The Treasury Research Company
INDEPENDENT TREASURY RESEARCH AND ASSESSMENT

TR/06 | TRC RESEARCH PAPER | 2026

The Adaptive TMS: Conway's Law, AI and the Future of Treasury Technology

Treasury systems have always carried something of the organisations that built them. Artificial intelligence may begin to reverse that relationship: instead of Treasury adapting to fixed software, governed software may increasingly adapt to Treasury.

Publication reference: TRC-RP-2026-006
Published 23 August 2026 · Updated 23 August 2026
Version 1.8
18 min read

TRC VIEW

The next important distinction in Treasury software may not be AI-enabled versus non-AI. It may be fixed versus adaptive. Today, artificial intelligence mainly works within functions that a vendor has already designed. The structural change begins when governed AI can help change the functions themselves. At that point the selection question starts to move from 'Does the system do what we need?' to 'Can the platform safely become what we need next?'

PLAIN ENGLISH

Essential terms

TMS - Treasury Management System	Software used to manage Treasury activities such as cash, payments, debt, investments, financial risk, accounting and reporting.
Conway's Law	The observation that a system tends to reflect the communication structure of the organisation that designed it.
Software architecture	The underlying structure of a system: how its components, data, interfaces and controls are organised and how they work together.
Orchestration	A layer that coordinates data, rules, systems and actions so that a process can run across several services rather than inside one screen or module.
Adaptive TMS - TRC term	A Treasury platform in which governed artificial intelligence can help propose, create, test and deploy changes to functionality as Treasury requirements change.
Revision authority	The degree to which artificial intelligence is permitted to change a system's own configuration, implementation or architecture rather than simply use functions that already exist.
Human-in-the-loop	A control in which a named person reviews or approves a recommendation or action before it takes effect.

RESEARCH FRAMEWORK

From fixed application to adaptive Treasury platform

01	Fixed TMS	Functions, data structures and workflows are largely designed by the vendor. Change depends on configuration options, professional services or a future product release.
02	Configurable TMS	Treasury can change parameters, workflows, reports and modules, but within boundaries that were designed in advance.
03	AI-orchestrated TMS	Artificial intelligence can query data, coordinate existing functions, call connected tools and prepare or execute governed workflows across the Treasury technology estate.
04	Adaptive TMS	Governed artificial intelligence can help translate a new business requirement into proposed functionality, controls and workflow, test the change and deploy it through an approved change process.

01

The argument in plain English

A TMS today is mostly a product the vendor has already built. Treasury chooses from the functions, workflows and configuration the vendor makes available, and often changes its own process to fit them.

The argument in this paper is that artificial intelligence may start to reverse that. Treasury could increasingly describe what it needs and the platform could propose, build and test the change within the existing control framework.

That does not mean allowing artificial intelligence to make uncontrolled changes to payments, accounting or risk. The controlled core remains. What changes is the speed and cost of turning a new Treasury requirement into usable functionality.

If that happens, the selection question changes from 'What can it do today?' to 'How safely can it become what we need tomorrow?'

02

A 1968 idea explains something important about the TMS

In 1968, Mel Conway argued that systems tend to reflect the organisations that design them. His original wording said that a system's structure becomes a copy of the organisation's communication structure. The point is simple: software is not designed in a vacuum. The way people divide work and communicate affects the way the product is divided and connected.

Later research has found strong evidence for this wider 'mirroring' effect. Product architecture and organisational architecture are related because governance, problem-solving routines and communication patterns influence the solutions that teams create.

That gives us a different way to look at Treasury technology. A Treasury Management System is not simply a list of functions. It is also the accumulated result of decisions about how a vendor organised product teams, separated responsibilities, acquired technology, connected components and released change over many years.

This paper asks what happens to that relationship when artificial intelligence starts to change the economics of software development itself.

