


Applied AI for Engineering and Delivery Modernization in Healthcare

How Brillio helps healthcare enterprises reimagine documentation, testing, migration, monitoring, and developer productivity with practical AI



A hand holding a glowing digital particle, symbolizing AI and technology. The background is a dark blue, futuristic digital environment with glowing lines and particles.

Software delivery teams are under growing pressure to move faster without compromising quality, traceability, or operational resilience. Across the SDLC, organizations are looking for practical ways to reduce manual effort, standardize workflows, improve testing and documentation, and modernize legacy environments without adding friction for already stretched teams.

This collection of case studies highlights how Brillio is applying GenAI, intelligent automation, and AI-assisted engineering to solve those challenges in focused, high-impact ways. From design-to-story generation and release documentation to multilingual QA, end-to-end test automation, migration acceleration, and developer productivity, these examples show how AI can create measurable value across engineering and delivery workflows.



Standardizing SDLC documentation using GenAI agents

With inconsistent SDLC documentation across teams, where varying story formats, incomplete acceptance criteria, and uneven documentation quality were creating rework and delivery delays, this leading healthcare organization turned to Brillio for help. Developers, BAs, and testers were spending significant time manually writing user stories, test cases, and specifications, while release note creation required manual collation across JIRA and GitLab. The lack of standardization was affecting alignment, slowing releases, and weakening audit readiness.

We've addressed this by deploying a suite of GenAI-powered agents to streamline documentation and development workflows across the SDLC. The solution included an AI Story Generator that converted epics or plain-English prompts into structured user stories, a Smart Acceptance Criteria agent that generated clear BDD-style testable conditions, an AI Code Reviewer that summarized pull requests and suggested improvements based on coding standards, and an Auto-Generate Release Notes capability that transformed JIRA tickets and GitLab commits into concise, readable summaries. A Tech Spec Generator also helped create first drafts of Confluence design and technical documents from epics or relevant code.

Because the solution was integrated directly into JIRA, GitLab, and Confluence using GenAI and existing APIs, it improved documentation quality without adding platform overhead. This created a more standardized, scalable documentation model that reduced manual effort, improved consistency, and accelerated delivery workflows across teams.

Business Impact

~50% time savings
for PMs and POs in preparing specs

65–75% reduction
reduction in manual effort

3x faster
documentation turnaround

40–50% improvement
in code review speed and quality with
GenAI-powered suggestions

80%+ consistency
in acceptance criteria



Automating design-to-knowledge capture using AI

A global healthcare enterprise was struggling to preserve design knowledge in a consistent, traceable way across delivery workflows. Design documents were often not linked to JIRA epics, creating inconsistency and oversight during sprint planning and QA. At the same time, siloed design docs, missing knowledge base artifacts, and rushed documentation were reducing visibility, slowing onboarding, and weakening support readiness after deployment.

Brillio addressed this by automating the flow from design artifacts to structured knowledge capture. The solution used template-driven automation to link Confluence design documents to their corresponding JIRA epics using predefined templates, improving traceability while reducing manual mapping effort. It also introduced a PR-to-KB pipeline that automatically generated Confluence draft knowledge base articles for customer-impacting merge requests using LLM-based summarization of PR descriptions and code diffs.

Confluence pages followed a standardized KB structure with placeholders, which improved consistency in tone, formatting, and coverage across teams.

This created a more reliable bridge between design, development, and support documentation. By reducing manual linking, improving documentation visibility, and accelerating KB creation, the solution strengthened onboarding, release communication, and post-deployment support readiness.

Business Impact

Accelerated developer

onboarding with instant access to design context

Reduced time spent

tracking missing documentation

Lower developer

overhead through LLM-generated KB drafts

Automated doc-to-epic

linkage visibility and reduced manual effort

Improved traceability,

transparency, and release communication

Faster support

team readiness post-deployment



Automating SDLC release documentation using GenAI agents

A leading American managed care organization was spending too much time manually drafting release documentation from JIRA ticket data and engineering updates. Release note creation was labor-intensive, error-prone, and inconsistent, often requiring teams to translate technical changes into business-friendly language by hand. Delays in documentation were affecting internal coordination, customer readiness, and overall release velocity, while developers were losing focus through constant context switching.

To help the client overcome these hurdles, we've built a GenAI-enabled release documentation workflow that automatically pulled merged ticket information and transformed it into clear, readable release notes. The solution generated contextual summaries and Gherkin-style stories to highlight business outcomes while reducing manual translation effort. It also supported seamless Confluence integration, allowing real-time publishing of drafts through direct integration with Confluence so release documentation stayed aligned with delivery timelines. Built for CI/CD-native deployment, the solution fit into existing delivery pipelines with minimal disruption and supported scaling across teams.

