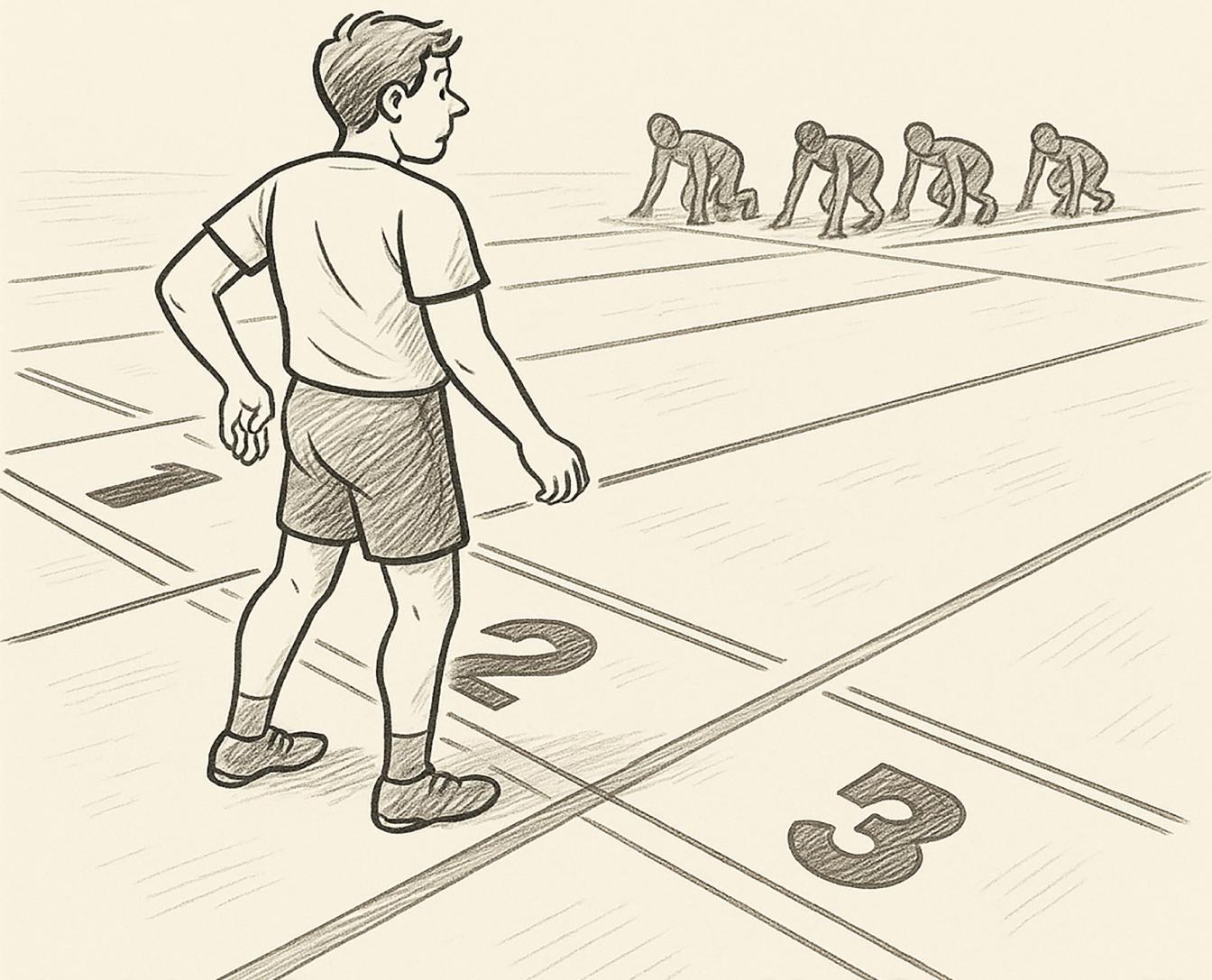


READY

WHY MOST SOFTWARE PROJECTS
FAIL AND HOW TO FIX IT



Luniel de Beer

Max Guernsey, III

Edición en Español

¡Tuitea Este Libro!

