AGENI Framework

Agile Generative AI Framework

Una metodología para equipos que construyen productos con IA Generativa — donde los outputs son probabilísticos, el modelo es una dependencia externa y "terminado" requiere una nueva definición.

Desarrollado por Daniel Salazar
Program Manager & AI Practitioner

Versión 1.0 · 2026 · agenIframework.com · CC BY-SA 4.0

RESUMEN

Las metodologías de desarrollo tradicionales — Agile, Scrum, PMBOK — fueron diseñadas para software determinista. Los sistemas de IA Generativa son fundamentalmente diferentes: sus outputs son probabilísticos, su comportamiento cambia con cada actualización del modelo, y su "código" incluye la ingeniería de prompts como un artefacto de primera clase. AGENI (Agile Generative AI Framework) es una metodología práctica que extiende los principios Agile específicamente para la era GenAI. Introduce cinco fases iterativas, dos loops de retroalimentación, gates de decisión y niveles de rigor calibrados al riesgo del proyecto.

> *Nota de madurez: AGENI v1.0 es un framework conceptual derivado de práctica directa en proyectos de automatización IA e IA generativa. Aún no ha sido formalmente validado con métricas de adopción a escala. Se ofrece como punto de partida estructurado para pilotos y medición — no como práctica probada. Las secciones sobre limitaciones, cuándo no usar AGENI y afirmaciones defendibles reflejan esto con honestidad.*

1. Introducción

Agile ha tenido una buena trayectoria. Dos décadas de sprints, retrospectivas e historias de usuario han hecho genuinamente más ágiles y receptivos a los equipos de software. Para la mayoría del software, eso se sostiene: escribes la función, la pruebas, la publicas y el comportamiento es predecible. El input entra, el mismo output sale.

La IA Generativa rompe ese supuesto.

Cuando la funcionalidad central de tu producto está impulsada por un modelo de lenguaje grande, no estás escribiendo lógica determinista. Estás orquestando comportamiento probabilístico. Un prompt que funciona perfectamente hoy puede producir resultados diferentes tras una actualización del modelo. Una evaluación que pasa en desarrollo puede fallar en producción porque la distribución de inputs reales de los usuarios no se parece en nada a tus casos de prueba. Lo que da forma a todo ese comportamiento — el system prompt, los ejemplos few-shot, la estrategia de recuperación — está escrito en lenguaje natural, que ningún sistema de control de versiones, pipeline de integración continua (CI) o framework de QA fue diseñado para manejar.

Los equipos que acoplan Scrum a un proyecto de IA Generativa sin modificaciones tienden a chocar con los mismos muros: sprint reviews donde nadie puede acordar si el output fue realmente bueno, criterios de "terminado" que se derrumban en el momento en que alguien pregunta cómo lo sabes, e incidentes en producción que se remontan a una actualización del modelo del proveedor que ocurrió hace tres semanas y que nadie notó.

AGENI surgió de construir estos productos, no de teorizar sobre ellos. Un fallo muy común es encontrar prompts sin dueño — nadie puede modificarlos con confianza porque nadie sabe por qué funcionan. Ese patrón es el origen del concepto de Prompt Debt. El momento más desconcertante es una actualización silenciosa del modelo del proveedor que cambia el tono del producto sin que cambie una sola línea de código — y sin que nadie tenga un proceso para manejarlo. Ese vacío es lo que aborda el loop externo de retroalimentación. AGENI conserva las partes de Agile que funcionan y reemplaza las que asumen que tu código hará lo mismo dos veces.

AGENI no es un framework académico. Es la respuesta de un practicante a un problema real: ¿cómo publicas productos fiables cuando tu tecnología central es intrínsecamente probabilística?

2. Por Qué las Metodologías Tradicionales Se Quedan Cortas

2.1 El Supuesto del Determinismo

El supuesto central de toda metodología de software importante es tan obvio que casi nunca se enuncia: si tu código es correcto, se comporta de la misma manera cada vez. Una compilación exitosa es una promesa. Ese supuesto está integrado en todo — tu pipeline de integración continua (CI), tu definición de terminado, tus criterios de aceptación, tu sprint review. Nadie lo cuestiona porque, para el software tradicional, siempre es verdad.

Cuando colocas un modelo de lenguaje en el centro de tu producto, cuatro cosas cambian que ninguna metodología tradicional contempla:

– Outputs probabilísticos: el mismo input puede producir diferentes outputs entre ejecuciones.

– Dependencia externa del modelo: el comportamiento central vive en un modelo de terceros que se actualiza independientemente de tu ciclo de lanzamiento.

– Lenguaje natural como lógica: los prompts son instrucciones, no código — son más difíciles de probar, versionar y razonar.

– Modos de fallo emergentes: el sistema puede fallar de formas contextuales, sutiles, que las pruebas unitarias no detectan.

2.2 Dónde Se Atascan los Equipos

Desafío	Por qué los métodos tradicionales fallan
Definir "terminado"	Los criterios de aceptación requieren pass/fail objetivo — imposible para outputs generativos sin una estrategia de evaluación
Velocidad de sprint	La iteración de prompts no se mapea a story points; un solo cambio de prompt puede llevar minutos o semanas
QA y pruebas	Las pruebas unitarias no cubren comportamiento probabilístico; el QA tradicional omite fallos semánticos
Deuda técnica	El Prompt Debt se acumula silenciosamente — prompts sin documentar, casos extremos sin probar, sin suite de regresión
Actualizaciones del modelo	Una actualización del proveedor puede romper el comportamiento en producción sin ningún cambio de código
Handoff a operaciones	Los runbooks tradicionales no cubren deriva del modelo, monitoreo de alucinaciones ni rollback de versiones de prompts

3. Visión General del Framework AGENI

El framework tiene cinco fases y dos loops de retroalimentación. Piensa en el primer loop como el trabajo diario: diseñas un prompt, lo pruebas, lo refinas y lo repites. Ese ciclo corre en días porque el costo del feedback es bajo. El segundo loop es más lento — se activa cuando algo fuera del control del equipo cambia. Un proveedor actualiza el modelo. Los usuarios comienzan a usar el producto de formas que nadie anticipó. El alcance del producto se expande. Cuando eso ocurre, el equipo regresa al principio y re-examina qué significa "bueno".