This shifted release note generation from a repetitive manual task to a more standardized and automated process. The result was faster publication, better readability for both business and technical audiences, and lower effort across release cycles.

Business Impact

50% faster time

to publish release notes

70–80% reduction

in effort for drafting and formatting release notes

20–30% gain

in developer productivity by eliminating non-core manual tasks

Standardized release

documentation, reducing rework and follow-ups

Improved readability

and structure for business and technical stakeholders



Re-shaping loyalty & rewards with AI-driven test automation

A major healthcare company operating loyalty and rewards workflows was struggling with slow, fragmented, and compliance-sensitive test data provisioning. Manual test data preparation was delaying loyalty payout testing, increasing the risk of defects in claim logic, and extending QA and UAT cycles.

Use of production-like data without sufficient anonymization also created compliance concerns around PHI and PII, while limited reuse across environments reduced scalability and test coverage.

To address this, we have developed an AI-augmented synthetic test data framework built on Talend. The solution simulated real-world claims behavior using configurable edge cases and rule-based anonymization to support HIPAA-compliant test environments through dynamic PHI/PII masking. When enhanced with GenAI, the framework could also generate edge conditions and defect-driven test assertions based on historical defect data and existing test patterns.

Modular decision logic embedded in Talend's transformation and filtering layers helped prioritize service types, shape response timelines, and simulate claims lifecycle variations. The framework continuously adapted reusable templates and scenarios based on test outcomes to improve regression coverage over time.

This created a more scalable and audit-friendly QA model that reduced manual effort while improving realism and reusability in test data preparation. By embedding compliance into the transformation layer and making the pipeline parameterized and reusable, the solution improved both QA velocity and data quality.

Business Impact

80% reduction

in manual parsing of claim data for testing purposes

5x increase

in QA team productivity by automating data provisioning

60% reduction

in handoffs between Dev, QA, and Infra teams

Improved CMS audit

readiness through audit-friendly test data workflows such as logs and masks

Lower production

defects due to high-fidelity test coverage

Faster time-to-release

from shorter QA and UAT cycles

Increased loyalty campaign

throughput and reduced compliance risk



Automating multilingual QA workflows using design-to-test AI agents

With growing complexity in multilingual QA for conversational AI flows across multiple languages and platforms, a major American healthcare player came to Brillio for help. Manual testing was time-consuming, inconsistent, and difficult to standardize, while translating Figma-based designs into executable test cases required significant human effort. Existing approaches also created traceability gaps that made it harder to meet audit expectations tied to HIPAA, PCI-DSS, and GDPR, and traditional tooling struggled to keep up with fast-paced enterprise deployments.

We've built a design-to-test automation framework that extracted structured test logic directly from Figma designs and automated prompt seeder and test case generation across platforms. Using Amazon Bedrock foundation models, the solution semantically interpreted and validated multilingual contact flows with minimal human input. A modular AWS-native architecture hosted containerized microservices on ECS to coordinate parsing, test generation, validation, and reporting, with integration through API Gateway and CI/CD pipelines. To support compliance and auditability, test assets and audit logs were stored in S3, made searchable via OpenSearch, and tracked through DynamoDB metadata.

The result was faster test generation, stronger compliance readiness, and better standardization of cross-platform QA assets.

Business Impact

5x faster test

case generation across Cyara, Hammer, and Selenium

70% reduction

in manual multilingual QA effort

Improved compliance

posture through structured audit logs and traceable test assets

Higher test accuracy

via AI-driven extraction and flow mapping

Accelerated release cycles,

reducing testing bottlenecks and enabling faster go-to-market

Cross-platform

standardization of test formats and structures



Standardizing design review using VisuaLint AI and Figma APIs

Struggling with slow, manual, and error-prone design QA as teams compared Figma files by hand to validate updates, a major healthcare enterprise partnered with Brillio. Traditional pixel-based diff tools surfaced visual changes without understanding their functional significance, while the lack of structured documentation created misalignment across design, product, and development teams. As design systems grew, these manual review processes became harder to scale and increased regression risk.

