Post-Generative Development

A practical framework for assessing, correcting and improving AI-assisted digital products

By John Mahlow | Cursive Media | Version 1.3 | November 29, 2026 | cursivemedia.com/post-generative-development/white-paper

Executive summary

Generative AI can accelerate development without guaranteeing that the resulting application is complete, maintainable, secure, efficient or appropriate for its intended audience. A functioning application can still contain unnecessary complexity, poor design, weak discoverability or operational risks. An application can also be well built and simply need further improvement.

Post-Generative Development (PGD) is the discipline of evaluating, correcting, restructuring, optimizing and maintaining existing digital products created or materially modified through generative AI. Its objective is fitness for intended use, not the removal of AI-generated work as such. The framework applies to websites, web applications and other digital products, whether they are prototypes, live systems or incomplete builds.

Framework at a glance

Layer	Role
Foundations	Defines Post-Generative Development (PGD), Generative Debt (GD), scope and principles.
Pillars	Nine domains of work: the part of the product being assessed or improved.
Classification	Distinguishes debt, defects, risks and improvement opportunities.
Lifecycle	Assessment, intervention, validation and optional ongoing maintenance.
Methods	Corrective and improvement techniques applied within a pillar.
Standards	Evidence-based criteria for judging quality and acceptance.
Services	Five primary service families, with specialized services qualified by pillar or subdomain.

PGD is not a claim that every AI-assisted product is defective. It is also not a replacement for conventional development, security, accessibility or SEO practice. It organizes those disciplines around an existing AI-assisted product and its quality, completeness and fitness for use.

Product foundations

Each pillar owns a distinct quality outcome. Activities can span several pillars, but every finding receives one primary classification.

01 Post-Generative Code

The structure, correctness and maintainability of the application implementation.

Scope: Architecture; source code quality; dependency hygiene; business rules; implementation completeness; code-level documentation.

Boundary: Owns implementation and logic. The Experience pillar owns user-facing workflow quality; the Data pillar owns persistence and information integrity; the Security pillar owns exploitability and controls.

Example services: Post-Generative Code Audit; Post-Generative Code Rescue; Post-Generative Code Optimization; Post-Generative Code Maintenance.

Evidence of improvement: Reproducible logic tests; reduced duplication or unused dependencies; documented architecture; confirmed implementation of intended behavior.

02 Post-Generative Design

The visual and interface quality of a digital product, including the coherence of its design system.

Scope: UI design; typography; layout; visual hierarchy; responsiveness; component consistency; visual assets and design systems.

Boundary: Owns presentation, not the user journey. The Experience pillar owns whether people can complete tasks, including functional accessibility. Content owns the meaning and accuracy of written material.

Example services: Post-Generative UI Audit; Post-Generative UI Rescue; Post-Generative UI Optimization.

Evidence of improvement: Consistent components and visual states; verified layouts across specified viewports; removal of redundant visual elements; comparison to agreed design requirements.

03 Post-Generative Experience

The usability, accessibility and effectiveness of the interactions through which people accomplish intended tasks.

Scope: UX; navigation; interaction flows; forms; user-facing functional completeness; accessibility; conversion paths and friction.

Boundary: Owns task outcomes and interaction behavior. If a task fails because of incorrect implementation, classify the root cause under the Code or Data pillar and tag Experience as an impact.

Example services: Post-Generative UX Audit; Post-Generative Experience Optimization; Post-Generative Accessibility Audit; Post-Generative Accessibility Rescue; Post-Generative Conversion Optimization.

Evidence of improvement: Task completion tests, accessibility checks, clear interaction states and measured or observed reductions in friction.

Visibility, information and protection

These domains address whether the product can be discovered, understood and used without unacceptable security or privacy exposure.

04 Post-Generative Search

The discoverability and search-readiness of a digital product across relevant search and answer systems.

Scope: Technical SEO; crawlability; indexability; canonicalization; on-page metadata; semantic structure; structured data; AI search accessibility. Off-page diagnosis is included where it informs remediation.

Boundary: Owns discovery mechanisms and search-specific signals. The Content pillar owns substantive accuracy and editorial value; link acquisition and ordinary ongoing SEO campaigns are not inherently PGD.