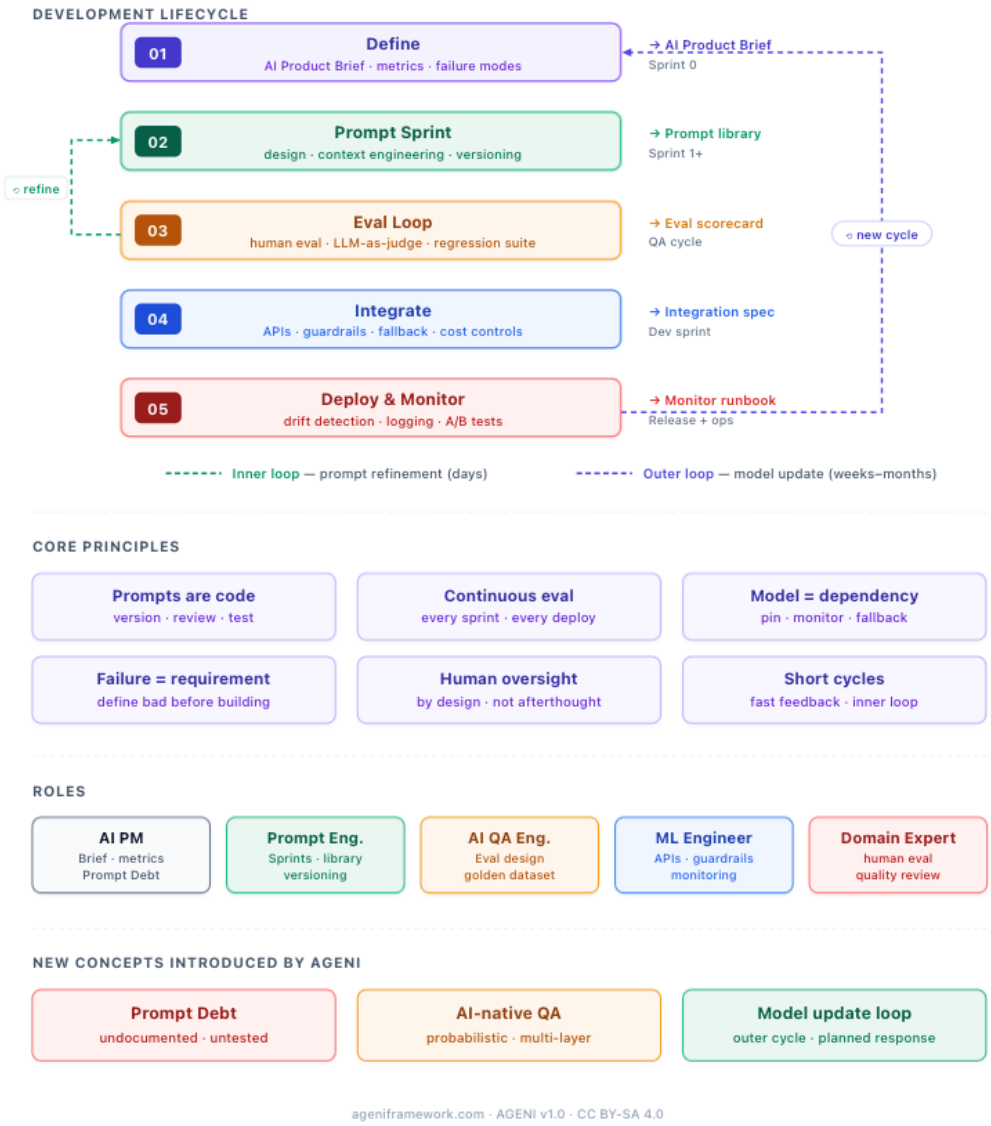

Figura 1 — Mapa completo del framework AGENI: 5 fases, 2 loops de retroalimentación, 6 principios, 5 roles y 3 nuevos conceptos.

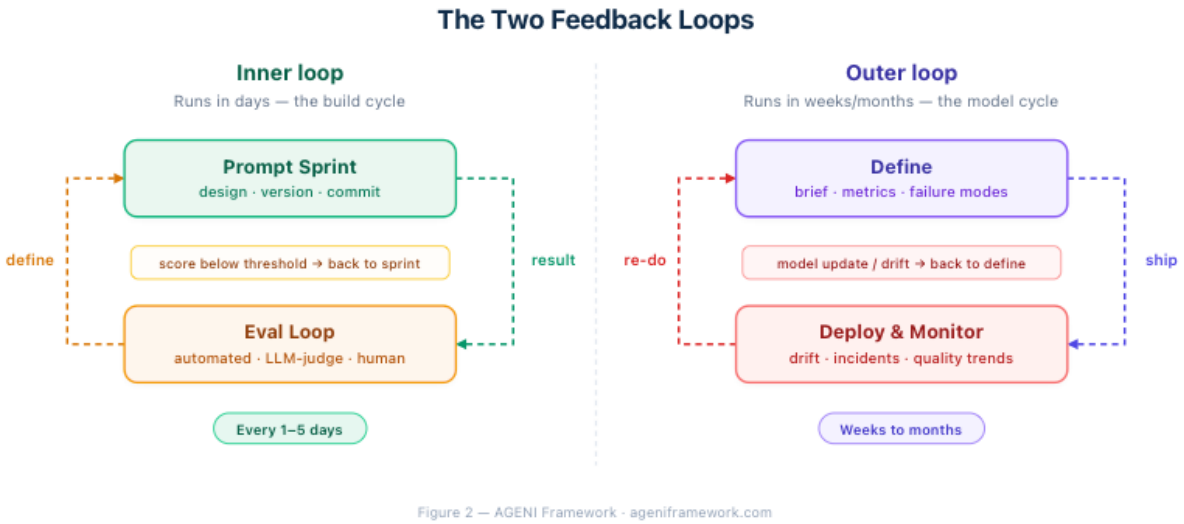

Figura 2 — Los dos loops de retroalimentación: loop interno (días) y loop externo (semanas–meses).

Fase	Enfoque	Entregable clave	Análogo en Agile
01 · Define	Alcance, restricciones, métricas de éxito	AI Product Brief	Sprint 0 / Discovery
02 · Prompt Sprint	Diseño e iteración rápida de prompts	Librería de prompts versionada	Sprint
03 · Eval Loop	Evaluación sistemática de outputs	Eval scorecard + suite de regresión	QA / Ciclo de pruebas
04 · Integrate	Conectar el modelo a los sistemas del producto	Spec de integración + guardrails	Dev sprint
05 · Deploy & Monitor	Observabilidad en producción y detección de deriva	Monitor runbook	Release + ops

4. Las Cinco Fases

01 Define

La mayoría de los equipos omite Define o lo trata como una reunión rápida de alineación. Eso funciona bien cuando el comportamiento de tu producto está determinado por código que escribes. Cuando está determinado por un modelo que no controlas, saltarse Define es donde empiezan los problemas. El objetivo no es escribir una especificación perfecta — es obligar al equipo a responder preguntas que de otra forma se responderán mal, en producción, bajo presión.

El AI Product Brief no necesita ser largo. Lo que necesita cubrir:

– ¿Qué modelo(s) se están considerando y cuáles son sus límites de capacidad?

– ¿Cómo se ve un "buen output"? ¿Cómo lo mediremos?

– ¿Cuáles son los modos de fallo aceptables y cuáles son inaceptables?

– ¿Quién es el humano en el loop, si existe? ¿Dónde tiene autonomía el modelo?

– ¿Qué sucede cuando el proveedor actualiza el modelo?

– ¿Qué datos, contexto o conocimiento necesita el modelo para hacer su trabajo?

> *Define está completo cuando el equipo puede escribir una prueba de evaluación que falla. Si no puedes describir cómo se ve un output malo, no estás listo para construir.*

02 Prompt Sprint

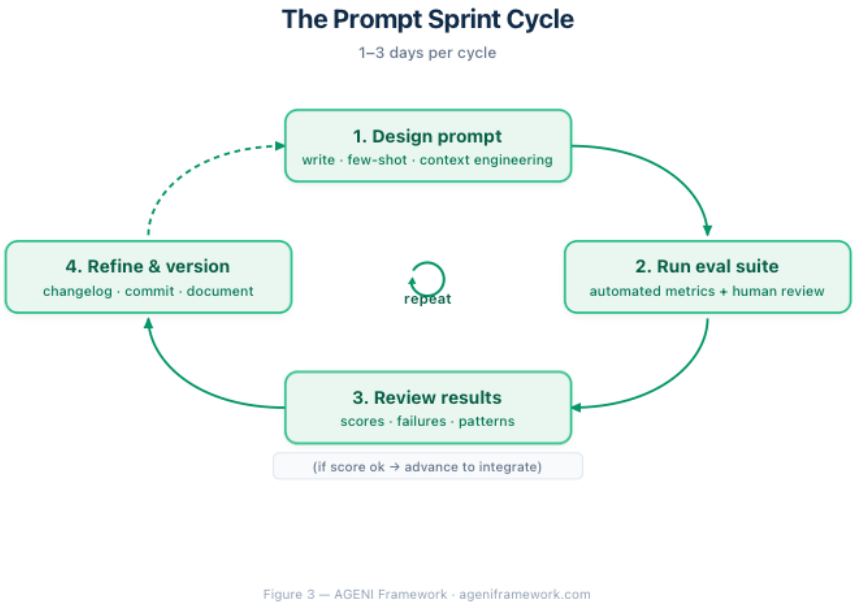

Figura 3 — El ciclo del Prompt Sprint: diseñar, ejecutar, revisar, refinar — repetido cada 1–3 días.