Brillio addressed this by building VisuaLint AI, a semantic comparison engine powered by OpenAI models via AWS Bedrock and integrated directly with the Figma API. Instead of only showing visual diffs, the solution semantically interpreted changes between Figma templates and highlighted meaningful layout, structural, and content differences that affected user experience. Design files were fetched directly from Figma, parsed into structured JSON, and compared using smart diffing logic to detect inconsistencies in hierarchy, spacing, alignment, and component usage. A modular Python microservices architecture handled normalization, diff generation, and report creation, while the output included human-readable changelogs and annotations to support QA, reviews, and sprint demos.

This made design review more intelligent, scalable, and useful for downstream teams. By prioritizing meaningful changes over cosmetic noise, the solution accelerated feedback loops, improved design-to-development accuracy, and reduced interpretation gaps that often lead to rework.

Business Impact

60–70% reduction
in manual design comparison time

Faster design QA cycles,
improving time-to-feedback and release alignment

Improved
design-to-dev accuracy, reducing
front-end rework

Clear, human-readable
changelogs that enhanced collaboration
across teams

Reduced regression
risk by catching layout and content issues early

Scalable solution
that grows with design systems and product teams



Translating Figma designs to user stories using GenAI agents

Spending significant time translating Figma designs into structured user stories for sprint planning and backlog grooming, with manual story creation often introducing errors, misinterpretation of design intent, and rework during development, this major healthcare product organization turned to Brillio for help. As teams and product complexity grew, maintaining consistent documentation across fast-moving agile environments became increasingly labor-intensive and difficult to scale.

We've helped the client by building a design-to-story automation engine that interpreted Figma layouts, components, and text to generate structured, actionable user stories mapped to design intent. The solution embedded design context directly within the generated stories so that key functional and UX details were preserved and communicated clearly to developers. Built on a scalable Python-based microservice architecture, the system integrated with design tools and agile delivery platforms, with optional JIRA connectivity to push stories into backlog workflows. The output was formatted for easy use by developers and QA teams, helping accelerate review and grooming cycles.

This created a more scalable and consistent way to convert design artifacts into delivery-ready stories. By reducing manual interpretation and preserving design intent more effectively, the solution improved planning speed, cross-functional alignment, and delivery quality.

Business Impact

60–70% reduction

in manual effort for story creation

Improved consistency

and accuracy in user stories, reducing development rework

Faster sprint planning,

enabling quicker release cycles

Better

cross-functional alignment, preserving design intent in development workflows

Scales across

teams and projects, supporting agile delivery at scale



Streamlining SRE L1/L2 Support with Grafana Monitoring

A major enterprise was using Grafana effectively for visualization, but still relying on manual effort to turn dashboards, logs, and alerts into meaningful operational action. Grafana did not understand the context behind what it displayed, so users had to manually configure thresholds, interpret anomalies, and assemble alerting logic across disjointed systems. Dashboard curation, panel naming, summaries, and root-cause investigation remained repetitive and time-consuming, especially as monitoring environments evolved.

We've enabled the client through an LLM plugin-based approach within Grafana to introduce more contextual, AI-assisted monitoring workflows. The solution supported anomaly detection and contextual root cause analysis by correlating logs, metrics, and traces, helping teams move faster from observation to action.

We have also defined KPIs around response-time reduction, crashloop pod automation rates, customer satisfaction, and cost savings, while identifying high-impact use cases such as AI-powered stylists, chat assistants, smart inventory forecasting, and generative marketing content. A modular microservices structure allowed discrete components for chat, analytics, and recommendations to scale independently, while governance included monitoring for data shifts, response quality, usage trends, and model drift, alongside alerting and rollback protocols.

This turned Grafana from a largely visualization-led experience into a more intelligent operational support layer for SRE teams. By combining AI-driven anomaly detection, RCA, and remediation recommendations, the solution helped reduce manual troubleshooting effort and improve response speed.

Business Impact

Smarter anomaly

detection through correlation of logs, metrics, and traces with AI-driven context

Contextual root

cause analysis

Faster incident

response and lower MTTR

AI-driven assistants

that surface critical errors, recommend fixes, and summarize incidents

Automated

remediation and recommendations

Reduced downtime

through suggested and in some cases automated service restarts or configuration tweaks



AI-driven end-to-end testing automation for 64% faster testing cycles

A major healthcare organization operating complex PBM workflows was struggling with fragmented, manual end-to-end testing that created inefficiencies, missed defects, and high coordination overhead. Limited intelligent test automation was increasing testing time, delaying releases, and driving up the cost of delivering critical PBM services. At the same time, weak visibility into QA progress and a shortage of PBM-specific QA resources were making it harder to maintain coverage across high-risk workflows.