¡Ayuda a Luniel de Beer y Max Guernsey, III difundiendo la palabra sobre este libro en [Twitter](#)!

El tuit sugerido para este libro es:

Acabo de comprar Ready — un libro para líderes de software y equipos que desean eliminar el retrabajo, el arrastre de trabajo y la desalineación en la entrega de software. [#CodeReady](#)

El hashtag sugerido para este libro es [#CodeReady](#).

Descubre lo que otras personas están diciendo sobre el libro haciendo clic en este enlace para buscar este hashtag en Twitter:

[#CodeReady](#)

Ready (Edición en Español)

Por qué la Mayoría de los Proyectos de Software
Fracasan y Cómo Solucionarlo

Luniel de Beer y Max Guernsey, III

Este libro está disponible en <https://leanpub.com/ready-es>

Esta versión fue publicada el 2025-10-22



Este es un libro de [Leanpub](#). Leanpub empodera a autores y editores con el proceso de Lean Publishing. [Lean Publishing](#) es el acto de publicar un libro electrónico en progreso utilizando herramientas ligeras y múltiples iteraciones para obtener retroalimentación de los lectores, pivotar hasta tener el libro correcto y generar tracción una vez logrado.

© 2025 Luniel de Beer y Max Guernsey, III

En memoria de Johann van Aardt, quien reconoció mi pasión, me introdujo a la programación real y me ayudó a encontrar el camino hacia un nuevo hogar. Y a mis padres, cuyo apoyo incondicional —desde mi primer cliente hasta mi primer hogar en los EE. UU.— hizo todo esto posible.

—Luniel

Para mi familia, con quienes sale y se pone el sol.

—Max

Índice general

| | |
|--|-----------|
| Acerca de Este Libro | i |
| Para Quién Es Este Libro | ii |
| Cómo Usar Este Libro | iii |
| Sobre los Autores | iv |
| Prefacio | v |
| | |
| Parte I: Algo Falta | 1 |
| Capítulo 1: El Problema Oculto | 2 |
| Capítulo 2: El Costo de las Bases Faltantes | 14 |
| Capítulo 3: Introducción al Requirements Maturation Flow (RMF) | 26 |
| Capítulo 4: ¿Es Ágil? | 28 |
| | |
| Parte II: Generando Espacio para la Preparación | 29 |
| Capítulo 5: La Primera Extensión | 30 |
| Capítulo 6: ¿Por Qué La Gente No Hace Esto? | 31 |
| Capítulo 7: Trabajo Explícito de Preparación (RMF 1) | 33 |
| Capítulo 8: Efectos del RMF 1 | 35 |
| Capítulo 9: Poniendo en Práctica RMF 1 | 36 |

| | |
|---|-----------|
| Parte III: Controlando la Finalización del Trabajo | 38 |
| Capítulo 10: La Siguiente Necesidad | 39 |
| Capítulo 11: Lo que la Gente Suele Hacer | 41 |
| Capítulo 12: Definiendo una Definición de Terminado | 43 |
| Capítulo 13: Definición de Terminado Personalizada (RMF 2) | 45 |
| Capítulo 14: La Vida Con RMF 1 y 2 | 49 |
| Capítulo 15: Instalación de RMF 2 | 51 |
| Parte IV: Implementación de Control de Fases | 53 |
| Capítulo 16: El Último Requerimiento | 54 |
| Capítulo 17: Antecedentes sobre la Definición de Preparación | 56 |
| Capítulo 18: Definiendo una Definición de Listo | 58 |
| Capítulo 19: Definición de Preparado Personalizada (RMF 3) | 60 |
| Parte V: Síntesis | 64 |
| Capítulo 20: La <u>Mayoría</u> de los Plazos No Importan | 65 |
| Capítulo 21: Competencia 1: Flujo de Maduración de Requisitos | 68 |
| Capítulo 22: Cómo Fluyen el Trabajo y la Información en Scrum con RMF | 70 |
| Capítulo 23: El Impacto del RMF | 72 |
| Capítulo 24: Transición a RMF | 73 |
| Capítulo 25: Depende de Ti | 75 |
| Parte VI: Recursos | 76 |
| Apéndice A: Scrum no es el problema | 77 |

