN ASSIST

As AI becomes more common in software engineering, spec-driven development is likely to become a more widely used approach in organisations. Future software projects may rely less on manually written code and more on clear, structured specifications that guide AI through the development lifecycle. Organisations that want to move in this direction should start small and identify a viable and achievable use case. A good first step is to pilot SDD on a few selected projects, create internal standards for writing specifications, train business analysts and architects, and connect the specifications into existing governance and DevOps processes.

At Monocle we are actively assisting our financial services clients in exploring, utilising and embedding artificial intelligence in their organisations through our AI offerings including AI strategy, AI enterprise embedment, agentic solution delivery, as well as AI governance and risk management. With 25 years of industry experience, we understand both the requirements of business as well as the technical requirements of technology stakeholders, enabling us to deliver measurable AI-driven value.

Contact us to learn more about our AI-accelerators, toolkits and frameworks that showcase key, viable AI use cases in financial services and provide a platform for rapid AI delivery and enterprise embedment.

i. AWS, 2026

ii. DefineX, 2025

iii. Stackoverflow, 2025

We manifest change.



Johannesburg

8th Floor, The MARC Tower 1
129 Rivonia Road, Sandton,
Johannesburg, 2196

info.za@monoclesolutions.com

+27 (0) 11 263 5600

Cape Town

Ratanga Precinct,
10 Elephant Lane,
Century City, Cape Town

info.za@monoclesolutions.com

+27 (0) 21 007 5979

London

51 Moorgate,
London EC2R 6LL,
United Kingdom

info.uk@monoclesolutions.com

+44 (0) 2030 022 514

Amsterdam

Office 136, Gustav Mahlerplein 2,
1082 MA, Amsterdam,
Netherlands

info.nl@monoclesolutions.com

+31 (0) 2 02 256 182



www.monoclesolutions.com