Sources: [1] | [2]

03

The TMS is partly a record of how it was built

Traditional Treasury systems are normally presented to the buyer through modules and features: cash, payments, forecasting, debt, investments, foreign exchange, derivatives, accounting, connectivity and reporting. Those categories make sense to Treasury, but they can also reflect how software has historically been designed, owned and extended.

Over time, a mature platform accumulates design choices. Some capabilities share data naturally. Others need interfaces. Some sit on newer technology. Others remain on older foundations. Acquired products may be integrated into a wider suite. Configuration screens exist because a vendor had to predict which choices future users might need to make.

A 2024 Treasury Management International article described legacy technology stacks, monolithic architecture, slow feature releases and customisation that can be

limited or costly. The article drew on perspectives from TMI's editorial team and several Treasury technology providers and practitioners. It also contrasted cloud-hosted systems with products built natively on modern cloud services.

None of this means an older TMS is automatically poor. Mature systems often contain deep functionality, proven controls and years of specialist Treasury knowledge. The point is narrower: the client inherits some of the history and structure of the product it buys.

Sources: [1] | [2] | [3]

04

Treasury has historically paid the adaptation cost

A good TMS project should start with the business requirement. In practice, the conversation often ends with the boundaries of the product.

Treasury explains what it wants. The vendor or implementation team works out which module supports it, which fields must be configured, which workflow is closest, whether an interface is needed and whether the requirement can be met without custom development. Sometimes the answer is an elegant fit. Sometimes it is a workaround. Sometimes Treasury changes its process.

That is not necessarily bad implementation. It is a rational consequence of buying packaged software. The vendor cannot rebuild the product for every client, so the product offers a controlled set of capabilities and configuration choices. The client receives stability and repeatability, but gives up some flexibility.

The hidden cost is therefore not only licence and implementation effort. It is adaptation. Treasury spends time translating its operating model into the language and structure of a system designed before the vendor knew that Treasury existed.

Sources: [3]

05

Artificial intelligence changes the cost of changing software

Most discussion about artificial intelligence in Treasury focuses on what AI can do inside the function: improve forecasting, explain variances, identify exceptions, answer questions or automate routine work. Those uses matter, but they may not be the most important long-term change.

Two 2026 software-engineering preprints help frame the possible direction, although neither should be treated as settled evidence of a production end-state. One defines AI-native systems in terms of 'revision authority': not simply whether artificial intelligence can execute a task, but whether it can change the system's own implementation. Its proposed ladder moves from self-tuning to self-rewriting and, eventually, self-architecting systems, while requiring verification, escalation and a known fallback and leaving purpose and correctness human-owned.

A second 2026 preprint describes a software-production architecture that translates product requirements into explicit contracts, routes work through specialised AI agents, independently verifies the output and continuously adapts the software over time. Its early deployment evidence is limited, but the relevant idea for this paper is that software delivery could become an ongoing operating capability rather than a sequence of large development projects.

A peer-reviewed paper involving researchers at the University of Cambridge explores the broader idea that end users could describe what software needs to do and generative AI could take that requirement much further through the software

lifecycle, potentially from conception to deployment. Requirements Are All You Need does not discuss Treasury or Treasury Management Systems and presents this as a research agenda rather than a finished capability. But it supports an important part of the argument in this paper: the distance between a business requirement and working software may become materially shorter.

For Treasury, that raises a different question. If the requirement can increasingly drive the software directly, what happens to a TMS model in which the vendor has historically defined the functionality in advance?

Sources: [8] | [9] | [13]

06

The orchestration layer is already arriving

TRC has not identified a current Treasury product that demonstrates the full Adaptive TMS definition used in this paper. But several parts of the architecture are already visible in public product and market evidence.

Citi's Treasury 2030 GPS report describes a possible future architecture in which Treasury services sit beneath a data-transmission layer and an orchestration layer. The orchestration layer uses policy, deterministic rules and artificial intelligence to guide workflow, recommendations, approvals and actions. In that model, incumbent TMS and Enterprise Resource Planning systems can remain underneath the new user experience rather than acting as the only interface.