El Prompt Sprint es el ciclo de desarrollo central de AGENI. Es más corto que un sprint tradicional — uno a tres días — porque el loop de retroalimentación es más rápido. Escribes un prompt, lo ejecutas contra tu suite de evaluación, ves qué falla, ajustas y repites. El entregable no es una funcionalidad. Es un prompt o cadena de prompts que ha sido probado, versionado y documentado lo suficientemente bien como para que otra persona del equipo pueda entender por qué funciona y modificarlo sin romper algo. Tres de los cuatro pasos siguientes requieren juicio humano; ejecutar la suite de evaluación es el único paso automatizado.

Qué sucede en un Prompt Sprint:

– Diseño de prompt: escribir o revisar system prompts, plantillas de turno de usuario y ejemplos few-shot — humano, Prompt Engineer

– Ingeniería de contexto: definir qué información recibe el modelo (RAG, outputs de herramientas, memoria) — humano, Prompt Engineer

– Ejecución ligera de evaluación: ejecutar el prompt contra tu conjunto de datos de referencia y revisar los scores — ejecución automatizada, interpretación humana

– Exploración de parámetros: probar temperatura, variantes del modelo, restricciones de formato de output — humano, Prompt Engineer

– Mapeo de casos extremos: identificar inputs que probablemente producirán outputs problemáticos — humano, QA Engineer

– Control de versiones de prompts: registrar prompts en control de versiones con un registro de cambios — humano, Prompt Engineer

> *La ejecución de evaluación dentro de un Prompt Sprint usa las mismas herramientas que la Fase 03 — Eval Loop, pero con un propósito diferente. Aquí guía el trabajo. En la Fase 03, lo valida.*

Los prompts sin documentar ni versionar se acumulan como Prompt Debt — ver Sección 6.

03 Eval Loop

El Eval Loop es donde AGENI es más diferente de cualquier cosa que hayas hecho antes. El QA tradicional es binario — la prueba pasa o falla. La evaluación de IA Generativa no lo es. Un output puede ser parcialmente correcto, contextualmente apropiado pero con tono incorrecto, factualmente preciso pero con formato incorrecto, o perfectamente razonable para el 80% de los inputs y completamente incorrecto para el otro 20%. El Eval Loop es la práctica de medir todo eso, de forma continua, para que la degradación de calidad aparezca antes de que los usuarios la reporten.

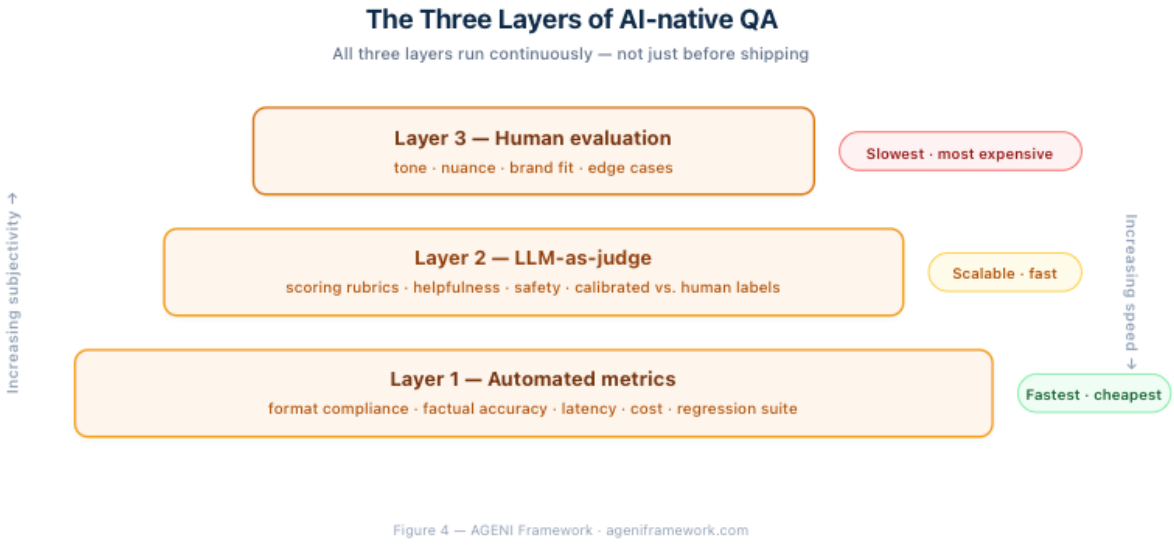

Figura 4 — Las tres capas del AI-native QA: métricas automatizadas, LLM-as-judge y evaluación humana.

Layer 1 (Capa 1) — Métricas automatizadas

– Precisión factual contra ground truth (donde esté disponible)

– Cumplimiento de formato — ¿el output coincide con la estructura requerida?

– Benchmarks de tiempo de respuesta y costo

– Pruebas de regresión contra un conjunto de datos de referencia curado

Layer 2 (Capa 2) — LLM como juez

Usa una llamada separada al modelo para evaluar el output del modelo principal. Define rúbricas de puntuación (utilidad: 1–5, seguridad: pass/fail) y ejecútalas sistemáticamente en el conjunto de datos de evaluación. Calibra los scores del LLM-as-judge contra etiquetas humanas antes de confiar en ellos para decisiones de gate. LLM-as-judge significa usar un segundo modelo de IA para evaluar el resultado del modelo principal — por ejemplo, puntuando si una respuesta fue útil, precisa o apropiada en una escala del 1 al 5.

Layer 2 (Capa 3) — Evaluación humana

Expertos de dominio, usuarios finales o el equipo de producto revisan periódicamente muestras de outputs reales. Aquí es donde se evalúa la calidad subjetiva — tono, matiz, alineación de marca. No todo output puede evaluarse por máquina. En sistemas de alto riesgo (A3), la evaluación humana es requerida, no opcional.

> *El Eval Loop nunca cierra. En AGENI, "terminado" significa evaluado, documentado y monitoreado — nunca probado una vez y olvidado.*

04 Integración

La integración es donde muchos proyectos de IA Generativa se desmoronan. El prompt que funcionaba perfectamente en el playground deja de funcionar cuando llega al producto real — porque la ventana de contexto ahora está llena de datos irrelevantes, porque el parser de output fue escrito para un formato ligeramente diferente, porque no hay fallback cuando el modelo agota el tiempo de espera. La integración en AGENI es la fase donde cierras la brecha entre "funciona en aislamiento" y "funciona en producción".

Antes de publicar, el equipo necesita confirmar:

- Integración de API y SDK con manejo de errores para indisponibilidad del modelo
- Gestión de la ventana de contexto: qué cabe, qué se trunca, qué se recupera
- Skills y system prompts: el prompt de producción (Prompt v1) es el skill — el conjunto de instrucciones persistentes que define el rol del modelo, el tono, las restricciones y el formato de output esperado. Un skill no provee información vigente u oficial por sí solo; especializa el comportamiento. Para entregar datos actualizados, combina el skill con RAG (recuperación desde documentos o bases de datos actualizadas), llamadas a APIs en tiempo real, o contexto pasado directamente en cada solicitud.
- Parseo y validación de output: nunca confíes en el output crudo del modelo en producción
- Guardrails: filtrado de input, filtrado de output, aplicación de política de contenido
- Lógica de fallback: ¿qué sucede cuando el modelo falla, agota el tiempo o produce output inutilizable?
- Controles de costo: presupuestos de tokens, limitación de velocidad, estrategias de caché
- Rutas de escalación humana: ¿cuándo debe un humano revisar o anular al modelo?