| | |
|---|-----------|
| Apéndice B: El Synapse Framework™ | 79 |
| Apéndice C: Objeciones y Obstáculos Comunes al RMF 1 | 81 |
| Apéndice D: Listas Iniciales de Criterios para la DT | 82 |
| Apéndice E: Listas Iniciales de Criterios de DoR | 83 |
| Índice | 84 |

Acerca de Este Libro

Ready es un libro para cualquier persona involucrada en el desarrollo de software que esté cansada de la **entrega insuficiente, el retrabajo crónico y los requerimientos poco claros**.

Es posible que ya hayas intentado invertir en habilidades de ejecución del equipo, mejorar la implementación de tu marco de proceso o renovar el código y aún necesites más mejoras.

Esto se debe a que la principal limitación para la mayoría de los equipos de desarrollo de software no son las habilidades del equipo, sino la madurez de los requerimientos. **Incluso los equipos maduros con las habilidades adecuadas siguen teniendo dificultades** cuando trabajan con requerimientos inmaduros.

Ready introduce el RMF (Flujo de Maduración de Requerimientos), un enfoque práctico y profundamente estructurado para alinear Producto e Ingeniería sin reemplazar tu proceso existente.

Ya sea que uses Scrum, Kanban o algo personalizado, RMF te ayuda a **estabilizar el alcance, eliminar el arrastre de trabajo y entregar lo que realmente importa**.

Si tus equipos se sienten estancados al borde de “casi terminado”, este libro te mostrará cómo **romper el ciclo y desbloquear tu(s) equipo(s) definitivamente**.

Para Quién Es Este Libro

Este libro es literalmente para cualquier persona involucrada en el desarrollo de software. Desde ingenieros hasta gerentes de producto, y desde contribuyentes individuales hasta ejecutivos.

Este libro es para ti si estás involucrado en el desarrollo de software y has notado que un equipo en el que trabajas o del que formas parte tiene uno o más de los siguientes problemas:

- El trabajo frecuentemente se arrastra de una iteración a la siguiente
- Los equipos de implementación sienten que están tratando de alcanzar objetivos móviles
- El trabajo se mantiene abierto durante demasiado tiempo
- El trabajo se marca como cerrado pero las cosas no están realmente terminadas
- El trabajo realizado no se alinea con las expectativas
- El trabajo genera regularmente una gran cantidad de defectos
- Grandes cantidades de trabajo tienen que rehacerse regularmente

Si alguno de estos problemas te resulta familiar, *Ready* puede ayudarte.

Cómo Usar Este Libro

Este libro fue diseñado para ser práctico. No es un tratado teórico ni una presentación estratégica: es un manual práctico para implementar RMF (el Flujo de Maduración de Requerimientos) basado en trabajo real con clientes y probado en campo bajo presión real de entrega.

Los capítulos están escritos en secuencia, pero RMF en sí es modular. Está compuesto por tres prácticas fundamentales:

- RMF 1: Colaborar para un entendimiento compartido
- RMF 2: Controlar la finalización del trabajo usando Definiciones Personalizadas de Terminado
- RMF 3: Controlar la implementación usando Definiciones Personalizadas de Listo

Cada parte, o “Hábito”, como los llamamos, funciona por sí solo, pero se construyen uno sobre otro. El libro está diseñado para ayudarte a abordarlos uno a la vez, en orden. Esta estructura refleja cómo recomendamos que los equipos adopten RMF en la práctica: con cada Hábito incorporado solo después de que el anterior esté funcionando.

Esto evita abrumar a los equipos y da a cada cambio la mejor oportunidad de arraigarse. Aprenderás más sobre cómo hacer esto comenzando en .

Si estás buscando ayuda, ya sea consejo, coaching o alguien para hablar con tu equipo de liderazgo, no dudes en contactarnos directamente.

