

From Legacy Code to AI-Ready:

Agentic Modernization for Custom Software

Strategies and Practice for CIOs and IT Decision-Makers

adesso |

CONTENT

- 3 | When yesterday's technology blocks tomorrow's strategy
- 4 | Why now: The convergence of drivers
- 6 | Why traditional modernization is reaching its limits
- 8 | How AI is changing modernization:
 - 10 | Phase 1
 - 13 | Phase 2
- 16 | Human in the Loop:
Not an optional feature, but a design principle
- 17 | adesso as your partner for application modernization

When yesterday's technology blocks tomorrow's strategy

Two out of three companies run business-critical applications that are so outdated they hinder the effectiveness of the entire organization.¹

Die Modernisierung ist keine offene Frage mehr – sie ist eine laufende Krise. Modernization is no longer an open question – it is an ongoing crisis.

Technological development cycles in the context of AI are accelerating at a pace that was hard to imagine just a few months ago. What was yesterday an AI-powered tool for drafting emails now automates entire workflows. What was experimental two years ago is now a productive standard. This momentum doesn't just affect new technologies—it fundamentally changes the benchmark by which existing systems are measured.

This has major consequences for custom software developed ten, fifteen, or twenty years ago. COBOL-based software systems, enterprise Java applications based on J2EE, JEE, JSF, Struts, or EJB 2.x, and custom solutions on older .NET stacks—these systems still form the backbone of business-critical processes in many companies. Nearly half of all companies run applications that are between 11 and 20 years old, and more than one in ten even use software that is 21 to 30 years old.² Forty percent of these aging legacy systems are proprietary, custom-built solutions for which no software vendor provides a migration path.³

What makes custom software unique is that companies built it to create a specific competitive advantage—and it did exactly that. No off-the-shelf product could have mapped these differentiating business processes so perfectly. But unlike standard software, there is no vendor working in the background to deliver updates, define migration paths, or create urgency through end-of-support deadlines. With custom software, this responsibility lies entirely with the company itself.

In practice, this often means the issue goes unaddressed until the pressure to act becomes critical. Maintenance costs creep upward, regulatory updates demand far more effort than necessary, and innovative initiatives—such as AI integration, cloud migration, and new digital services—stall because core systems lack the necessary connectivity. This isn't a flaw in the custom software itself; it is the direct consequence of lacking a continuous modernization mechanism.

This white paper focuses on modernizing these exact systems: custom software built on legacy code, regardless of the specific technology.

While the urgency to modernize has existed for years, the tools have fundamentally changed. Today, AI-driven approaches make modernization projects faster, more cost-effective, and more manageable than ever before. This white paper outlines the available strategies and demonstrates how an agentic approach is reshaping the economics of modernization.

¹ Lünendonk Study "IT Modernization between Legacy, Cloud, and AI", 2025

² <https://www.der-bank-blog.de/veraltete-legacy-systeme-technologien/studien/37724967/>

³ <https://www.thinkwisesoftware.com/de/blog/neue-studie-legacy-modernisierung-von-computerwoche>

Why now: The convergence of drivers

Several developments are currently colliding, and their combination creates a pressure to act that companies can no longer ignore.

AI requires modern system architectures.

For many companies, artificial intelligence has evolved from a mere experiment into a strategic priority. AI thrives on data—structured, accessible, and available in real time. It also demands systems that can seamlessly integrate AI services via APIs and event streams. Legacy applications simply cannot deliver this. They store data in proprietary formats, communicate through outdated protocols, and lack the interfaces required for modern AI integration. If you want to leverage AI effectively, you must first modernize your core systems. Without this foundation, AI strategies remain nothing more than slide decks.

New business requirements are outgrowing the capabilities of legacy systems.

Digital business models, self-service portals, real-time data processing, platform integration—business departments expect new features in weeks, not months. Legacy systems structurally cannot keep pace. This is not because they were poorly built, but because they were designed for an era of longer release cycles and minimal integration requirements. Modernizing your core systems is therefore a prerequisite for almost every goal on your strategic agenda.

The window of opportunity for knowledge-driven modernization is closing.