We helped the client by building an AI-driven test automation workflow that allowed QA and business users to input test scenarios as outcomes and instructions, then automatically map them to existing functional tests and documentation across PBM applications. The solution used AI agents to generate, execute, and monitor end-to-end tests, while capturing real-time evidence such as videos and screenshots and producing detailed validation reports for faster issue tracking. A self-driving test execution layer enabled the system to autonomously write, run, and monitor test cases, improving execution speed and validation quality. To keep the model practical, Brillio used a cost-conscious hybrid deployment that blended AI agents with rule-based automation and integrated into existing QA ecosystems.

This shifted end-to-end testing from a fragmented manual exercise to a more intelligent and scalable QA model. By improving coverage, execution speed, and reporting quality, the solution helped reduce risk in complex PBM workflows while strengthening release readiness.

Business Impact

Increased scenario

coverage from 35 to 80% across 30+ specialty drugs workflow

70% drop

in critical post-release defects, especially in high-risk flows like cold-chain Rx and same-day delivery

Reduced

end-to-end regression testing cycles from 7 days to 2.5 days per release



Revolutionizing concierge services with AI for a 5-star experience

A consumer-facing enterprise was struggling to simplify everyday coordination across schedules, vehicle needs, shopping lists, weather, and appointments. Users were dealing with fragmented notifications and stand-alone apps that lacked a unified, context-aware experience, while many existing voice assistants could not handle context switching or multi-step tasks effectively. The result was cognitive overload, low continuity across tasks, and limited value from otherwise useful consumer tools.

We've addressed this by building JAI ("Just AI"), a personalized concierge application designed to unify voice, calendar, weather, and task management into a single contextual assistant. Powered by a contextual conversational engine built on IBM Watson, ChatGPT, and Unreal Speech, the solution allowed users to interact naturally while retaining awareness of tasks, preferences, and flows from one topic to the next. It also introduced multimodal context awareness across vehicle integration, calendar and scheduling, and weather and environmental context, so the assistant could deliver reminders, routing updates, alerts, and task support based on real-world conditions. Additional capabilities for task and list management, plus proactive suggestions such as oil-change reminders or meeting prep prompts, made the experience more anticipatory and useful over time.

This transformed the experience from a set of disconnected utilities into a more intelligent and context-aware concierge service. By unifying multiple signals into one assistant, the solution improved stickiness, user engagement, and the day-to-day usefulness of AI-driven consumer support.

Business Impact

Simplified everyday

tasks with a unified AI assistant, leading to strong early adoption

~60K beta users

onboarded pre-launch

82% rated

the experience 4 or 5 stars

Increased daily active

usage through unified voice, calendar, weather, and list management

Context retention

boosted retention to a 45-day DAU/MAU ratio of 37%



Autonomous testing & PRC using Playwright and OpenAI

Spending significant effort documenting and validating portal customizations made for each customer implementation, this healthcare technology provider came to Brillio for a solution. Changes across HCP portals, consent product flows, and field portals had to be captured for legal review through annotated documentation, while development PODs were spending roughly 25–30% of their effort on testing and documentation of these product changes per implementation. This created both delivery drag and ongoing maintenance overhead.

To solve this, we have used OpenAI and Playwright to automate test generation, validation, and PRC documentation. The solution analyzed application behavior and user interactions to generate Playwright test scripts, then introduced self-healing tests that could detect and update broken selectors dynamically as the UI changed. It also used AI-powered visual regression testing to identify meaningful UI changes while reducing false positives during comparison. For documentation, the solution generated Playwright screenshots as PDFs with the required annotations, packaging them as PRC materials for review.

This reduced the manual effort tied to both validation and documentation while improving the resilience of the testing workflow as the UI evolved. It also demonstrated how AI can support faster implementation cycles by combining automated testing, self-healing scripts, and review-ready output generation in a single model.

Business Impact

80%

reduction in manual effort on PRC packets and validation

50%

reduction in TAT

1.25x

productivity gain

Self-healing

test cases



Enhanced coding productivity through Cursor AI + OpenAI/Anthropic integrations

A leading healthcare enterprise was facing development slowdowns caused by a complex legacy codebase, incomplete documentation, and the need for front-end reskilling in modern UI frameworks. Enhancing the existing eCOA Questionnaire Builder required significant manual effort, while gaps in documentation limited context availability for development and impact analysis. Traditional coding approaches were also constraining team velocity and slowing project timelines.