FinanceKey announced in May 2026 that it had launched a Model Context Protocol server which, according to the company, allows enterprise AI tools to access live Treasury data through existing application programming interfaces, data structures and permission models. The relevance to this paper is the architectural separation of the AI interaction layer from the underlying Treasury data and systems; TRC has not independently tested the product capability.

Kyriba's own product material says users can query Treasury data in plain language, run repeatable skills and orchestrate actions while retaining human approval, permissions and audit trails. Again, this is vendor-described capability rather than independent TRC product testing, but it provides evidence of the direction in which established TMS providers are moving.

On the public evidence reviewed, these examples primarily demonstrate orchestration of existing capabilities rather than the full Adaptive TMS model defined in this paper. They do, however, weaken the assumption that the Treasury user's workflow has to be defined by the menu structure of one application.

Sources: [4] | [5] | [6] | [7]

07

The next step is more important: from using functions to changing them

Bond is the closest current Treasury-market example of a deliberate break with the traditional feature-led TMS model found in this review. In a March 2026 treasuryXL article, Bond founder Marcus Strych argued for re-designing the TMS from the ground up around AI, with an IDE-style interface in which simple commands can manipulate and control Treasury activity. He argued that the future should be a re-engineering of software rather than another layer of AI features. Bond's own product material also describes AI agents operating across cash, payments, risk, liquidity, trading and reporting on a common intelligence layer.

That is materially closer to TRC's argument than a conventional TMS with an AI assistant added on top. But the distinction still matters. Bond's published material focuses on AI-native interaction, automation, context, agents and progressive learning. TRC's Adaptive TMS argument asks a further question: can a Treasury requirement begin to create or reshape governed production functionality itself, rather than only invoke, combine or automate capabilities the platform already contains?

There is a large difference between an AI agent using an existing payment function and artificial intelligence helping create a capability that did not previously exist. TRC's argument sits in that gap.

At the orchestration stage, the vendor still defines the available building blocks. AI chooses, sequences or explains them. At the adaptive stage, the business requirement can start to reshape the building blocks themselves.

That could mean generating a new workflow, extending a data model, creating a report, configuring a control, building an interface, adding a calculation or assembling existing services in a way that was not pre-designed as a product option. The proposed change would then be inspected, tested, approved, versioned and deployed through a controlled process.

This is where the relationship begins to reverse. Historically, the vendor's organisation helped shape the software and the client then adapted to it. In an adaptive model, the client's requirement has a much more direct route into the software's structure.

Artificial intelligence does not abolish Conway's Law. It potentially gives the Treasury team more influence over which organisation the software is mirroring.

Sources: [1] | [8] | [9] | [17] | [18]

A simple example: Treasury starts hedging commodities

Assume a company has never hedged commodities. The business changes and Treasury is asked to introduce commodity risk management.

Today the team would normally assess whether its existing system supports the instruments, exposure data, market data, valuations, limits, approvals, accounting and reporting. It may need another module, specialist software, configuration, interfaces, consultants and a formal change project.

Now consider a future controlled interaction. Treasury tells the platform: 'We are introducing commodity hedging. These are the exposures. These are the permitted instruments and counterparties. These are our limits, approval authorities, market-data sources and accounting requirements. Design the required workflow and controls, identify the data we are missing and show me the proposed change before anything is enabled.'

The system could produce a proposed operating workflow, data structure, control points, interfaces, reports and test cases. People would still decide the policy, approve the design and validate the result. But much of the translation between requirement and technology could happen in near real time rather than through a conventional development queue.

That is the Adaptive TMS in practical terms. Not uncontrolled self-modifying software. A much shorter, governed path from a new Treasury requirement to working functionality.

Adaptive must not mean uncontrolled

Treasury is not a sensible place to confuse flexibility with freedom from control. A system that can change more quickly needs stronger boundaries around what may change, who may request it and how the result is verified.