Example services: Post-Generative SEO Audit; Post-Generative Technical SEO Rescue; Post-Generative SEO Optimization.

Evidence of improvement: Verified crawl/index directives; correct metadata and structured data; accessible rendered content; no promises of search ranking or AI citations.

05 Post-Generative Content

The factual, editorial and communicative quality of content in an existing AI-assisted digital product.

Scope: Accuracy; relevance; completeness; duplication; tone; readability; product claims; documentation and user-facing copy; content and media consistency.

Boundary: Owns meaning, claims and editorial quality. The Design pillar owns presentation; the Search pillar owns discovery; the Experience pillar owns the impact of content on task completion.

Example services: Post-Generative Content Audit; Post-Generative Content Rescue; Post-Generative Content Optimization; Post-Generative Copy Optimization; Post-Generative Content Maintenance.

Evidence of improvement: Verified factual claims; removal of placeholder or duplicate copy; coherent tone; alignment with the product and intended audience.

06 Post-Generative Security

The protection of applications, accounts, systems and information against unauthorized access, misuse and exposure.

Scope: Authentication; authorization; secrets; input/output controls; dependency vulnerabilities; abuse protection; privacy and secure configuration.

Boundary: Owns protection requirements regardless of where a vulnerability occurs. The Reliability pillar owns operational failure and recovery; the Infrastructure pillar owns runtime environments and deployment.

Example services: Post-Generative Security Audit; Post-Generative Security Hardening; Post-Generative Security Maintenance.

Evidence of improvement: Reproduced and remediated findings; verified permission boundaries; appropriate secret handling; security retests against scoped requirements.

Operational foundations

Reliability, data and infrastructure are independent outcomes. They often share technical components, but their acceptance criteria differ.

07 Post-Generative Reliability

The ability of an application to behave consistently, perform acceptably and recover from operational failure.

Scope: Testing strategy and coverage; performance; error handling; resilience; availability; concurrency behavior; failure recovery and verification.

Boundary: Owns how dependably the system behaves under intended conditions. The Code pillar owns implementation structure; the Security pillar owns protection; the Infrastructure pillar owns the operational environment.

Example services: Post-Generative Reliability Audit; Post-Generative Reliability Rescue; Post-Generative Performance Optimization; Post-Generative Reliability Hardening.

Evidence of improvement: Repeatable regression tests; measured latency or resource use; verified error paths and recovery under defined conditions.

08 Post-Generative Data

The correctness, integrity and movement of information within and between systems.

Scope: Data models; schemas; migrations; validation; consistency; persistence; API contracts; integrations; synchronization; analytics and event instrumentation integrity.

Boundary: Owns information semantics, accuracy and exchange. The Infrastructure pillar owns database hosting and service availability; the Security pillar owns access and confidentiality; the Code pillar owns internal implementation.

Example services: Post-Generative Data Audit; Post-Generative Data Rescue; Post-Generative Integration Rescue; Post-Generative Data Optimization.

Evidence of improvement: Reconciled records; validated constraints; verified API contracts and end-to-end data flows; correct instrumentation and migration outcomes.

09 Post-Generative Infrastructure

The environments, configuration and operational systems required to build, deploy and run the application.

Scope: Hosting; build pipelines; deployment; environment configuration; DNS; runtime resources; operational monitoring; backup infrastructure; scaling and rollback mechanisms.

Boundary: Owns the environment and delivery mechanism. The Data pillar owns information integrity; the Reliability pillar owns application behavior under failure; the Security pillar owns protection controls.

Example services: Post-Generative Infrastructure Audit; Post-Generative Deployment Rescue; Post-Generative Infrastructure Optimization; Post-Generative Infrastructure Hardening; Post-Generative Infrastructure Maintenance.

Evidence of improvement: Reproducible builds and deploys; verified environment configuration; working monitoring and rollback; demonstrated recovery mechanisms.

Classify the finding, then choose the work

Generative Debt is an accumulated liability introduced or materially increased through AI-assisted development that creates avoidable future effort, risk or ongoing cost. It may exist in any pillar, even when the product works. A finding may have multiple characteristics; its AI origin must not be assumed from appearance alone.