05 Despliegue & Monitoreo

Publicar no es la línea de llegada. Si acaso, es donde empiezan los problemas interesantes. Los usuarios reales envían inputs que nunca pensaste. El proveedor del modelo actualiza algo silenciosamente y tu producto comienza a comportarse diferente. La calidad del output deriva de formas que no aparecen como errores — sin 500, sin crash, solo respuestas sutilmente peores que los usuarios notan antes que tu monitoreo.

Qué monitorear en producción:

– Deriva de calidad de output: ¿están empeorando los outputs con el tiempo? ¿Después de actualizaciones del modelo?

– Análisis de patrones de uso: ¿qué inputs están enviando realmente los usuarios?

– Tendencias de costo y tiempo de respuesta: ¿el sistema opera dentro del presupuesto?

– Seguimiento de modos de fallo: ¿qué tipos de fallos están ocurriendo? ¿Con qué frecuencia?

– Señales de feedback de usuarios: calificaciones explícitas, señales implícitas (tasa de reintento, abandono)

El loop externo de retroalimentación

Cuando el monitoreo revela deriva significativa, nuevos modos de fallo o una actualización del modelo del proveedor, el loop externo se activa: regresar a Define, revisar el AI Product Brief y comenzar un nuevo conjunto de Prompt Sprints. Este es el comportamiento esperado y planificado en AGENI — no una crisis.

5. Principios Fundamentales

Seis ideas aparecen a lo largo del framework. Vale la pena enunciarlas desde el principio porque son los lugares donde AGENI pide a los equipos que trabajen de manera diferente a la que están acostumbrados.

Los prompts son código. No metafóricamente — en un producto de IA Generativa, el system prompt es código de producción. Necesita un historial de versiones, un proceso de revisión y una forma de probar que cambiarlo no rompe algo más. Un prompt sin documentación es Prompt Debt esperando a manifestarse.

La evaluación es continua. No hay "terminamos con el QA" en un sistema probabilístico. Ejecutas evaluaciones durante el desarrollo, después del despliegue y cada vez que cambia el modelo subyacente. El momento en que dejas de medir es el momento en que la calidad comienza a derivar de forma invisible.

El modelo es el código de otro. OpenAI, Anthropic, Google — quien sea que provea el modelo puede actualizarlo sin pedirte permiso. Esa actualización puede cambiar el comportamiento de tu producto en producción sin que toques un solo archivo. Trátalo como trataríais cualquier dependencia que no controlas: fija versiones donde sea posible, monitorea los cambios y ten un plan para cuando algo se rompa.

Los modos de fallo son requisitos. Antes de escribir el primer prompt, el equipo debe poder responder: ¿cómo se ve un output malo? ¿Qué es inaceptable bajo cualquier circunstancia? Si no puedes responder eso, no puedes escribir una evaluación significativa — y sin una evaluación significativa, estás adivinando.

La supervisión humana es una decisión de diseño. En algún momento, alguien del equipo necesita decidir: ¿dónde revisa un humano el output antes de que llegue al usuario? ¿Dónde actúa el modelo de forma autónoma? Esa decisión debe ocurrir en la fase de diseño, no después del primer incidente en producción.

Mantén ciclos cortos. Una de las ventajas de construir con IA Generativa es que el loop de retroalimentación puede ser rápido — ejecutar un prompt contra una suite de evaluación lleva

minutos, no semanas. Úsalo. Diseña el trabajo en ciclos cortos para que el equipo aprenda rápidamente y no pase tres semanas construyendo en la dirección incorrecta.

6. Nuevos Conceptos Introducidos por AGENI

Tres términos aparecen a lo largo del framework que no tienen equivalentes directos en Scrum o PMBOK. Vale la pena definirlos claramente porque si las personas los usan de forma imprecisa, el framework deja de ser útil.

Prompt Debt — el equivalente en IA Generativa de la deuda técnica

Se acumula cuando los prompts no están documentados ni versionados, se copian de ejemplos sin entender por qué funcionan, se modifican en producción sin una actualización correspondiente de la evaluación, o se optimizan para la demo en lugar de la distribución completa de inputs reales de usuarios. Como la deuda técnica, el Prompt Debt es invisible hasta que causa un incidente en producción. Se manifiesta como degradación de la calidad del output tras una actualización del modelo, comportamiento inconsistente entre inputs similares, o incapacidad para explicar por qué el producto se comporta diferente de lo esperado. Los equipos AGENI rastrean el Prompt Debt explícitamente como un elemento del backlog y asignan capacidad de sprint para reducirlo.

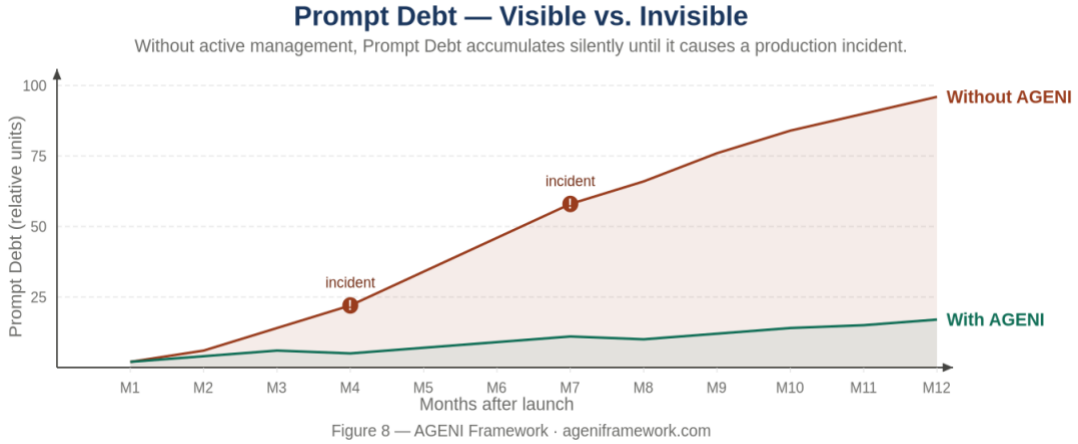

Figura 8 — Acumulación de Prompt Debt: sin gestión activa, la deuda crece hasta causar un incidente en producción.

> *¿Cómo evita AGENI que el Prompt Debt crezca sin control? No lo evita — lo hace visible y manejable. Igual que Scrum no elimina la deuda técnica pero la pone en el backlog para que el equipo la gestione conscientemente, AGENI hace el Prompt Debt visible en el Eval Loop, lo registra como elemento del backlog y asigna capacidad de sprint para*

> *abordarlo. Sin un framework, el Prompt Debt es invisible hasta que causa un incidente en producción.*

AI-native QA — control de calidad rediseñado para sistemas probabilísticos

El AI-native QA es la práctica de evaluar outputs de IA generativa de forma sistemática, aceptando que no existe pass/fail binario. Opera en tres capas: métricas automatizadas (cumplimiento de formato, precisión factual, benchmarks de costo y tiempo de respuesta, pruebas de regresión en un conjunto de datos de referencia), LLM-as-judge (se usa un segundo modelo de IA para evaluar el resultado del modelo principal — por ejemplo, puntuando si una respuesta fue útil, precisa o apropiada en una escala del 1 al 5), y evaluación humana (expertos de dominio revisan muestras para calidad subjetiva). El AI-native QA no es una fase; es una práctica continua que corre en cada Prompt Sprint y continúa en producción vía Deploy & Monitor.