Demographic shifts are not an abstract forecast— they are a lived reality in IT departments. When the last experts for a legacy system retire, they take more than just technical know-how with them. They also take the understanding of the business rules hardcoded into the software. Why does the system calculate this edge case this way? What regulatory requirement drives this validation logic? This knowledge is rarely documented. It exists implicitly—in the minds of people who will soon leave the company. If you wait until these experts are gone, you will be modernizing blindly. Every year of delay makes the transition not only more expensive, but far riskier.

Regulatory requirements are tightening rapidly.

NIS2 introduces strict new cybersecurity standards, while DORA mandates operational digital resilience for financial institutions. At the same time, GDPR enforcement is growing increasingly strict. Industry-specific regulations—ranging from BaFin requirements to public sector transparency mandates—continue to expand. Systems must do more than just work; they must be demonstrably secure, auditable, and compliant. Under these conditions, legacy systems with patchy documentation and incomplete test coverage become a major liability—not just for the IT department, but directly for executive management.

The economic equation is shifting.

Maintaining aging systems drains budgets that are desperately needed elsewhere. Industry analyses consistently show that maintaining existing systems consumes 60 to 80 percent of IT spending. The older the systems, the larger this share becomes—and the less room remains for innovation. Modernization is not just an added expense. Done right, it shifts the balance between "run" and "change" budgets in favor of the future.

Any one of these factors would be a powerful driver on its own. Combined, they make one thing clear: the risk of doing nothing now far outweighs the risk of modernization.

[Executive Box]

Application modernization is not just an IT issue. Stalled AI strategies, regulatory liability, demographic knowledge loss, and shrinking room for innovation put it squarely on the board's agenda. Those who fail to act now risk not only rising costs, but strategic paralysis.

Drivers for application modernization

Technological Drivers

- Technology stack EoL
- No security updates
- Lack of skilled talent
- Difficult & costly maintenance
- Risk of system failures
- Integration issues
- Data silos

Business Drivers

- Time-to-market
- Flexibility for new market opportunities
- Customer expectations
- Competitive pressure
- Cost optimization

Regulatory & Security Drivers

- Compliance
- Cybersecurity
- Auditability

Strategic Drivers

- Digital transformation of business models
- Integration of innovative technologies like AI
- Business continuity
- User acceptance

Why traditional modernization is reaching its limits

In the past, modernization projects were tedious, resource-intensive, and highly risky. Analysts spent months analyzing code, writing documentation, conducting interviews, and mapping dependencies. While thorough, this approach simply doesn't scale. Effort increases almost linearly with the size of your application landscape. For large environments with dozens or hundreds of systems, this leads to analysis phases dragging on for months before you can even make a decision. Project teams remain tied up for long periods without modernizing a single line of code, and budgets run dry during the analysis phase—long before the actual transformation even begins.

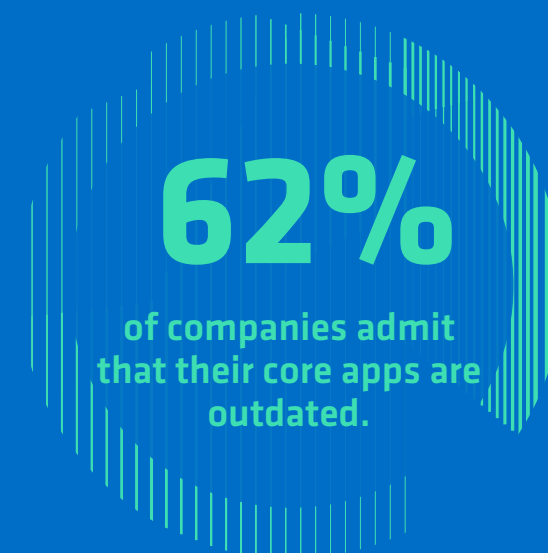
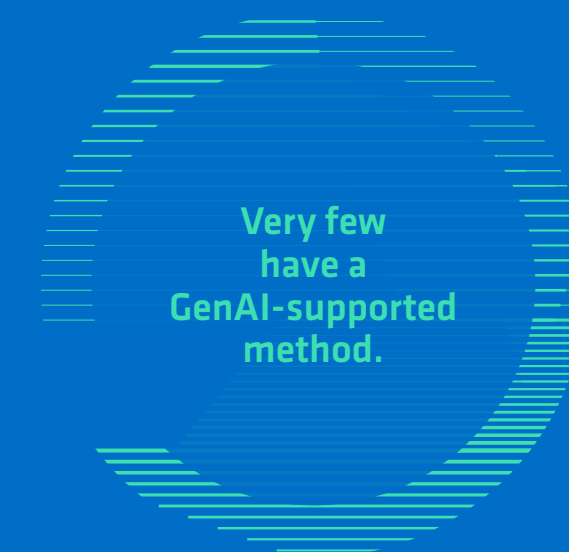
This is precisely why many companies repeatedly put off modernization, even when they know it is necessary: the projects are simply too massive, expensive, and risky to manage alongside day-to-day operations.