Some parts of the architecture should become more stable, not less. Identity and access management, segregation of duties, trusted data, policy, bank connectivity, accounting integrity, audit evidence and the authoritative record of transactions form the controlled core. AI-generated or AI-configured functionality should inherit those controls rather than recreate them each time.

The AI-native systems research reaches a similar conclusion. Revision authority requires a verification process, an escalation mechanism and a verified fallback. Kyriba's agentic-finance guidance makes the Treasury version more concrete: permission scope, logged tool calls, human approval, audit trails and reversibility are design requirements, not optional features.

The National Institute of Standards and Technology also treats governance, measurement and risk management as part of the AI lifecycle. The more authority a system receives, the more important it becomes to know what changed, why it changed, what evidence supported the change and how it can be stopped or rolled back.

A 2026 conference paper by Bharath Yadla, affiliated with the University of Oxford and Workato, argues that enterprise-scale agentic AI needs explicit maturity models, calibrated human oversight and integration with existing systems and processes. This is the author's practitioner framework, not an institutional Oxford position. An Economist Impact whitepaper, sponsored by Oracle and KPMG, similarly reports that enterprise agents need a well-governed digital estate, observability, auditability, accountability and a unified integration layer. The Cambridge Centre for Alternative Finance at Cambridge Judge Business School adds a useful warning from its 2026 financial-services study: software engineering was the most mature AI application among the financial-industry respondents it surveyed, while the volume and velocity of AI-generated code can make traditional manual review increasingly ineffective. Governance therefore has to be designed for the speed of the technology, not bolted on afterwards.

The future TMS therefore does not become less controlled. The better version becomes easier to change and easier to prove.

- Separate the trusted Treasury core from the functionality that may adapt.
- Require every generated change to be inspectable, testable and version-controlled.
- Apply existing permissions and segregation of duties to AI-created workflows.
- Keep human approval proportional to the financial consequence.
- Maintain a tested rollback or fallback for material changes.

Sources: [7] | [8] | [10] | [14] | [15] | [16]

10

Requirements become more important, not less

It would be easy to conclude that adaptive software makes requirements obsolete. TRC takes the opposite view.

Requirements Are All You Need, the peer-reviewed paper involving University of Cambridge researchers, makes the point from the software side: natural requirements may become the primary way end users direct software creation. TRC's Treasury

conclusion follows from that. If the requirement becomes executable more quickly, defining the requirement properly becomes a control in its own right.

If the cost of translating an instruction into software falls sharply, the quality of the instruction becomes more important. Artificial intelligence can build the wrong thing faster as easily as it can build the right thing faster.

Treasury still has to decide its operating model, risk appetite, decision rights, controls, permitted instruments, data ownership, accounting treatment and service expectations. Technology should implement those choices. It should not invent them.

In the traditional model, an unclear requirement creates implementation delay and change requests. In an adaptive model, an unclear requirement could create working software that faithfully implements the wrong assumption. The discipline moves upstream.

The future therefore strengthens the sequence: understand the business problem, define the scope, document the operating requirement and controls, then allow the technology to adapt around them.

Sources: [8] | [9] | [13]

11

Treasury may stop looking so similar

Treasury functions are different, but they are also recognisably similar. Some of that is unavoidable. Regulation, accounting standards, banking infrastructure, market conventions and shared ideas of good practice pull Treasury teams towards common patterns.

Technology has reinforced that convergence. Packaged systems are built to serve many clients, so they offer a common set of modules, workflows, data structures and connectivity choices. Research on packaged enterprise systems more broadly has described a standardising effect, where software can embed common procedures across organisations. Treasury commentary has made a similar practical point: Treasury technology is often standard and generic because it has to satisfy the widest possible market, while customisation can be limited or costly.

That suggests a second, looser mirroring effect. Conway's Law describes how the organisation building a system can shape the software. The packaged TMS can then shape the organisations buying it. Different Treasury teams end up running loose variations of the same operating model because they are choosing from much of the same technology and connectivity.

An Adaptive TMS could weaken that constraint. If a Treasury team can describe a requirement and reliably turn it into controlled functionality at speed, the technology can be shaped much more closely around the organisation it serves.