Y si estás buscando apoyo formal para implementar RMF, Producore ofrece una serie completa de Programas diseñados para guiar la adopción paso a paso. Puedes obtener más información en <https://ready-book.link/rmf>.

Sobre los Autores

Luniel de Beer es el creador del Flujo de Maduración de Requerimientos (RMF), un sistema práctico para corregir las brechas entre la intención del producto y la ejecución de ingeniería. Tiene más de 15 años de experiencia liderando transformaciones Ágiles, conectando producto e ingeniería, y ayudando a los equipos a entregar con claridad y confianza.

Luniel también originó el sistema de Gestión de Capacidades de Producore, un enfoque trazable y escalable para modelar capacidades de producto. Concibió el Desarrollo Dirigido por PKB (PKBDD), un sistema versionado para gestionar requerimientos persistentes de producto. Estas herramientas forman parte de un marco de entrega más amplio desarrollado en Producore.

Max Guernsey, III es arquitecto de software, educador y cofundador de Producore, una consultora dedicada a corregir fallos de entrega a través del rigor estructural y técnico. Con más de dos décadas de experiencia en diseño orientado a objetos, refactorización, desarrollo dirigido por pruebas y patrones de diseño, ha entregado sistemas críticos y ha capacitado a equipos de ingeniería a escala. Su trabajo combina prácticas técnicas profundas con transformación conductual y de procesos para ayudar a las organizaciones a lograr una excelencia sostenible en la entrega.

Max contribuyó significativamente a PKBDD y lideró el desarrollo del enfoque de Producore hacia el Desarrollo Guiado por Comportamiento (BDD) a través de su profunda experiencia en especificación del comportamiento.

Juntos, su trabajo integra la claridad, trazabilidad y control de calidad en un sistema cohesivo para la entrega de software que escala desde la práctica del equipo hasta la capacidad organizacional.

Prefacio

Nota para Líderes de Ingeniería

Si eres un líder senior en una organización de ingeniería, probablemente no te falte esfuerzo, disciplina o personas inteligentes. Y sin embargo, de alguna manera, los proyectos siguen estancándose. Los objetivos se retrasan. Las expectativas no se cumplen. No porque tus equipos sean perezosos, sino porque algo fundamental está roto en la forma en que el trabajo se define, se moldea y se entrega.

Este libro no es una guía de liderazgo. Es una herramienta para el diagnóstico estructural. Revela lo que realmente está sucediendo dentro de tus equipos—por qué “casi terminado” sigue convirtiéndose en “no terminado”, y por qué el progreso local raramente se traduce en resultados estratégicos.

Puede que no te veas a *ti mismo* en estas páginas. Pero si tus equipos no pueden entregar lo que necesitas, los verás a *ellos*. Y cuando lo hagas, finalmente tendrás el lenguaje—y el sistema—para arreglarlo.

De Luniel

En primer lugar, este libro no habría sido posible sin Max, cuya capacidad para ver a través de la niebla y la paja y destilar una idea hasta su esencia está absolutamente fuera de mi alcance.

¿Cómo llegamos aquí?

Si miro hacia atrás, creo que es porque siempre he querido entender cómo funcionan realmente las cosas. Ya fuera religión, nutrición o desarrollo de software, seguía encontrándome con el mismo problema: respuestas superficiales que no se sostienen bajo presión. Así que seguí cavando—preguntando no solo qué hacemos, sino por qué, y qué falta cuando no funciona.

Una de las primeras grietas en el sistema apareció en un rol donde llevaba tres sombreros: Scrum Master, Product Owner y Development Manager (!! para un equipo que entregaba

servicios de datos en una conocida empresa tecnológica. Estábamos haciendo lo que Scrum decía—Sprints cortos, historias en un backlog, planificación en medio día—pero cada vez que comenzábamos un nuevo Sprint, encontrábamos fricción. El equipo no entendía completamente el problema, teníamos que revisar y modificar requisitos durante el Sprint, surgían dependencias prevenibles que nos retrasaban, y se pasaban por alto pasos clave.