We also see a recurring pattern in practice: modernization projects that fail during implementation—despite exhaustive analysis—because teams underestimated the true complexity of their legacy systems. Hidden dependencies only surface during migration. Critical business rules buried deep in legacy

code never made it into the documentation, only to be missed when the new system fails to replicate them. After months of hard work, teams realize their chosen strategy is unviable and have to start all over again.

Even when projects do cross the finish line, many approaches only solve part of the problem. Rehosting moves applications to the cloud without addressing their underlying weaknesses—you simply migrate your technical debt to a new infrastructure. Syntactic code converters translate code into a new language without improving the architecture. The code compiles in a modern framework, but the monolithic structure, lack of testability, and opaque dependencies remain. This creates what we call "new legacy in new technology"—a system that looks modern but is just as difficult to evolve as its predecessor. Meanwhile, greenfield rebuilds consistently underestimate the deep functional complexity and implicit rules of the legacy system, resulting in a new system that actually does less than the one it was meant to replace.

What has been missing until now is an approach that combines the depth of manual analysis with the speed and scalability of automated processing.



How AI is changing modernization: adSCALE, the agentic approach from adesso

Until now, application modernization was primarily one thing: complex. Understanding, evaluating, and transforming legacy systems with hundreds of thousands of lines of code tied up highly qualified specialists for months or years. This is exactly what made modernization expensive – and led companies to repeatedly postpone it.

With adSCALE, adesso changes this equation. What is adSCALE?

adSCALE stands for adesso Smart Cycle for AI-enhanced Lightweight Software Engineering – a proven process framework that brings together AI agents and human expertise in a structured, repeatable cycle. The framework includes best practices, models, and standards for defining, implementing, and optimizing agentic workflows in software engineering.

The core principle: humans orchestrate, AI executes. Specialized AI agents handle repetitive intellectual tasks – analysis, code investigation, documentation, and dependency detection. Humans retain strategic control: they validate results, prioritize applications, define target states, and make business-critical decisions. The result is ultra-short development cycles, faster iterations, better quality and a sustainable integration of agentic AI approaches. adSCALE is based on over 25 years of software engineering experience, deep industry-specific domain knowledge, and more than six

adSCALE

years of experience with AI technologies. This combination of methodological maturity and technological competence makes the difference between an AI experiment and an industrially robust modernization approach.

adSCALE can also massively accelerate embedded software development.

In safety-critical and industrial systems, traceability, certifiability, and data sovereignty determine success.