To help the client, we've used Cursor AI as an advanced AI-powered code editor to streamline software development through autocomplete, code generation, and intelligent rewrites. The solution gave developers an interactive chat-based development interface that could query the codebase, reference specific code blocks, and integrate web search results directly into the workflow. Cursor also supported complex multi-file code generation, making it useful for large, interdependent development tasks, while its interface was configured to support integrations with OpenAI, Anthropic, and custom models for project-specific needs. This helped developers work with greater context and less manual effort across both modernization and ongoing enhancement activities.

The result was a more efficient and flexible engineering workflow that improved productivity without requiring a full reset of the development environment. It also showed how context-aware AI tooling can accelerate UI development and reduce manual coding effort when paired with strong prompt discipline and human oversight.

Business Impact

Reduced manual

coding effort significantly by generating 70% UI code

~40%

sprint velocity increase

Improved

development efficiency



Intelligent Test Automation and Validation for Power BI Dashboards

Lacking a standard automated testing approach for Power BI dashboards forced teams to rely on time-consuming manual validation across reports and widgets. For this healthcare enterprise, test case creation was inconsistent and labor-intensive, complex scenarios required significant additional refinement, and data validation often depended on direct database querying to verify what was shown in visualizations. As dashboard usage grew, these inefficiencies were limiting testing speed and time to market.

To help the client, we have built an AI-assisted test automation and validation framework for Power BI. Using Cursor AI, the solution applied predefined templates to structure test scenario inputs and generate test scripts and test cases tailored to specific Power BI report scenarios. It also integrated direct PostgreSQL-based validation to compare dashboard widgets against underlying data for greater accuracy and less manual checking. To fit within existing delivery workflows, the framework produced output in a format compatible with JIRA import, maintaining consistency across reports and making the approach easier to scale.

This created a more repeatable and efficient model for dashboard QA. By combining AI-generated test assets with direct data validation and reusable automation components, the solution reduced testing effort, improved consistency, and helped accelerate dashboard delivery.

Business Impact

25% reduction

in test case preparation effort with
~80% accuracy

50% improvement

in testing efficiency resulting in
~\$55k saving

Faster time

to market by 2 months



Achieving 75% efficiency gains through hybrid-AI in Struts to Spring Migration

A major enterprise was looking for a more efficient way to migrate large volumes of legacy Java files from Struts to Spring JSP architecture. File-level migration was too slow, JSP rule conversion added complexity, and existing approaches were delivering only about 25% efficiency in development workflows. The organization needed a faster and more scalable migration strategy that could reduce cost while improving execution quality across a large application suite.

To tackle this, we've used a hybrid AI approach that combined Cursor AI, a Struts-to-Spring JSP Migrator Tool, OpenRewrite, and rule-based execution frameworks. The Cursor AI workflow introduced standardized prompts, automatic controller grouping, module-based batch processing, and rule-based execution to improve workflow orchestration and developer efficiency. In parallel, the Struts-to-Spring JSP migration tool used intelligent context-aware tag processing, flexible custom tag-to-field mapping, enhanced error detection, and batch-mode execution to improve transformation speed and quality. Brillio then combined these tools strategically based on migration requirements to maximize overall development efficiency and lower delivery cost.

This created a more practical modernization model than relying on a single migration tool alone. By blending AI-assisted workflow management with specialized modernization utilities, the solution improved migration efficiency, reduced projected cost, and made large-scale Java modernization more viable over a multi-year horizon.

Business Impact

Estimated cost

reduction from ~\$5 million to ~\$2 million for migrating a suite of applications over a 2 year time period

~45% efficiency

improvement with Cursor AI workflow

~70% efficiency

gains with Struts to Spring JSP migration tool

ABOUT BRILLIO

Brillio is a digital technology services company that drives AI-first engineering and design-led experiences for global enterprises. Born digital in 2014, its consulting-led services span Customer Experience, Data & AI, Product Engineering, and Digital Infrastructure. With an industry-leading NPS of 71, Brillio accelerates time to market through its proprietary BrillioOne.ai platform, powered by AI-ready talent with deep domain expertise.

Brillio is the official Digital Transformation Partner and the official Data and AI Services Provider of Atlassian Williams Racing. Brillio partners with leading technology providers including Microsoft, AWS, Google Cloud, Salesforce, Adobe, Databricks, and Snowflake and operates with 6,000+ “Brillians” across 15 global delivery centers. Consistently recognized as a Great Place to Work® since 2021, Brillio blends innovation, talent, and purpose to deliver measurable outcomes for clients and fulfilling careers for employees.



<https://www.brillio.com/>

Contact Us: info@brillio.com

brillio