One Treasury might build liquidity decisions around volatile commodity exposure. Another might design cash controls around an acquisition model. Another might build forecasting around operational data that is unique to its industry. The technology no longer needs to push all of them towards the same answer.

That creates the possibility of something more important than software flexibility: Treasury itself could become more differentiated. Research outside Treasury has linked flexible IT infrastructure with organisational responsiveness and competitive advantage. The Treasury implication is not that bespoke is automatically better. It is that Treasury could build capabilities that are unusually well matched to the organisation where that difference genuinely creates value.

But the opposite is equally true. If a good requirement can become working functionality faster, a bad requirement can too. Weak processes can be automated. Poor assumptions can become workflows. Unnecessary complexity can be created

simply because it is easy to create.

Adaptive technology therefore raises both the opportunity and the consequence. Good Treasury design could create more organisation-specific value. Bad Treasury design could erode value faster.

That is another reason requirements, operating-model design and governance become more important, not less.

Sources: [3] | [19] | [20] | [21]

12

TMS selection may need a new dimension: adaptability

Treasury technology selection has traditionally concentrated on current functional fit. That remains necessary. A buyer cannot rely on a promise that missing functionality might be generated later.

But current functionality becomes a less complete measure if software can change much faster. Buyers may need to assess two things separately: what the product can do today and how safely it can evolve tomorrow.

That changes due diligence. Architecture, data access, application programming interfaces, permission inheritance, testing, version control, reversibility and the ability to connect external services become part of functional resilience, not simply technical detail.

A platform with 95% of today's feature list but a closed and difficult-to-change architecture may carry a different long-term risk from a platform with 85% of today's list but strong evidence that controlled new capability can be added quickly.

TRC therefore proposes adaptability as a separate evaluation dimension rather than hiding it inside a generic technology score.

- Can users express a new requirement in business terms and receive an inspectable proposed design?
- What can be changed without vendor professional services or a product release?
- Can new workflows and integrations be built in an isolated test environment?
- Do generated changes automatically inherit permissions, policy and segregation of duties?
- Is every change logged, versioned, explainable and reversible?
- Can the platform use open interfaces and external services without weakening control?
- How does the vendor prove that AI-generated functionality is correct before production use?
- What remains impossible to change without replatforming?

Sources: [3] | [4] | [5] | [6] | [7]

13

The economics of the TMS could change as well

A large feature catalogue has historically represented years of product investment and specialist development. It has therefore been a meaningful competitive advantage.

If the cost of creating some functionality falls, the value may move down the stack. The durable assets become the trusted data model, connectivity, control framework, accounting integrity, workflow engine, integration layer and the evidence that generated change can be governed safely.

Implementation economics could change too. Some work currently performed through workshops, configuration teams, custom development and change requests may be compressed. That does not remove the need for Treasury expertise. It shifts expert effort away from translating every requirement into a product-specific configuration and towards defining the requirement, validating the design and governing the outcome.

The vendor relationship may consequently become less about buying a fixed bundle of functions and more about buying a controlled environment in which Treasury capability can evolve.

That is a significant business-model question for the market. It is also one reason why today's largest feature catalogue does not automatically identify tomorrow's strongest platform.

Sources: [3] | [4] | [8] | [9]

14

There are good reasons this may develop more slowly in Treasury

The direction is plausible; the timing is uncertain. Treasury software carries consequences that a generated marketing website does not. Payments move money. Hedging creates market exposure. Accounting errors affect reported results. Connectivity failures can stop a business operating.

Legacy architecture also matters. Artificial intelligence cannot make a closed data model open or remove years of technical debt by instruction alone. A platform needs suitable interfaces, test environments, permissions and change controls before adaptive functionality is safe or useful.

There is also a determinism problem. Many Treasury calculations and controls require the same input to produce the same result. A language model may help interpret the requirement and orchestrate the work, but calculations and controls that require repeatable, deterministic output should continue to use deterministic engines and controlled data.