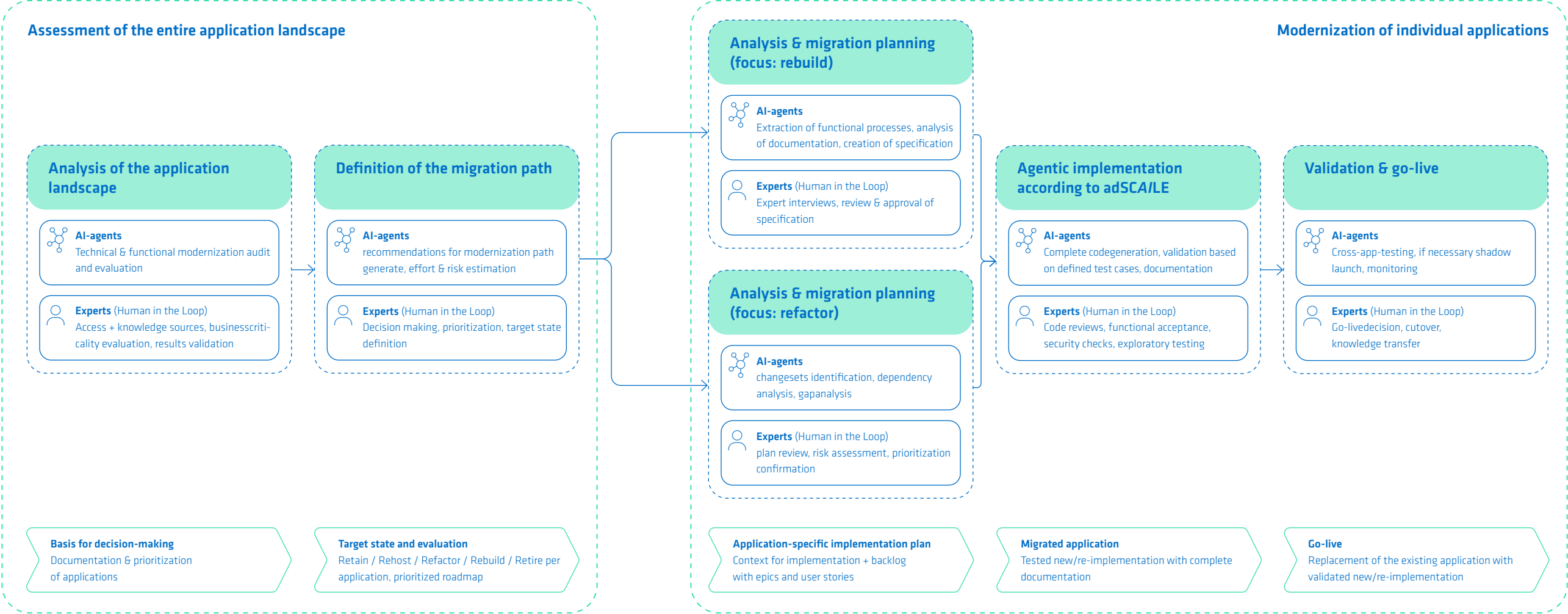
While traditional software projects primarily need to meet functional requirements, additional success criteria apply to embedded and real-time systems: deterministic runtime behavior, hardware dependencies, regulatory requirements, and the protection of intellectual property.

Especially in standardized environments (e.g., IEC 61508 or IEC 62304), qualified professionals must document and approve AI support. adSCALE brings the benefits of agentic software development to these unique environments. The framework combines AI-powered development with human-in-the-loop governance, complete traceability, and standards-compliant quality assurance. This not only accelerates development cycles but also sustainably improves auditability, compliance, and technical quality.

adSCALE generates test cases, regression tests, and test data. Final verification in embedded environments then takes place on the target hardware, on simulators, or in a HIL environment.

adSCALE helps development teams in the embedded sector create, test, and document certifiable and traceable software much more efficiently.