Así que empecé a hacer algo diferente. Reunía al equipo y a los interesados en una sala para cada historia, analizábamos el problema en detalle, ideábamos la solución juntos, y solo entonces escribía la historia. La planificación del Sprint se redujo a una hora, y nuestro éxito en las entregas se disparó.

Poco a poco, comencé a darme cuenta de que el éxito no viene de trabajar más duro dentro del Sprint. Viene de la estructura que estableces *antes* de que comience.

Más tarde, después de escuchar a Jeff Sutherland hablar sobre las “Definiciones de Preparado”, supe que había algo valioso allí—pero no era suficiente. Mi experiencia con requisitos, UX, UI, investigación, y más tarde con BDD me mostró que diferentes elementos de trabajo exigen diferentes tipos de preparación. Algunos necesitan especificaciones de comportamiento. Algunos necesitan acceso al sistema. Algunos necesitan un rastreo completo de capacidades.

Y todos ellos necesitan un entendimiento compartido que esté realmente confirmado—no asumido.

A medida que trabajaba con más equipos, veía el mismo patrón en todas partes: pasos faltantes, dependencias no cumplidas, equipos haciendo su mejor esfuerzo pero constantemente luchando para arreglar problemas que deberían haberse prevenido. Incluso los grandes equipos tenían dificultades—no porque fueran débiles, sino porque les faltaba una estructura que hiciera explícita la preparación.

El resultado de todo ese aprendizaje, iteración y frustración es un sistema estructurado para gestionar la preparación.

De eso trata este libro.

Espero que te dé claridad sobre dónde están los problemas reales y cómo solucionarlos. Espero que te dé el lenguaje para defender prácticas que pueden parecer “extras” pero que son realmente esenciales. Y sobre todo, espero que ayude a los equipos a entregar con menos estrés, menos sorpresas y resultados mucho mejores.

Si lo hacemos bien, ahorraremos miles de millones de dólares a la industria.

Pero más importante aún, devolveremos la cordura a las personas.

De Max

He estado trabajando en este problema desde varios ángulos durante décadas, pero mi progreso se vio obstaculizado hasta que conocí a Luniel.

Esto es porque, antes de conocerlo, estaba abordando el problema, fundamentalmente, como uno técnico. Estaba enfocado en ayudar a los equipos a adoptar cosas como el Desarrollo Guiado por Pruebas (TDD), la refactorización, el diseño avanzado de software y, más tarde, el Desarrollo Guiado por Pruebas de Aceptación (ATDD) o el Desarrollo Guiado por Comportamiento (BDD).

En la mayoría de esos casos, el problema abordado en este libro era tratado como un detalle de implementación para establecer esas prácticas técnicas.

Esto no quiere decir que ya no valore las prácticas técnicas. Todavía creo que son profundamente importantes, pero no abordan directamente el problema de la preparación en el desarrollo de software. En cambio, *sacan a la luz* ese problema y luego la gente pone un parche en su proceso para abordarlo “lo suficiente” como para apoyar las prácticas técnicas que están tratando de implementar.

También quiero abordar la cuestión de a quién puede ayudar este libro. La respuesta corta es “probablemente a casi todos en el desarrollo de software”, pero la respuesta real contiene matices que ayudan a mapearlo a varios entornos sin cambiar el significado básico.

Hay equipos que necesitan la solución proporcionada en este libro. Conocerás una versión depurada de uno de ellos en el Capítulo 1.

También hay equipos que no **necesitan** estrictamente un sistema como el que proponemos, pero aún así podrían beneficiarse de él.

El mejor equipo con el que he trabajado —fácilmente una unidad estándar de desviación completa por encima del **siguiente mejor equipo**, si no dos— estaba ubicado en las tierras del interior de Central Oregon. Tenían un rendimiento tan alto que podían superar la ausencia de tal sistema por puro volumen de competencia. Sin embargo, mi gerente en ese momento, Tom Barreras, me dijo algo parecido a “He notado que nuestras historias van mejor cuando dedicamos algo de tiempo a hablar sobre las pruebas desde el principio.”