Model update loop — el ciclo externo de retroalimentación activado por cambios en el modelo

Cuando el proveedor de LLM publica una nueva versión del modelo, el model update loop se activa: re-ejecutar la suite de evaluación completa contra el nuevo modelo, analizar el impacto en prompts y guardrails, re-promptear si la calidad degradó, re-validar de forma adversarial, luego decidir si migrar, migrar con fallback o mantener la versión anterior. Este loop corre en un ciclo de semanas a meses — mucho más lento que el loop interno — pero debe planificarse en todo proyecto AGENI. Es el comportamiento que más distingue las operaciones de productos de IA Generativa de las operaciones de software tradicional.

7. ¿Es AGENI una Metodología, una Herramienta o un Framework de Gestión de Proyectos?

Vale la pena responderlo directamente.

7.1 El argumento de "la IA como herramienta"

La objeción va así: la IA es una herramienta. Los equipos siempre han usado herramientas — compiladores, servicios en la nube, IDEs — y esas herramientas nunca requirieron nuevas metodologías. Si usas Claude para escribir código o ChatGPT para redactar una especificación, tienes razón. El output de esos flujos de trabajo sigue siendo código determinista o un documento que un humano revisa antes de que importe. Tu proceso existente maneja eso bien.

La distinción crítica: AGENI aplica cuando el output del modelo ES el producto entregado al usuario — no cuando la IA se usa internamente para construir ese producto. Un equipo que usa Copilot para escribir código determinista no necesita AGENI. Un equipo que construye un asistente virtual, una herramienta de

generación de contenido o cualquier sistema donde el usuario final interactúa directamente con respuestas de un LLM, sí. Importante: ese mismo producto puede tener otras partes — una UI, una base de datos, lógica de negocio — que continúan gestionándose con Scrum, PMBOK u otra metodología. AGENI complementa esas metodologías para la parte específica donde el LLM genera el output al usuario.

7.2 Metodología de implementación vs. framework de gestión de proyectos

AGENI abarca ambas categorías intencionalmente. El problema que aborda existe en ambas capas simultáneamente. Un equipo con excelentes prácticas de ingeniería pero sin gobernanza de PM para actualizaciones del modelo publicará productos que fallan silenciosamente. Un equipo con excelente gobernanza pero sin disciplina de evaluación entregará calidad degradada sin mecanismo para detectarla.

Fase	Metodología de implementación	Gestión de proyectos
01 · Define	—	AI Product Brief, métricas de éxito, inventario de modos de fallo
02 · Prompt Sprint	Diseño de prompts, ingeniería de contexto, versionado	Planificación de sprint, gestión del backlog
03 · Eval Loop	Diseño de suite de evaluación, implementación de LLM-as-judge	Quality gates, criterios de aceptación, decisiones de release
04 · Integrate	Integración de API, guardrails, lógica de fallback	Gestión de dependencias, mitigación de riesgos
05 · Deploy & Monitor	Logging, detección de deriva, pruebas A/B	Gobernanza, reporte a stakeholders, gestión de cambios del modelo

La categoría formal de AGENI es un Framework de Desarrollo de Producto — se sitúa en la intersección de metodología de implementación y gestión de proyectos, porque el problema que resuelve vive en esa intersección.

8. AGENI vs. Metodologías Existentes

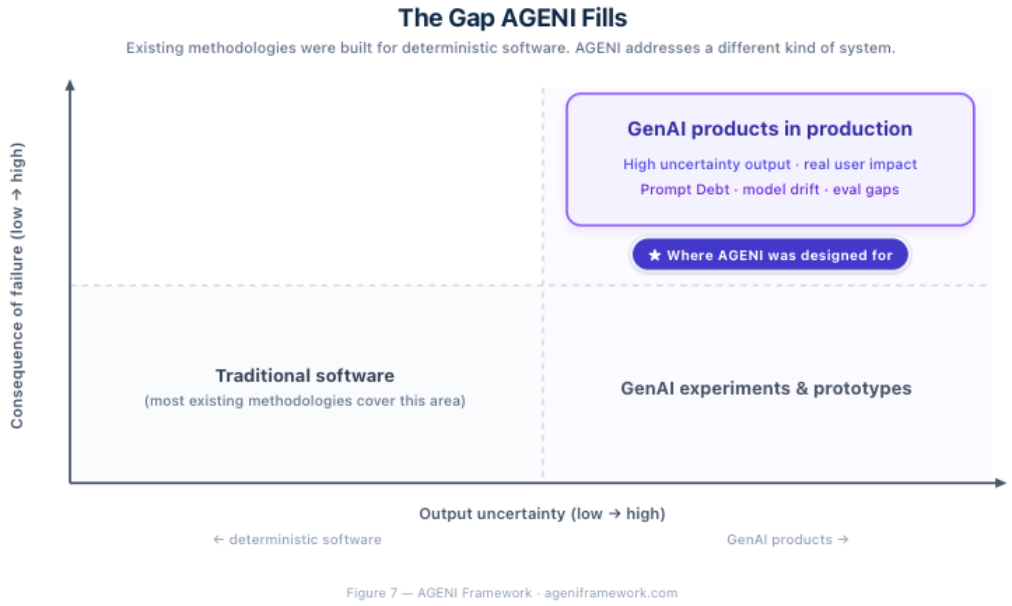

Figura 7 — El vacío que llena AGENI: los productos GenAI en producción combinan alta incertidumbre con impacto real en usuarios.

AGENI no intenta reemplazar Scrum o PMBOK. La mayoría de los equipos que construyen productos de IA Generativa ya tienen una metodología que maneja bien las partes no-IA de su trabajo. La comparación a continuación no es una afirmación de que AGENI supera a los frameworks establecidos — mapea dónde fue diseñado para operar cada uno y dónde aparece el vacío cuando un modelo de lenguaje está en el núcleo del producto. La posición de AGENI en estas comparaciones refleja intención de diseño, no rendimiento validado.

8.1 AGENI vs. Agile / Scrum Tradicional

Concepto	Agile / Scrum tradicional	AGENI
Unidad de trabajo	Historia de usuario / funcionalidad	Experimento de prompt / resultado de evaluación
Definición de "terminado"	Pruebas pasan, funcionalidad enviada	Score de evaluación cumple umbral, documentado, monitoreado
Enfoque de QA	Pruebas unitarias, pruebas de integración	Suites de evaluación, LLM-as-judge, revisión humana

Artefacto de código	Código fuente	Código fuente + librería de prompts
Deuda técnica	Deuda de código	Deuda de código + Prompt Debt
Dependencia externa	Librerías, APIs (versionadas)	Proveedores de modelos (probabilísticos, actualizaciones externas)
Output del sprint	Incremento publicable	Versión de prompt evaluada o hito de integración
Monitoreo en producción	Uptime, errores, tiempo de respuesta	+ Deriva de calidad, fallos semánticos, cambios del modelo

> *Nota: Las historias de usuario pueden coexistir con AGENI, especialmente en la fase Define para establecer el contexto y los objetivos del usuario. La diferencia es que la unidad de progreso medible es el resultado de evaluación — no la funcionalidad entregada.*

8.2 AGENI vs. PMBOK / Waterfall

PMBOK fue construido para proyectos con un inicio, mitad y fin claros — alcance fijo, requisitos definidos, entrega predecible. Los proyectos de IA Generativa raramente funcionan así. Dicho eso, PMBOK todavía tiene piezas útiles: comunicación con stakeholders, frameworks de riesgo (que pueden adaptarse para cubrir riesgo del modelo), planificación presupuestaria y documentación formal para entornos regulados. AGENI y PMBOK pueden coexistir — solo manejan diferentes preocupaciones.