Agentic-first approach to application modernization →



Phase 1: Understanding and assessing the application landscape

Step 1: ----->

Analysis of the application landscape

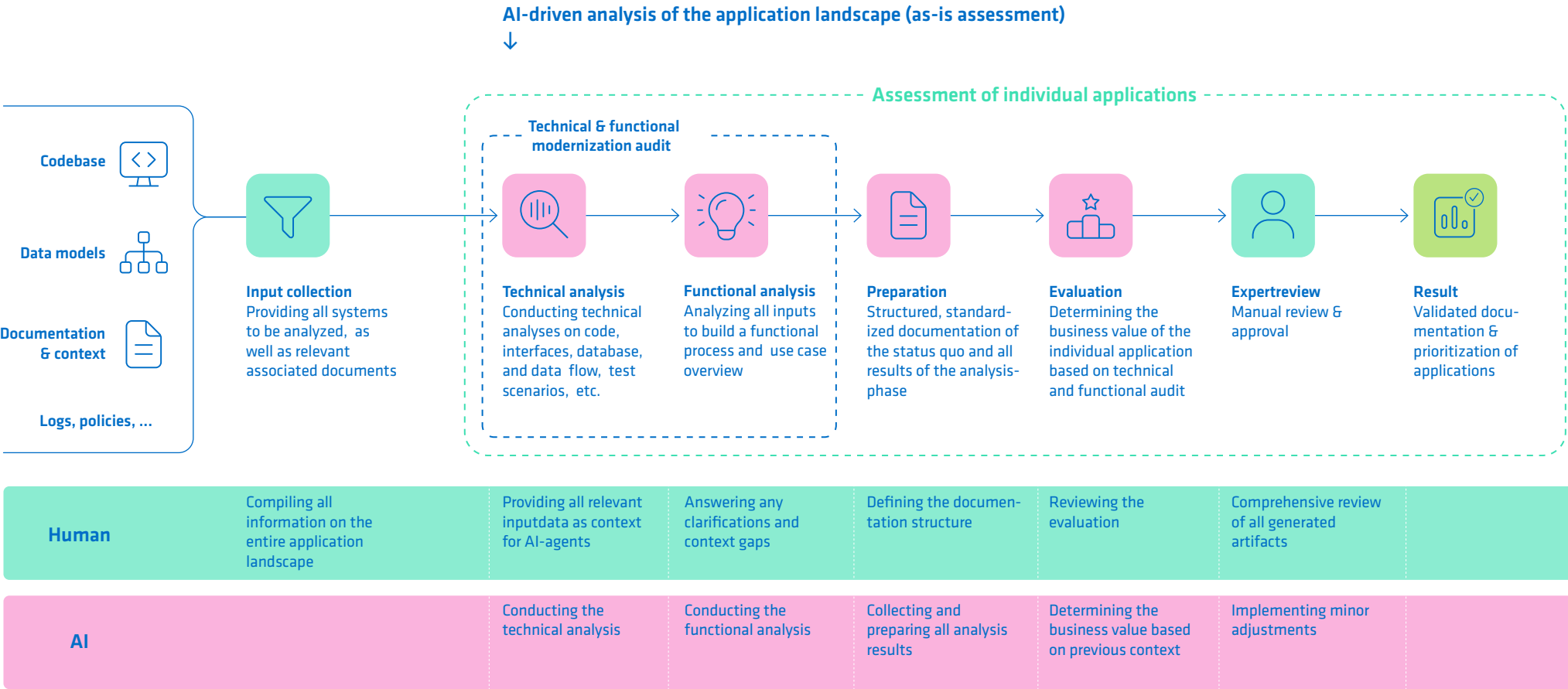
Traditionally, a comprehensive analysis of a large application landscape takes months: analysts read code, conduct interviews with knowledge carriers, document dependencies, and create assessments. The effort scales linearly with the size of the landscape – and depends on the availability of the right experts.

AI agents fundamentally change this phase. They automatically analyze codebases, architectures, data models, API interfaces, test scenarios, and logs – even across system boundaries. They extract business contexts and use them to create comprehensive technical and functional documentation in a structured, standardized format. What used to be a 'blank slate' – who understands this code? what processes are embedded in it? what must absolutely not be touched? – quickly becomes transparent through AI-supported analysis.

Human experts provide input and validate the results. They ensure that the business context is correctly captured and add knowledge that is not in the code itself: business backgrounds, regulatory constraints, and organizational dependencies.

The result is a complete modernization audit: transparency regarding dependencies, technical debt, risks, and costs – serving as a reliable basis for decision-making for management and business departments. The ROI of this analysis phase is positive in its own right.

Even if a company decides after the analysis to leave individual applications in their current state, it has gained a well-founded basis for decision-making that did not exist before. This significantly lowers the entry barrier to a modernization program – a strong argument to present to management.



[Executive Box]

The AI-supported analysis phase quickly delivers a reliable basis for decision-making – with a positive ROI in its own right. Afterward, you know where your greatest risks lie, which systems should be modernized with priority, and what it will cost. Without this transparency, modernization decisions are just gut feelings.

Step 2:
Definition of the migration path

Based on the analysis, AI agents propose the appropriate modernization path for each application or application component. They evaluate technical debt, business value, risks, and complexity, and derive a recommendation for action – along the 7 Rs:

Retain deliberately leaves stable systems in their current state – an active decision combined with regular reassessment.

Rehost moves the application to a new infrastructure environment without code changes; a pragmatic first step that solves infrastructure issues but does not address architectural weaknesses.