Esto, nuevamente, era algo que yo veía en ese momento a través del lente del desarrollo de pruebas y la ejecución técnica, pero ahora sé que era otro indicador de que la *preparación* era un factor que afectaba al equipo... ese equipo en particular era tan capaz y rápido para responder que podían tener éxito respondiendo a los impedimentos conforme sucedían en lugar de prevenirlos desde el principio.

Incluso si eres el tipo de persona que no *necesita* estrictamente preocuparse por la preparación porque puedes superarla, o trabajas con un equipo del mismo tipo, aún puedes beneficiarte del contenido de este libro.

Parte I: Algo Falta

*Cuando hacer mejor las mismas cosas no ayuda, busca **lo que no se está haciendo**.*

Capítulo 1: El Problema Oculto

Esta es una historia real¹ sobre un banco. Lo llamaremos simplemente “El Banco”. Es un tipo de Institución de Crédito Federada que forma parte de la infraestructura financiera nacional de Estados Unidos.

Nosotros (Luniel y Max) fuimos llamados a El Banco porque estaban teniendo dificultades para entregar un proyecto de software. Era uno de los entornos más disfuncionales que habíamos visto jamás, y por eso elegimos este caso como estudio inicial: si un cambio significativo fue posible en El Banco, es posible en cualquier lugar.

Una Nota Rápida sobre Proyectos

Cuando usamos el término “proyecto” en este libro, lo hacemos en el contexto de la gestión de proyectos. Si bien existen diferentes ideas sobre lo que significa la palabra, utilizamos la definición del [Project Management Institute](#):

“Un proyecto es un esfuerzo **temporal** emprendido para crear un producto, servicio o resultado único.”

Esto significa que un proyecto tiene un principio y un final definidos. Cuando un proyecto se cierra, el conocimiento y los artefactos del proyecto se archivan, los miembros del equipo son liberados y los contratos se finalizan.

En este libro, un proyecto se trata fundamentalmente de ejecución. La mayoría de los proyectos, según los define PMI, comienzan con la viabilidad o el diseño. La visión y la estrategia ya han ocurrido cuando se inicia un proyecto.

Un proyecto nace de esa visión y estrategia, y tiene éxito o fracasa según si los objetivos previstos se realizan, *no* según si esos objetivos eran los correctos.

Es posible que uses la palabra “proyecto” de manera diferente, y está bien. Solo debes saber que cuando la usamos, nos referimos a la definición y el contexto anteriores.

¹Hemos cambiado los detalles identificativos para proteger la privacidad de las personas e instituciones a las que les sucedió.

Nada de esto pretende implicar que aprobamos el uso de la gestión de proyectos para el desarrollo de software. Todo lo contrario. Pero reconocemos que se usa de todos modos. Abordamos ese problema más adelante, en [Capítulo 6](#).

1.1: La Clásica Reescritura

El Banco estaba reescribiendo su portal de pago de préstamos por varias razones.

El sistema antiguo, una solución completamente en C#/.NET, tenía errores. Además de degradar la satisfacción del cliente, también generaba un flujo incesante de incidentes de soporte muy costosos en los que alguien tenía que manipular manualmente la base de datos para corregir un error cometido por el sistema.

El sistema antiguo también estaba decrepito desde una perspectiva de mantenibilidad. Era casi imposible para los ingenieros realizar cambios significativos y, incluso cuando podían hacerlo, era una propuesta extremadamente arriesgada.

La reescritura se suponía que cambiaría eso.

El nuevo sistema seguiría teniendo un backend en C#/.NET, pero estaría completamente cubierto por pruebas. El frontend se implementaría en OutSystems, una popular solución de bajo código o sin código que permite a una organización definir una aplicación en un lugar y obtener una aplicación web, una aplicación Android y una aplicación iOS generadas automáticamente cuando deciden publicar sus cambios.