8.3 AGENI vs. MLOps / LLMOps

MLOps y LLMOps (prácticas de infraestructura para operar modelos de IA en producción) manejan el lado de infraestructura — cómo desplegar, monitorear y actualizar modelos técnicamente. AGENI maneja el lado del proceso — cómo el equipo decide qué construir, cómo evaluarlo, quién aprueba qué y cómo responder cuando algo sale mal. Resuelven problemas diferentes y funcionan bien juntos.

8.4 Comparación de metodologías

Dimensión	Scrum	PMBOK	AGENI
Maneja la incertidumbre	Alto	Bajo	Alto
Outputs probabilísticos	Ninguno	Ninguno	Nativo
Velocidad de ciclo corto	Alto	Bajo	Alto
Escalabilidad enterprise	Medio	Alto	Medio
Nativo para GenAI	No	No	Sí
Gestión de prompts	No	No	Sí
Proceso ante cambios del modelo proveedor	No	Parcial	Sí

** Proceso ante cambios del modelo proveedor: si el framework tiene un proceso explícito cuando el proveedor del modelo cambia el modelo subyacente sin la intervención del equipo.*

9. Fortalezas y Limitaciones

Prefiero ser directo: un documento que solo lista fortalezas es material de ventas. Lo que sigue es una mirada honesta a ambos lados.

9.1 Fortalezas

Construido para la incertidumbre. AGENI acepta los outputs probabilísticos como la norma. Cada fase está diseñada bajo el supuesto de que el comportamiento del modelo variará, derivará y cambiará.

Control de versiones y gestión de prompts. Al tratar los prompts como artefactos de primera clase, AGENI previene la forma más común de deuda técnica invisible en proyectos de IA Generativa.

Cultura eval-first. Los quality gates se definen antes de que comience el desarrollo. Este es el mayor fallo de proceso en proyectos de IA Generativa hoy.

Resiliencia ante cambios del modelo. El loop externo de retroalimentación planifica explícitamente para actualizaciones del modelo del proveedor — un escenario que ninguna metodología de PM existente aborda.

Adopción composable. No tienes que adoptarlo todo a la vez. Un equipo que enfrenta caos de prompts puede empezar solo con la disciplina del Prompt Sprint. Un equipo con incidentes en producción puede empezar con Deploy & Monitor. El framework está diseñado para adoptarse por partes.

Puente interfuncional. AGENI da a PMs, ingenieros, expertos de dominio y equipos de QA un vocabulario compartido para la calidad de productos de IA.

9.2 Limitaciones

Sin historial todavía. Esta es la limitación más importante y la que vale la pena enunciar primero. AGENI v1.0 no ha sido formalmente validado. Viene de la práctica — proyectos reales, fallos reales — pero no tiene la base de evidencia que Scrum o PMBOK construyeron en veinte años. Eso requiere tiempo y pilotos.

Curva de aprendizaje. Los equipos entrenados en QA determinista necesitan repensar fundamentalmente qué significa "terminado". Este es un cambio cultural, no solo de proceso.

Inversión en infraestructura de evaluación. Construir conjuntos de datos de referencia, pipelines de LLM-as-judge y suites de regresión requiere esfuerzo real de ingeniería antes de que valga la pena.

Inmadurez del tooling. El ecosistema para versionado de prompts, gestión de evaluaciones y monitoreo específico de IA Generativa todavía es temprano.

La medición de velocidad es más difícil. Los story points no se traducen limpiamente a experimentos de prompt. Los equipos necesitarán desarrollar nuevos proxies para el progreso.

No para productos no-AI-core. Si la IA Generativa es una característica menor en lugar del núcleo del producto, AGENI introduce overhead sin beneficio proporcional.

10. Cuándo NO Usar AGENI

Un framework que pretende funcionar para cada situación no funciona para ninguna.

Escenario	Mejor enfoque
La IA se usa internamente por el equipo (Copilot, Claude para redactar) pero el producto entregado al cliente es determinista — hace lo mismo cada vez	Usa tu metodología existente sin cambios. Las herramientas de IA no cambian tus requisitos de QA.
La IA Generativa es una característica menor, no central	Adopta solo el Eval Loop de AGENI dentro de tu flujo de trabajo existente de Scrum o Kanban.
Alcance fijo, mandato regulatorio, fecha límite inamovible	Usa PMBOK para gobernanza e integra las fases Define y Eval Loop de AGENI para los componentes específicos de IA.
Herramienta interna con prompts restringidos y bien definidos	Evaluación simplificada + versionado de prompts es suficiente. El framework completo es excesivo.
Constructor individual, pre-product-market-fit	Usa AGENI Lite: Define + Prompt Sprint + Eval Loop únicamente.

11. Objeciones Comunes — y Respuestas Honestas

Estas son las preguntas que aparecen cada vez que alguien presenta AGENI a un equipo experimentado de PM o ingeniería. Planteadas tan justamente como sea posible, con las respuestas más honestas disponibles.

1. "Esto es solo Agile con una lista de verificación de IA Generativa."

Parcialmente verdad. AGENI preserva deliberadamente el núcleo de Agile — ciclos cortos, retroalimentación continua, software funcional sobre documentación. Lo que agrega no es cosmético: el Eval Loop no tiene análogo en Scrum, el Prompt Debt es una categoría genuinamente nueva de deuda técnica, y el loop externo de retroalimentación aborda un problema que no existía antes de que las APIs de LLM se convirtieran en infraestructura.

2. "La IA es una herramienta de ejecución, no un motor de metodología."

Correcto — cuando la IA se usa internamente. El alcance de AGENI es más estrecho de lo que parece: aplica solo cuando los outputs del modelo se entregan directamente a los usuarios finales como el valor central del producto. En ese escenario, el modelo no es una herramienta — es el sistema.

3. "MLOps y LLMOps ya resuelven este problema."

MLOps y LLMOps (prácticas de infraestructura para operar modelos de IA en producción) resuelven el problema de infraestructura: cómo servir, monitorear y re-entrenar modelos a escala. AGENI resuelve el problema de proceso del producto: cómo un equipo interfuncional organiza su trabajo. Estas son capas diferentes.

4. "Sin un cuerpo de certificación, esto es solo un blog post con nombre."

Justo. AGENI v1.0 es el comienzo de un cuerpo de conocimiento. Todo framework establecido comenzó como un documento. La medida de legitimidad es la adopción y el refinamiento con el tiempo — no la existencia de una certificación el día uno.

5. "¿Es metodología o framework de PM — parece ser ambos?"

Ambos, deliberadamente. El problema existe en ambas capas simultáneamente. Separarlos crea exactamente el vacío que causa la mayoría de los fallos de productos de IA.

6. "Los story points y la velocidad no funcionan aquí."

Correctamente identificado como una limitación. AGENI propone la progresión del score de evaluación, el throughput de experimentos de prompt y la reducción del Prompt Debt como proxies alternativos. Estos son menos precisos que la velocidad de story points — una limitación honesta en esta etapa.

12. Roles en un Equipo AGENI

No se necesitan nuevos títulos de trabajo. Los roles a continuación mapean títulos existentes a lo que poseen en un proyecto AGENI — las responsabilidades cambian, pero el organigrama no.

Rol	Responsabilidad principal en AGENI	Nueva habilidad requerida
AI Product Manager	Posee el AI Product Brief, define criterios de éxito de evaluación, gestiona el backlog de Prompt Debt	Diseño de evaluación, conciencia de capacidades de LLM
Prompt Engineer / Developer	Ejecuta Prompt Sprints, mantiene la librería de prompts, posee el versionado y la suite de regresión	Diseño de prompts, ingeniería de contexto
AI QA Engineer	Diseña y ejecuta Eval Loops, mantiene conjuntos de datos de referencia, interpreta métricas de calidad	Técnicas de LLM-as-judge, evaluación semántica
ML / AI Engineer	Integración, guardrails, infraestructura del modelo, pipeline de monitoreo	APIs de modelos, sistemas RAG, validación de output
Domain Expert / Reviewer	Evaluación humana de outputs, define qué significa "bueno" en contexto	Feedback estructurado para outputs de IA

As with other PM Methodologies, one person could play multiple roles in smaller teams.