Replatform goes further and selectively adapts the application to leverage platform benefits such as managed services or automatic scaling.

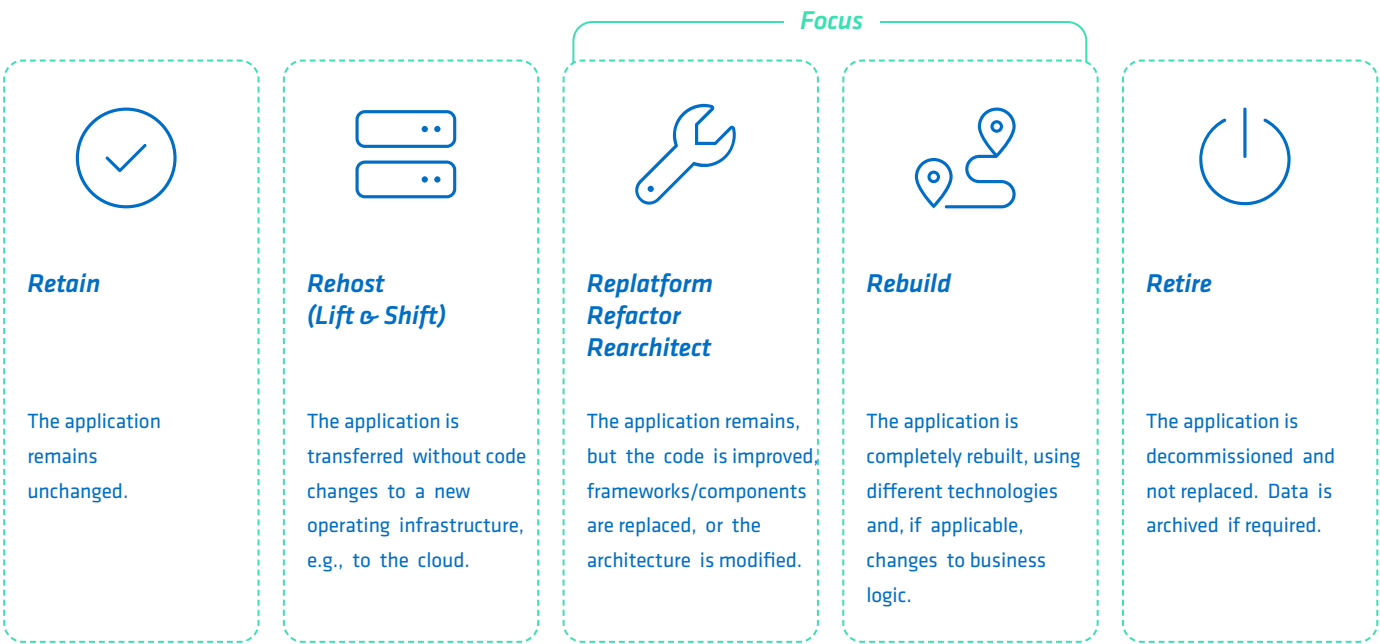
Refactor refactors the internal code – less complexity, more modularity, higher testability – without changing external behavior.

Rearchitect fundamentally transforms the architecture, for example by shifting from a monolith to a service-oriented structure.

Rebuild fundamentally changes the architecture, for example from a monolith to a service-oriented structure.

Retire finally is the deliberate, orderly decommissioning: including data archiving, knowledge transfer, and consideration of legal retention requirements – a controlled end-of-life process that reduces costs and creates clarity.

Experts make the final decision and set priorities. In practice, no single path emerges, but rather a differentiated portfolio of strategies – economically justified, not one-size-fits-all.



Phase 2:
Modernizing individual applications

Step 3:
Application-specific analysis and migration planning

Once the migration path for an application is determined, detailed planning begins. The process differs depending on the strategy – and the choice between Refactor and Rebuild is one of the most consequential decisions in the entire modernization process.

Refactor is the right path when the existing codebase is fundamentally viable but suffers from technical debt, outdated structures, or lack of testability. The business logic is preserved, and the code is modernized internally. AI agents create change sets and perform gap analyses: Where does the existing code deviate from modern standards? Which dependencies need to be resolved? Which components can be isolated? Experts evaluate the proposed changes and ensure the business context.