Finally, vendors have economic choices. A technically possible model may challenge revenue from modules, professional services and long implementation programmes. Technology does not determine the commercial model on its own.

For those reasons, TRC does not predict the rapid disappearance of the TMS. The more realistic near-term path is a controlled core with an increasingly intelligent and adaptable layer around it.

Sources: [3] | [7] | [8] | [10]

15

Where this argument sits against existing thinking

The individual building blocks of this argument are not new. Conway's Law is nearly 60 years old. Research has tested the relationship between organisations and product architecture. Treasury commentators have questioned legacy TMS architecture and the cost of customisation. Citi GPS has described an orchestration layer above existing Treasury services. Treasury vendors are describing AI agents, natural-language interfaces and open AI connectivity. Recent software-engineering

preprints are also exploring systems with greater authority to revise their own implementation.

The closest general research we found is Requirements Are All You Need, a peer-reviewed paper involving researchers at the University of Cambridge. It asks whether end users could own the software development lifecycle from conception to deployment using natural requirements. TRC therefore does not claim to have originated the broader idea that AI could turn business requirements more directly into software. That idea already exists in the software-engineering literature.

The closest Treasury-specific thinking we found is Bond. In a March 2026 treasuryXL article, Marcus Strych argued that AI should not simply be bolted onto the TMS, but that the TMS should be re-designed from the ground up with an IDE-style interface and a move away from a feature-led model. That is a substantive challenge to traditional TMS design and clearly precedes this paper. TRC therefore does not claim to have originated the idea of an AI-native or fundamentally re-engineered TMS.

FinanceKey's own material is also directionally close to this argument. In May 2026 it described 'programmable treasury', where Treasury teams move from users of software towards builders of their own operating environment, and said dashboards, views, reports and workflows could be configured by finance teams without IT or consultant involvement so that the platform reflects how the organisation actually operates. In July 2026 FinanceKey separately published its view of what makes a Treasury platform adaptable. The distinction remains important: FinanceKey's published position centres on configuration, open infrastructure, application programming interfaces and tools built on top of the platform. TRC's Adaptive TMS argument asks whether governed artificial intelligence can go further by translating a new Treasury requirement into new or reshaped production functionality itself.

The distinction is narrower. Bond's published proposition centres on AI-native interaction, automation, intelligent context, agents and progressive learning across Treasury. TRC's argument connects the issue to Conway's Law and the adaptation cost of packaged software, then asks whether AI-assisted software creation can shorten the path from a new Treasury requirement to new governed functionality. From that, TRC proposes adaptability itself as a separate TMS selection and due-diligence dimension.

A 2026 conference paper by Bharath Yadla, affiliated with the University of Oxford and Workato, strengthens the governance side of the argument through a practitioner framework for agentic enterprise oversight. An Economist Impact whitepaper, sponsored by Oracle and KPMG, also addresses the architecture and governance needed to scale agents. The Cambridge Centre for Alternative Finance at Cambridge Judge Business School provides broader financial-services evidence that software engineering is already a mature AI use case and that faster AI-generated code creates new assurance challenges.

Conway's Law has also been applied directly to modern capital-markets technology. A-Team Group used it in March 2026 to discuss composable trading platforms and the organisational change required to support them. In July 2026, a software-engineering essay examined what happens to Conway's Law when AI agents write code, arguing that AI can amplify existing organisational structure rather than remove it.

TRC's contribution is therefore deliberately narrower and Treasury-specific: the historical TMS can be understood partly as a product of the organisation that built it; packaged software has required Treasury to absorb part of the adaptation cost; and the same packaged technology can create a second, looser mirror as Treasury functions converge around common functionality and connectivity. Generative and agentic AI may reduce those constraints by shortening the path from Treasury requirement to governed functionality. That could allow Treasury functions to become more organisation-specific, while making the quality of requirements even more important. TRC therefore proposes adaptability as a separate dimension in TMS selection and due diligence.