13. Cómo Empezar con AGENI

No intentes implementarlo todo a la vez. Elige el problema que está causando más dolor ahora mismo y empieza por ahí.

Si tu mayor problema es la calidad del output:

– Empieza con el Eval Loop. Construye un conjunto de datos de referencia de 50–100 inputs representativos y outputs esperados. Ejecútalo después de cada cambio de prompt.

Si tu mayor problema es el caos de prompts:

– Empieza con la disciplina del Prompt Sprint. Versiona tus prompts en Git, escribe un registro de cambios para cada modificación y requiere una ejecución de regresión antes de hacer merge.

Si tu mayor problema son los incidentes en producción:

– Empieza con Deploy & Monitor. Agrega logging de calidad de output, configura alertas para cambios en la API del modelo y define un procedimiento de rollback para versiones de prompts.

Si tu mayor problema es la planificación y el alineamiento:

– Empieza con Define. Usa el AI Product Brief para forzar decisiones explícitas sobre métricas de éxito, modos de fallo y supervisión humana antes de que comience el desarrollo.

Si eres un constructor individual o equipo en etapa temprana:

– Usa AGENI Lite: Define + Prompt Sprint + Eval Loop. Omite la especificación formal de integración y el monitor runbook hasta que la escala lo exija.

> *Si aplicas AGENI a un proyecto real, lo más útil que puedes hacer por el framework es medir qué sucede. Antes de empezar: define qué rastrearás (tiempo desde el cambio de prompt hasta la decisión de release, número de items de Prompt Debt encontrados, tiempo para detectar deriva). Después de terminar: comparte qué funcionó y qué no. Esa evidencia es lo que convierte un framework conceptual en uno validado. AGENI v1.0 necesita pilotos más que advocates.*

14. Gates de Decisión — G0 a G3

Figura 5 — Gates de decisión G0–G3 mapeados a las cinco fases de AGENI.

Sin puntos de decisión claros, un framework es solo consejo. Los cuatro gates a continuación definen cuándo detenerse y verificar antes de avanzar. Cada uno tiene una pregunta simple, un conjunto mínimo de evidencia requerida para responderla, quién toma la decisión y qué causa un stop definitivo.

> *Los gates a continuación son checkpoints de gobernanza propuestos, no procedimientos probados en batalla. Representan la estructura que un equipo piloto adoptaría y refinaría a través del uso real.*

G0 — Viabilidad — ¿Debemos usar IA Generativa para esto?

Pregunta de decisión	¿Debemos usar IA Generativa para este problema?
Evidencia mínima	Caso de uso definido con usuario e impacto esperado. Alternativa no-IA evaluada. Restricciones iniciales identificadas (costo, privacidad, cumplimiento). Riesgos clave documentados.
Aprobador	Business Sponsor con consulta del PM
Condición de bloqueo	Ninguna alternativa no-IA fue evaluada. El caso de uso no justifica la variabilidad inherente en los outputs generativos.

G1 — Diseño Evaluable — ¿Podemos construir y medir esto?

Pregunta de decisión	¿Podemos construir y medir esta solución?
Evidencia mínima	Métricas de calidad definidas y cuantificables. Conjunto de datos de evaluación inicial creado. Modos de fallo identificados. Arquitectura de modelo, RAG y tooling propuesta.

Aprobador	PM con consulta del QA Engineer
Condición de bloqueo	Sin forma de medir la calidad del output. Sin conjunto de datos de evaluación. Modos de fallo desconocidos.

G2 — Listo para Producción — ¿Es el riesgo aceptable para publicar?

Pregunta de decisión	¿Es el riesgo aceptable para publicar en producción?
Evidencia mínima	Resultados del Eval Scorecard dentro de umbrales. Guardrails probados de forma adversarial. Fallback configurado y probado. Costo dentro del presupuesto. Observabilidad activa. Kill-switch probado. Riesgo residual aceptado por el Sponsor.
Aprobador	PM (go/no-go). Sponsor (riesgo residual). Veto de Legal/Compliance si está regulado.
Condición de bloqueo	Eval Scorecard falla. Guardrails no cubren riesgos críticos. Fallback sin probar. Riesgo residual no aceptado.

G3 — Operación Controlada — ¿Sigue siendo aceptable en producción?

Pregunta de decisión	¿El sistema sigue siendo aceptable en producción?
Evidencia mínima	Tendencias de calidad estables o mejorando. Sin incidentes Sev1 no resueltos. Feedback de usuarios dentro de tolerancia. Cambios del proveedor del modelo evaluados. Monitor de deriva activo.
Aprobador	QA Engineer (métricas de calidad). PM (decisión de mantener/revertir).
Condición de bloqueo	Deriva crítica detectada. Incidente Sev1 activo. Actualización del modelo sin re-evaluación. Degradación de calidad sostenida.

15. Niveles de Rigor — A1, A2 y A3

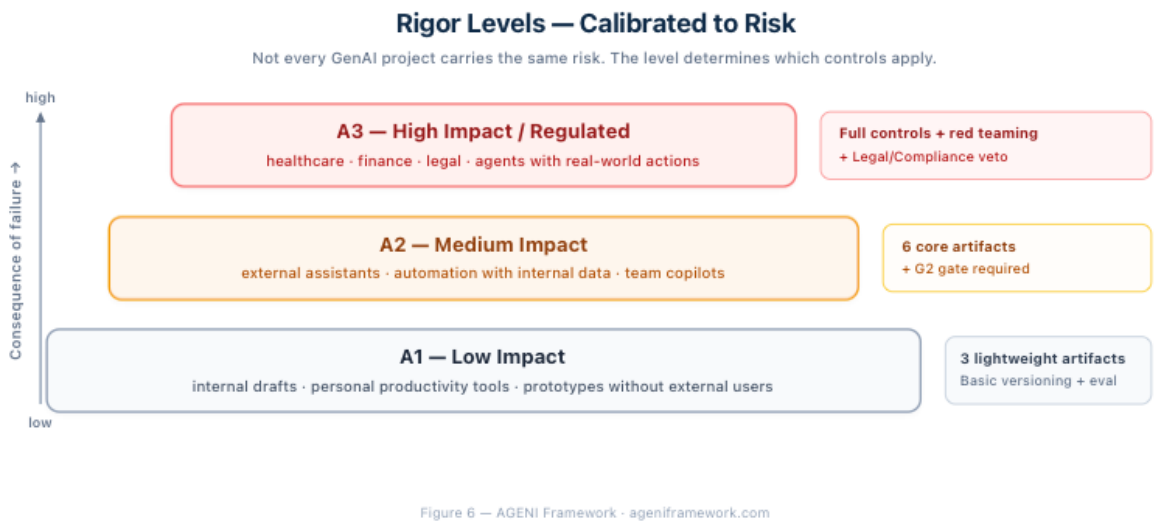

Figura 6 — Niveles de rigor A1, A2, A3: controles calibrados a la consecuencia del fallo.

No todo proyecto de IA Generativa conlleva el mismo riesgo. Una herramienta que ayuda a un empleado a escribir borradores internos es diferente de un sistema que responde preguntas médicas. Los tres niveles a continuación permiten a los equipos calibrar cuánto proceso aplicar en función de lo que sucede si algo sale mal.