Rebuild focus becomes the better option when the technical foundation is so outdated or heavily entangled that any incremental improvement would cost more than starting fresh. AI agents extract the existing business processes and requirements from the legacy code. They identify business rules, data flows, and interface contracts, and transfer them into a structured specification. Experts review the extraction, add business context, and approve the specification. This is particularly crucial where implicit knowledge is embedded in the code, which cannot be correctly transferred without human interpretation.

Step 4:

Agentic Implementation

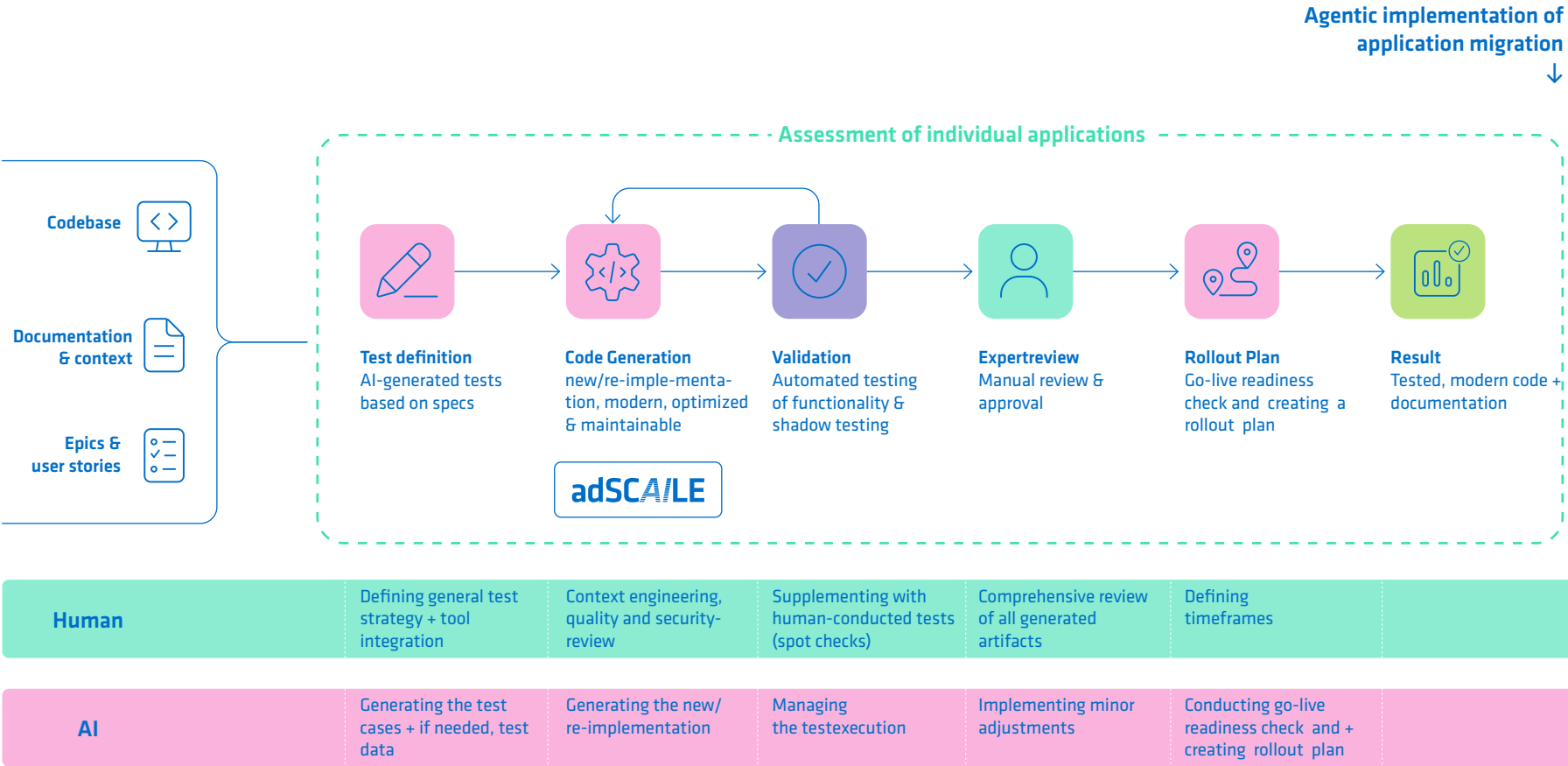
During the implementation phase, adSCALE unleashes its full potential. AI agents handle code generation and documentation, automatically generating refactoring proposals instead of requiring manual effort. During the rebuild, the extracted business logic serves as the foundation for new, modern codebases, while the system automatically generates test cases and architectural artifacts.