In the sources reviewed through 23 August 2026, TRC did not identify a published Treasury paper making that combined argument. This is a scoped research finding, not a claim that no person has ever expressed a similar thought.

Sources: [1] | [2] | [3] | [4] | [5] | [6] | [8] | [11] | [12] | [13] | [14] | [15] | [16] | [17] | [18] | [19] | [20] | [21] | [22] | [23]

16

The question changes

The TMS is unlikely to disappear. Treasury will continue to need trusted data, connectivity, accounting integrity, security, permissions, controls and an authoritative record of financial activity. Those foundations become more valuable when artificial intelligence is given greater authority.

What may disappear gradually is the assumption that the functionality sitting above that core must remain largely fixed until the vendor chooses to change it.

If that happens, the best Treasury platform will not necessarily be the one that contains the most functions on the day of selection. It may be the one that can most safely, quickly and transparently become what Treasury needs next.

For decades the practical question has been: 'Does the system fit our Treasury?'

The next question may be more important: 'How safely can the system adapt when our Treasury changes?'

PUT THE RESEARCH TO WORK

Define what the technology must adapt to

More adaptable software does not remove the need for scope and requirements. It makes them more important. Use the TRC Scope & Requirements Engine to document the Treasury outcome, data, controls and operating requirements before assessing the technology.

Define your scope and requirements

METHODOLOGY AND RESEARCH CONTROL

Research question	Could artificial intelligence change the Treasury Management System from a largely fixed or pre-configured application into a governed platform that can adapt more directly to Treasury requirements?
Research type	Independent synthesis of software-architecture research, treasury technology research, current product evidence and TRC analysis.
Evidence treatment	Current capability and future inference are kept separate. Vendor material is used to show what is available or being developed today, not as proof that the proposed end-state has been achieved.
Existing research and related thinking	We reviewed existing Treasury articles, academic research, vendor material and wider technology thinking to see what had already been said or built around this idea. A peer-reviewed paper involving researchers at the University of Cambridge has already explored the broader idea of end users directing software creation from requirements through deployment using generative AI. Bond is the closest Treasury-specific example we found of a deliberate AI-native break with the traditional feature-led TMS. FinanceKey has separately described programmable Treasury, finance-team configuration without IT or consultants, and an adaptable Treasury platform. We also reviewed research on the standardising effect of packaged enterprise systems and the link between flexible IT infrastructure, organisational responsiveness and competitive advantage. TRC therefore does not claim that requirements-to-software, AI-native TMS design, programmable Treasury or agentic enterprise architecture are new ideas. The argument here is narrower: applying Conway's Law to the historical TMS, identifying the adaptation cost absorbed by Treasury, recognising the looser mirroring effect packaged technology can have across Treasury functions, and asking whether governed AI can shorten the path from a Treasury requirement to new or reshaped production functionality. From that, TRC proposes adaptability as a separate TMS selection dimension.
Independence	The analysis and conclusions are TRC's own. TRC accepts no vendor funding for its research or assessments.
Control	Prepared by TRC Research. Editorial review completed. Evidence cut-off 23 August 2026. Controlled version 1.8.

REFERENCES

Sources and further reading

1. Conway's Law / How Do Committees Invent?
Melvin E. Conway | April 1968; source page accessed 23 August 2026
https://www.melconway.com/Home/Conways_Law.html

2. Exploring the duality between product and organizational architectures: A test of the 'mirroring' hypothesis
Research Policy / Harvard Business School | 2012
https://www.hbs.edu/ris/Publication%20Files/Research%20Policy%2041%20%282012%29%201309%E2%80%93%201324_c5c2350e-013c-4065-a2f9-d95eb32177d5.pdf

3. Are the Days of the TMS Finally Numbered?
Treasury Management International | 28 August 2024
https://treasury-management.com/articles/are-the-days-of-the-tms-finally-numbered

4. Treasury 2030: Modernize or Risk Irrelevance
Citi Institute | 3 December 2024
https://www.citigroup.com/global/insights/treasury-2030