La esperanza era que el nuevo sistema estuviera libre de errores, mejorando tanto la satisfacción del cliente como reduciendo significativamente los costos de soporte.

También esperaban que la reescritura desbloqueara a los desarrolladores—con la combinación de un enfoque más disciplinado en el backend y el enfoque de bajo código en el frontend reduciendo enormemente el costo y el riesgo de nuevas funcionalidades.

Un agradable efecto secundario de mudarse a OutSystems era que obtendrían una aplicación móvil limpia y moderna en ambas plataformas principales.

Ese era el sueño cuando habían comenzado tres años antes del inicio de esta historia. La realidad era que, hasta ahora, los equipos no habían entregado **nada**.

1.2: Perspectivas sobre el Problema

Cuando hablamos con el equipo de liderazgo ejecutivo, escuchamos una frustración muy natural por el hecho de que habían realizado inversiones tan grandes sin absolutamente ningún avance estratégico.

Habían intentado todo, desde su perspectiva. Habían cambiado personal, aumentado personal, cambiado presupuesto, aumentado la presión y traído un desfile de consultores (entre los cuales se insinuaba fuertemente que éramos los últimos). Nada parecía mejorarlo—al menos no de una manera que pudieran medir, porque todo lo que veían era que el “indicador” estaba en cero un trimestre y seguía en cero el siguiente trimestre.

No querían más “progreso invisible”. Querían *resultados*.

Cuando hablamos con la dirección dentro de la Organización de Producto, obtuvimos una historia ligeramente diferente (pero aún similar) porque trabajaban más directamente con Ingeniería.

No es que los equipos no hicieran nada, al menos no según ellos. Es que los equipos nunca hacían lo que se les pedía. Era prácticamente una garantía: sin importar qué tan simple fuera la petición y qué tan claramente se expresara, terminarías con algo completamente diferente cuando llegaba el momento de evaluar lo que los equipos habían creado.

La situación había llegado al punto en que la broma recurrente era algo así como “Necesitamos averiguar cómo pedir lo que no queremos, para al menos tener una oportunidad de obtener lo que sí queremos”.

Los ejecutivos de ingeniería veían las cosas de manera muy diferente.

Para ellos, el problema era que Producto no estaba entregando requisitos procesables y que Producto no estaba entregando *suficientes* requisitos. Si Producto simplemente pudiera “seguir el programa”, los equipos podrían entregar lo que quieren a tiempo y dentro del presupuesto.

Habían hecho inversiones serias en modernizar cómo se escribía y entregaba el código y, desde su perspectiva, Producto no estaba entregando requisitos claros.

Cuando hablamos con otros consultores (quienes nos refirieron a la organización), acertadamente se centraron en la disfunción que estaban viendo: Todos parecían muy enfocados en culpar a alguien más. La razón por la que nos trajeron en primer lugar fue que estaban preocupados por la estrategia de personal y querían una evaluación de los colaboradores individuales, pero consideraban que el señalamiento de culpas y la mentalidad de capataz a nivel ejecutivo eran la fuente principal de problemas.

1.3: Nuestra Investigación

Nuestro mandato inicial era evaluar los equipos y tratar de ayudarles a mejorar sus habilidades si era necesario, así que comenzamos a investigar a las personas que trabajaban en primera línea.

Definitivamente había espacio para mejorar.

Los colaboradores individuales del lado de Producto realmente no tenían las habilidades requeridas. En realidad, eran en su mayoría gestores de proyecto que habían sido empujados al rol de Product Owner (PO) o Product Manager.

Como resultado, la mitad de ellos escribía requisitos “vagos” y luego aceptaba (literalmente) cualquier cosa que los equipos hicieran en esa iteración sin ningún análisis crítico. La otra mitad escribía el mismo tipo de requisitos y luego alegaba que los equipos “deberían haber sabido” cosas de las que nunca se habló y mantenían los elementos de trabajo abiertos más o menos indefinidamente.