Experts then review and approve the results. The process remains the same as in a classic software engineering project—the only thing that changes is the effort required. Modernization projects deliver valid results in a significantly shorter time after analyzing individual applications.

Step 5:

Validation and Go-Live

AI agents perform automated testing and monitoring. Shadow testing—running the legacy and new systems in parallel—ensures that the modernized applications are functionally identical or even improved. Experts make the final go-live decision.



Human in the Loop: Not an optional feature, but a design principle

At no point in the process does AI make autonomous decisions on business-critical matters. AI agents analyze, generate, test, and document. Humans validate, prioritize, decide, take responsibility, and add implicit, company-specific knowledge that AI cannot know. This division of labor is not a compromise; it is the prerequisite for ensuring that this approach works reliably in regulated environments and for the business-critical systems of professional enterprises.

The combination of AI-powered automation and human expertise fundamentally transforms the cost structure of modernization projects. In practice, adesso achieves 30 to 60 percent lower modernization costs compared to traditional approaches. Project timelines are significantly shorter, and risks decrease because AI-driven analyses quickly establish transparency. This also makes custom software competitive again compared to expensive off-the-shelf solutions—allowing companies to continue mapping their differentiating business processes in tailored software at a cost that makes modernization economically viable.

[Executive Box]

The AI-powered analysis phase quickly delivers a reliable basis for decision-making—with an independent, positive ROI. You will know exactly where your greatest risks lie, which systems to prioritize for modernization, and what it will cost. Without this transparency, modernization decisions are just guesswork.

adesso as Partner for your Application Modernization

Modernization programs in custom development environments are complex, long-term, and business-critical. They require a partner who does not just deliver the initial analysis, but walks the entire path with you—from the initial assessment to the live operation of the modernized systems.

adesso offers a rare combination of strengths to make this happen.

Over 25 years of software engineering experience.

adesso has been developing and modernizing custom software solutions for more than two decades—across a wide variety of technology stacks and industries. We know the patterns of aging application landscapes because we built many of them ourselves or supported them for years.

More than six years of operational AI experience.

adesso has been working with AI and Large Language Models for over six years. This operational maturity makes the difference between a simple technology demo and an industrially viable modernization approach.

Industry expertise, not just technology.

With experience in key sectors—from financial services and insurance to public administration, automotive, energy, manufacturing, and retail—adesso understands the business and regulatory frameworks of your industry in detail. Modernization is always a business task as well: the business logic in legacy systems must be understood, evaluated, and correctly migrated to the new environment. This is only possible with genuine domain expertise.

Pilot projects turn into reliable references.

References turn into a modernization program that systematically future-proofs your application landscape—not as a one-off major project, but as a permanent capability for renewal.

Let's work together to identify where you can achieve the greatest leverage in your system landscape.

End-to-end, without friction.

adesso supports modernization programs from start to finish—from strategy and architectural planning to implementation and operations. There is no handoff between consulting and implementation, and no loss of knowledge between project phases.

Technology-agnostic.

adesso is not an extended sales arm of any single technology provider. Whether Azure, AWS, on-premises, or hybrid—our recommendation for a target architecture is based solely on your requirements, not on a partnership agreement.

Getting Started: Clarity in no time

The fastest path to modernization begins with an AI-powered analysis of your application landscape. In just a short time, you get a comprehensive assessment of your technical health, business value, risks, and modernization priorities. This analysis phase delivers an independent ROI and provides the decision-making foundation for your prioritized roadmap.



Michaela Gasser
CEO

marketing@adesso.ch

adesso Schweiz AG
Vulkanstrasse 106
CH-8048 Zurich
T +41 58 520 97 10
E marketing@adesso.ch
www.adesso.ch