Finding type	Working distinction
Debt	A present shortcut, structural deficiency or omission with a continuing cost or liability.
Defect	A confirmed failure to meet intended behavior or a defined requirement.
Risk	A plausible adverse outcome or exposure, whether or not failure has occurred.
Improvement opportunity	A beneficial change not required to correct an existing deficiency.

Record a primary type and relevant secondary tags. Track AI attribution separately: confirmed (direct evidence), suspected (supporting but inconclusive evidence), unknown, or not attributable to AI. Confirmed and suspected Generative Debt must be distinguished; an observed issue does not establish AI causation.

PGD lifecycle

Stage	Output
Assessment / Audit	Scope, baseline, findings, evidence and prioritized recommendations, including whether to repair or rebuild.
Intervention	Agreed corrective or improvement work; change record and implementation evidence.
Validation	Check the agreed deliverables and outcomes against acceptance criteria; record evidence, regression checks where relevant and unresolved limitations. Required before closing every engagement.
Maintenance	Optional ongoing engagement: monitoring, recurring review, upkeep and Generative Debt management.

The lifecycle lists the stages available, not a sequence every engagement must follow. An engagement can be an assessment only. Intervention is whatever work was agreed. Validation closes every engagement, scaled to its scope. Maintenance is optional.

Cross-pillar corrective methods

Method	Distinct purpose
Sanitization	Remove unnecessary, redundant or unsupported elements.
Repair	Correct defects or complete omitted, already-intended functionality.
Refactoring	Improve internal structure while preserving intended observable behavior.

Methods may be combined within Rescue, Optimization, Hardening or Maintenance engagements. Rescue is not a mandatory lifecycle stage. Validation is a required closing check for every engagement, not a separately sold service. Audit engagements validate their findings and report; intervention engagements verify the agreed changes. Optimization and Hardening are defined as service families in Section 05.

Five primary service families

The pillars classify the work. The service families classify the reason a customer engages a practitioner. Neither replaces the other.

Service family	What customers say, and the defined outcome
Post-Generative Audit (PGA)	"I like what I built. Did I miss anything?" "It could be so many things." "I just want an expert opinion." Assess an existing AI-assisted product, across one or more applicable pillars, and deliver evidence, findings, priorities and recommendations, including whether to repair the product or rebuild it.
Post-Generative Rescue (PGR)	"I keep getting this error." "Everything works great except…" "I like how it looks, but I can't figure out how to make it work." Restore broken, unusable or substantially incomplete intended functionality to an agreed working state.
Post-Generative Optimization (PGO)	"It works. It's just slow." "Can you change a few things?" "Help me get it where I want it." Improve measurable efficiency, quality or effectiveness; remove excess or restructure existing implementations as appropriate.
Post-Generative Hardening (PGH)	"Could we get hacked?" "I don't know if it's secure." "Can you get it live?" Reduce exposure to security threats and operational failure, and verify scoped protection and resilience requirements.
Post-Generative Maintenance (PGM)	"Can you keep it running?" "Can you host it and manage it?" "Who makes edits when I need them?" Preserve product quality through recurring upkeep, monitoring, updates, testing and Generative Debt management.

The quotes are how people actually describe their situation, not scripts. Classify an engagement by its objective, not by the pillar it touches: an Audit determines the product's condition; a Rescue corrects an existing failure; an Optimization improves existing quality; a Hardening reduces vulnerability and fragility; Maintenance preserves quality over time.

Naming convention

Post-Generative + [Domain or specialization] + [Service family]

Use a family name by itself for broader engagements (for example, Post-Generative Audit). A qualified name identifies the focus, such as Post-Generative Code Audit, Post-Generative UI Optimization or Post-Generative SEO Optimization. A pillar may also be referenced independently, such as Post-Generative Design.

Spell out Post-Generative and Generative in all formal names. The defined framework acronyms are PGD, GD, PGA, PGR, PGO, PGH and PGM. Introduce the full term on first use. Pillars, methods and specialized services do not receive additional framework acronyms. Standard technical abbreviations such as UI, UX, SEO and API remain valid.