5. FinanceKey launches native MCP server bringing real-time treasury data to enterprise AI
FinanceKey | 27 May 2026
https://financekey.com/press/financekey-launches-native-mcp-server/

6. TAI: Kyriba's trusted approach to Agentic Finance
Kyriba | Accessed 23 August 2026
https://www.kyriba.com/products/agentic-ai-tai/

7. Agentic finance: a no-hype guide for treasury teams
Kyriba | Accessed 23 August 2026
https://www.kyriba.com/resources/insights/agentic-finance-guide/

8. Defining AI-Native Systems: Autonomy as Revision Authority
arXiv | 22 July 2026
https://arxiv.org/abs/2607.21659

9. Meta-Engineering Harnesses for AI-Native Software Production
arXiv | 25 May 2026
https://arxiv.org/abs/2605.25665

10. Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile
US National Institute of Standards and Technology | 26 July 2024; updated 8 April 2026
https://www.nist.gov/publications/artificial-intelligence-risk-management-framework-generative-artificial-intelligence

11. Breaking Conway's Law: Why Composable Trading Platforms Demand Organisational Change, Not Just Better APIs
A-Team Insight | 16 March 2026
https://a-teaminsight.com/blog/breaking-conways-law-why-composable-trading-platforms-demand-organisational-change-not-just-better-apis/

12. The Photocopier: What happens to Conway's Law when AI writes the code
modularthinkin | 20 July 2026
https://www.modularthinkin.com/blog/whitepaper-conways-law-ai/

13. Requirements Are All You Need: The Final Frontier for End-User Software Engineering
ACM Transactions on Software Engineering and Methodology; University of Cambridge repository | 2025
https://www.repository.cam.ac.uk/items/039a5ddf-4157-4b6e-8b57-623f4892b450

14. Becoming an agentic enterprise: a practitioner methodology and frameworks for human-AI governance
Bharath Yadla; Oxford University Research Archive | 16 July 2026
https://ora.ox.ac.uk/objects/uuid%3A92120dbd-b0ec-4ce2-a61d-31b107326d1d

15. Agents of change: Rise of the autonomous AI enterprise
Economist Impact; sponsored by Oracle and KPMG | Accessed 23 August 2026
https://impact.economist.com/technology-innovation/agents-of-change

16. 2026 Global AI in Financial Services Report: Adoption, Impact and Risks
Cambridge Centre for Alternative Finance, Cambridge Judge Business School | 28 April 2026
https://www.jbs.cam.ac.uk/faculty-research/centres/alternative-finance/publications/2026-global-ai-in-financial-services-report/

17. How AI is Revolutionizing Treasury Operations
treasuryXL / Marcus Strych, Bond | 26 March 2026
https://treasuryxl.com/blog/how-ai-is-revolutionizing-treasury-operations/

18. Bond: Agentic Money Management
Bond | Accessed 23 August 2026
https://www.bondtreasury.com/

19. Sticking to standards; technical and other isomorphic pressures in deploying ERP-systems
Information & Management | March 2006
https://www.sciencedirect.com/science/article/pii/S0378720605000480

20. The Paradox of Sophistication
Treasury Management International / François Masquelier | 17 May 2017
https://treasury-management.com/articles/the-paradox-of-sophistication

21. Building and leveraging information in dynamic environments: The role of IT infrastructure flexibility as enabler of organizational responsiveness and competitive advantage
Information & Management | December 2010
https://www.sciencedirect.com/science/article/pii/S0378720610000674

22. Treasury orchestration: how real-time data and AI are reshaping corporate finance
FinanceKey / Veikko Koski | 8 May 2026
https://financekey.com/insights/treasury-orchestration-real-time-data-and-ai/

23. What makes an adaptable treasury platform?
FinanceKey | 29 July 2026
https://financekey.com/insights/adaptable-treasury-platform/

Sources accessed 23 August 2026. External statistics retain their original definitions and populations.