> *Aplicar controles A3 a un proyecto A1 desperdicia recursos. Aplicar controles A1 a un proyecto A3 introduce riesgo no gestionado.*

A1 — Bajo Impacto

Casos típicos: generación de borradores internos, herramientas de productividad personal, prototipos sin usuarios externos.

Requisitos mínimos:

– AI Product Brief — ligero (una página)

– Librería de Prompts — versionado básico

– Eval Scorecard — métricas básicas, feedback de usuarios como señal principal

– Fallback — respuesta predeterminada simple

– Monitoreo de costo — seguimiento básico de uso

A2 — Impacto Medio

Casos típicos: asistentes externos (no regulados), automatización con datos internos, copilots de productividad.

Requisitos mínimos:

– AI Product Brief — completo, con métricas cuantificables

– Librería de Prompts — versionada con registro de cambios y revisión de QA para cambios mayores

– Eval Scorecard — umbrales definidos, LLM-as-judge calibrado

– Guardrails Spec — pruebas adversariales requeridas

– Protocolo HITL — umbrales de escalación definidos

– Monitoreo de deriva activo

– Aprobación de Gate G2 antes de producción

A3 — Alto Impacto / Regulado

Casos típicos: salud, finanzas, legal, agentes con permisos para actuar en sistemas externos, exposición pública masiva.

Todos los requisitos A2, más:

– Revisión humana formal en el loop de evaluación

– Red teaming adversarial antes de producción

– Kill-switch probado bajo condiciones de fallo

– Plan de respuesta a incidentes con roles asignados

– Revisión y veto de Legal/Compliance

– Post-mortem requerido después de cualquier incidente Sev2+

16. Roles y Responsabilidades

16.1 Los cinco roles

En equipos pequeños, una persona tendrá múltiples responsabilidades. Lo que importa es que las responsabilidades estén asignadas.

Rol	Responsabilidad principal	Nueva habilidad requerida
AI Product Manager (PM)	Posee el AI Product Brief, define criterios de éxito, gestiona el backlog de Prompt Debt, firma el go/no-go en G2	Diseño de evaluación, conciencia de capacidades GenAI
AI Engineer (AIE)	Ejecuta Prompt Sprints, mantiene la Librería de	Diseño de prompts, ingeniería de contexto, RAG

	Prompts, construye guardrails e integración	
QA / Eval Engineer (QAE)	Diseña y ejecuta Eval Loops, posee el conjunto de datos de referencia, define umbrales de calidad — no puede ser anulado por el PM o AIE en resultados de evaluación	Calibración de LLM-as-judge, evaluación semántica
ML / Platform Engineer (MLO)	Despliegue, infraestructura, observabilidad, respuesta a incidentes — pre-autorizado para kill-switch sin aprobación del PM en Sev1/Sev2	LLMOps, pipelines de monitoreo
Business Sponsor (SPN)	Financia el proyecto, acepta el riesgo residual en G2, punto final de escalación	Alfabetización en riesgos de IA

Legal/Compliance es un rol condicional — no un participante permanente. Se activa cuando el sistema procesa PII, datos de salud, financieros o legales; cuando los outputs se exponen públicamente; o cuando un dominio regulado está involucrado.

16.2 RACI Simplificado

R = Responsable. A = Accountable (uno por fila). C = Consultado. I = Informado.

Artefacto / Decisión	PM	AIE	QAE	MLO	SPN
AI Product Brief	A/R	C	C	I	C
Librería de Prompts	I	A/R	C	—	—
Eval Scorecard	C	C	A/R	I	I
Guardrails Spec	I	A/R	C	I	—

Gate G0 — Viabilidad	C	C	—	—	A
Gate G1 — Evaluable	A	C	C	—	I
Gate G2 — Producción	A	C	C	C	A*
Gate G3 — Ops en curso	C	—	A	R	I
Kill-switch (Sev1/Sev2)	I**	R	—	A	I
Backlog de Prompt Debt	A	R	C	—	—

** El SPN firma la aceptación de riesgo residual en G2 — no la decisión de go/no-go en sí.*

*** El PM es informado post-acción en 15 minutos — no es pre-aprobación.*

17. Definición de Terminado — Por Fase

En el Agile tradicional, "terminado" significa que el código funciona y las pruebas pasan. En el desarrollo de IA Generativa, terminado significa algo más: el output ha sido evaluado, la configuración está rastreada y la calidad está documentada.

Fase	Terminado significa...	Gate habilitado
01 · Define	Las métricas de éxito son cuantificables. Los modos de fallo están documentados. El Sponsor está alineado. Se consideró una alternativa no-IA.	G0 — Viabilidad
02 · Prompt Sprint	Los prompts están versionados en control de versiones con una entrada en el registro de cambios. Los parámetros están documentados. Al menos una ejecución de evaluación ha sido ejecutada.	G1 — Diseño Evaluable
03 · Eval Loop	Los resultados del Eval Scorecard son reproducibles. Las métricas dentro de los	Entrada para G2

	umbrales definidos o la decisión de iterar está documentada. Sin fallos críticos sin resolver.	
04 · Integrate	Los guardrails están probados de forma adversarial. El fallback está configurado y probado. La observabilidad está activa. El kill-switch está probado. La configuración está completamente rastreada.	G2 — Listo para Producción
05 · Deploy & Monitor	Los umbrales de deriva están definidos y monitoreados. La respuesta a incidentes está en su lugar. Los resultados de evaluación de referencia documentados para comparación futura.	G3 — Operación Controlada

18. Qué Afirmar — y Qué Evitar

La credibilidad viene de decir lo que realmente sabes — no lo que suena bien. Las tablas a continuación separan las afirmaciones que AGENI puede respaldar de las que se desmoronarían bajo escrutinio.

18.1 Afirmaciones a evitar

Afirmación	Por qué evitarla
"AGENI es el primer framework para IA Generativa"	No es demostrable. MLOps, LLMOps y prácticas de gobernanza de IA ya existen.
"Estos conceptos no existen en ningún otro lugar"	El versionado de prompts, LLM-as-judge y el monitoreo de deriva son conocidos en LLMOps. El valor está en integrarlos en un flujo de trabajo accesible para PMs.
"AGENI reemplaza a Scrum o PMBOK"	AGENI complementa las metodologías existentes. Llena el vacío que dejan para sistemas probabilísticos.
"AGENI garantiza calidad en sistemas de IA Generativa"	Ningún framework garantiza calidad en sistemas probabilísticos. AGENI gestiona el riesgo y hace visible la degradación antes.

"AGENI es irrefutable"	Nada en gestión de proyectos es irrefutable. Las prácticas se validan con evidencia del uso real.

18.2 Afirmaciones defendibles

Afirmación	Por qué se sostiene
"Un framework para equipos donde el output del modelo es lo que el usuario ve"	Precisamente delimitado. Excluye a equipos que usan IA como herramienta interna de desarrollo.
"Integra evaluación, gobernanza y operaciones para IA Generativa"	Las prácticas individuales existen en otros lugares. La integración en una estructura accesible para PMs es la contribución.
"Complementa Scrum, Kanban, PMBOK y LLMOps"	Preciso y facilita la adopción. Los practicantes no tienen que abandonar lo que saben.
"Adapta el rigor al nivel de riesgo del proyecto (A1–A3)"	Una característica diferenciadora y verificable.
"Trata las actualizaciones del modelo como un evento planificado, no un incidente de producción"	Un problema genuinamente desatendido que ninguna metodología de PM existente aborda explícitamente.

La posición más sólida que AGENI puede adoptar es la honestidad intelectual. Un framework que reconoce sus propios límites, nombra lo que no cubre y se niega a prometer de más ganará más confianza de los practicantes experimentados que uno que pretende resolverlo todo. Esa confianza es lo que impulsa la adopción.

Te invito a usarlo y enviarme cualquier feedback - ageni.framework@gmail.com