Generative Debt is a classification of liability, not a service-name prefix. A Post-Generative Audit can investigate whether it exists; evidence determines attribution. Generative Debt management is part of Post-Generative Maintenance.

Specialized services and team usage

A specialized service combines a domain or subdomain with a primary service family. The domain describes what is being worked on; the family describes the outcome the customer needs. Use the full service name, and map the work to its applicable pillars.

Customer need	Specialized service	Primary pillar
Working app with excess code	Post-Generative Code Optimization	Code
Generic or inconsistent interface	Post-Generative UI Optimization	Design
Confusing checkout flow	Post-Generative UX Optimization	Experience
Accessibility barriers prevent use	Post-Generative Accessibility Rescue	Experience
Pages cannot be indexed due to a defect	Post-Generative Technical SEO Rescue	Search
Inaccurate claims in the copy	Post-Generative Content Rescue	Content
Repetitive or generic copy	Post-Generative Content Optimization	Content
Exposed access credentials	Post-Generative Security Hardening	Security
Slow application	Post-Generative Performance Optimization	Reliability
Crashes or fails under normal use	Post-Generative Reliability Rescue	Reliability
Records fail to sync	Post-Generative Integration Rescue	Data
Build will not deploy	Post-Generative Deployment Rescue	Infrastructure

A Post-Generative Audit may be comprehensive or limited to agreed pillars; a pillar audit is a specialized assessment. Specify the scope rather than assuming every audit covers all nine. Not every pillar-family combination is a distinct offering.

Communication rules by team

Sales: Start with the customer's symptom and desired outcome. Identify the relevant service family and pillar(s). Propose a scoped engagement without requiring customers to diagnose their own technical problems or promising a repair before scope is known.

Marketing: Use full framework names consistently, paired with plain-English descriptions. Distinguish a comprehensive Post-Generative Audit from a single-domain assessment. Describe the actual scope and outcomes offered.

Content creation: Use the glossary as the vocabulary reference. Explain terms on first use, avoid treating AI use as evidence of poor quality, and distinguish framework definitions from established industry standards.

Delivery: Assign the primary pillar by the quality outcome or root cause, tag secondary pillars, record evidence, and validate the agreed outcome before closing each engagement.

Suggested customer-facing explanation

Post-Generative Development helps businesses assess, fix and improve websites and applications built with AI. Whether the product already works or needs substantial repair, the focus is making it more reliable, usable, secure and maintainable for its intended purpose.

Consistent assessment criteria

Every PGD assessment should define product scope, intended use, applicable pillars, current evidence and acceptance criteria. A prototype and a production payment application do not require the same depth of review. A finding must describe an observed condition rather than an assumed cause.

Assessment element	Required information
Scope	Product, environment, version, intended use and exclusions.
Finding	Primary pillar, type, observed condition and supporting evidence.
Attribution	Evidence connecting the issue to AI-assisted development, if available.
Impact	Severity, likelihood where relevant, business impact and priority.
Action	Recommended method or service family, owner or workstream, and acceptance criterion.
Verification	Pass, fail, not applicable or not verified; evidence and remaining limitations.

Example: Post-Generative Data finding: duplicate customer records following repeated synchronization. Reproduce the issue, determine the cause, repair the integration and verify that repeated events no longer create duplicate records. Classify AI origin only if supporting evidence exists.

This framework establishes assessment principles, not an independently validated conformance or certification standard. Specific testable requirements and thresholds can be developed per pillar without changing the taxonomy. A critical unresolved finding must not be concealed by an overall score.

Glossary: foundational terms

Term	Canonical meaning
Generative Development	Creation or material modification of digital products using generative AI.
Post-Generative Development (PGD)	Quality-focused assessment, correction, improvement and maintenance of existing AI-assisted digital products.
Post-Generative	Work within the discipline; does not imply that AI is prohibited during the work.
Generative Debt (GD)	Accumulated liabilities attributable to AI-assisted development that create avoidable future effort, risk or cost.
Confirmed Generative Debt	Debt with direct evidence of AI-assisted origin or material increase.
Suspected Generative Debt	Debt with supporting but inconclusive evidence of AI-assisted origin or material increase.
Pillar	A domain of specialization defined by its quality outcome.
Method	A technique used to correct or improve an identified condition.
Lifecycle	The stages available to an engagement: assessment, intervention, validation and maintenance. Not every engagement uses every stage.
Debt	A present shortcut, structural deficiency or omission with a continuing cost or liability.
Defect	A confirmed failure to meet intended behavior or a defined requirement.
Risk	A plausible adverse outcome or exposure, whether or not failure has occurred.
Improvement opportunity	A beneficial change not required to correct an existing deficiency.
AI attribution	The evidence-based classification of whether a finding originated in AI-assisted development: confirmed, suspected, unknown or not attributable to AI.
Acceptance criteria	Defined conditions that must be met to verify an agreed outcome.

Methods, service terms and distinctions

Term	Canonical meaning
Service family	A category of engagement defined by the primary outcome sought.
Pillar audit	An audit focused on one defined pillar or its specialization.
Specialized service	A service family qualified by a domain or specialization, mapped to one or more pillars.
Post-Generative Audit (PGA)	Evidence-based assessment of an existing AI-assisted product across agreed pillars, ending in prioritized recommendations, including whether to repair or rebuild.
Post-Generative Rescue (PGR)	Restoration of broken, unusable or substantially incomplete intended functionality.
Post-Generative Optimization (PGO)	Improvement in measurable quality, efficiency or effectiveness.
Post-Generative Hardening (PGH)	Reduction of security exposure or operational fragility.
Post-Generative Maintenance (PGM)	Ongoing upkeep and quality assurance, including Generative Debt management.
Sanitization	Removal of unnecessary, redundant or unsupported elements; not synonymous with input sanitization.
Repair	Correction of defects or completion of omitted, already-intended functionality.
Refactoring	Internal restructuring that preserves intended observable behavior.
Validation	Required verification of agreed deliverables and outcomes against acceptance criteria.
Generative Debt management	Ongoing identification, prioritization, reduction and prevention of Generative Debt within Maintenance.
Assessment standard	Defined criteria and verification procedures for assessing product quality.

Everyday terms and their framework equivalents

People rarely search for framework terms. These are the words they use, mapped to the framework for marketing, sales and content. Use them to name a phenomenon or a search, never to describe a client's own product.

What people say	Framework term
Vibe coding	Building or changing a product by prompting an AI tool: Generative Development.
A vibe-coded app or site	An AI-assisted product, the subject of Post-Generative Development.
AI slop (in code)	Generative Debt.
Deslop, deslopification	Sanitization.
Vibe fixing, vibe code rescue, AI app rescue, software project rescue	Post-Generative Rescue.
Fix a vibe-coded website, site or app	Post-Generative Rescue, or a Post-Generative Audit first when the cause is unknown.
Vibe code audit, AI code audit	Post-Generative Audit.
Vibe coding to production	Post-Generative Hardening.

What people say	Framework term
Vibe code cleanup specialist	A practitioner of Post-Generative Development.
The vibe coding hangover	The moment Generative Debt comes due.

Relationship to existing work

The concept of Generative Debt has prior published use. Tyler Slater's 2025 preprint distinguishes structural deficiencies, unnecessary generated complexity and omitted implementations. Mujahid and Imran's 2026 preprint studies explicitly acknowledged AI-related technical debt, including deferred tests and incomplete adaptation. These works support the narrower code-level problem; the nine-pillar PGD framework extends the proposed scope to the broader digital product. This framework does not claim invention of the terminology, independent validation or industry-wide adoption of PGD.

Slater, T. (2025). Quantitative Analysis of Technical Debt and Pattern Violation in Large Language Model Architectures. arXiv:2512.04273. https://arxiv.org/abs/2512.04273

Mujahid, A. A., & Imran, M. M. (2026). "TODO: Fix the Mess Gemini Created": Towards Understanding GenAI-Induced Self-Admitted Technical Debt. arXiv:2601.07786. https://arxiv.org/abs/2601.07786