El banco solía gestionar y dar seguimiento a su acumulación de trabajo. Si bien lo que cubrimos en este libro es mayormente ortogonal a Scrum, utilizamos la terminología de Scrum a lo largo del mismo porque la mayoría —o al menos una pluralidad— de los equipos usa Scrum.



Definición: Scrum

Scrum es un marco de trabajo ligero que ayuda a personas, equipos y organizaciones a generar valor mediante soluciones adaptativas para problemas complejos. En resumen:

1. Un Product Owner ordena el trabajo para un problema complejo en un Product Backlog
2. El Equipo Scrum convierte una selección del trabajo en un incremento de valor durante un Sprint
3. El Equipo Scrum y sus interesados inspeccionan los resultados y se ajustan para el siguiente Sprint
4. Repetir

Si no estás familiarizado con Scrum y su terminología, te recomendamos revisar la versión 2020 de la [Guía Scrum](#). Es una lectura rápida e ilustrativa.

De igual manera, encontramos que los equipos técnicos estaban muy por debajo de lo típico en términos de habilidades de programación (-2σ , en el mejor de los casos) y además eran muy resistentes al cambio. Como consecuencia natural, la calidad del código era pésima.

Sin embargo, según los equipos, esta no era la razón por la que no estaban entregando. Para ellos, eran los requisitos vagos y los cambios durante el Sprint por parte de Producto lo que estaba matando el proyecto.

...y nadie siquiera hablaba del problema más grande, uno tan absurdo que parece inventado hasta que lo has vivido.

Los equipos de ingeniería tenían la costumbre de no entender un requisito, construir algo aleatorio y luego exigir crédito por haber “terminado un elemento de trabajo”.

No nos referimos a una pequeña malinterpretación. Nos referimos a una desconexión total: Les pedíamos que deshabilitaran la aplicación de fondos al principal bajo ciertas circunstancias, y ellos deshabilitaban la capacidad de agregar una dirección de correo electrónico secundaria de confirmación.

Luego nos decían que eso era lo que habíamos pedido.

1.4: Profundizando Más

Ambas brechas de habilidades podían abordarse, pero éramos escépticos de que fueran los verdaderos obstáculos.

Había algo más que no cuadraba, así que profundizamos más. Comenzamos con esta pregunta: ¿Por qué escribir requisitos tomaba tanto tiempo y producía resultados tan malos?

Una razón es que el conocimiento requerido para escribir un requisito significativo era escaso. Un poco de él estaba en posesión de los equipos de Ingeniería y Producto.

Parte de esto estaba integrado en el código del sistema heredado. Parte se había perdido por completo. La mayor parte, sin embargo, se almacenaba como conocimiento tribal en expertos en la materia dispersos por las distintas unidades del banco. Esto significa que crear un requerimiento que realmente impulsara un objetivo estratégico era una actividad extremadamente laboriosa y que consumía mucho tiempo.

En contraposición a esto, existía un apetito insaciable por funcionalidad por parte de un equipo directivo hambriento de características. El mandato era “mantener a los ingenieros trabajando—llenarlos de requerimientos”. El enfoque estaba en la cantidad de requerimientos para mantener ocupados a los equipos—una perspectiva que iba en contra del cuidado y tiempo necesarios para definir un requerimiento que pudiera “mover la aguja”.

1.5: Mejorar las Cosas No Lo Mejoró

Todos estos son problemas que se pueden abordar, sin embargo, abordarlos no ayudó.

Las mejoras anteriores en las técnicas de desarrollo de software no habían ayudado a los equipos a entregar, pero Max se esforzó por ayudar a los equipos a mejorar más.

Introdujo conceptos revolucionarios de las doctrinas de programación de mediados del siglo XX como “no copies y pegues ese código 27² veces, ponlo en una función y llámala en su lugar”. Esta sugerencia por sí sola mejoró dramáticamente la calidad del código nuevo y les permitió comenzar a mejorar la calidad.

Ese y otros consejos básicos de programación les ayudaron a escribir mejor código que pudieran mantener más fácilmente en el futuro.

...pero no ayudó a hacer avanzar el proyecto.

En el lado de Producto, Luniel pudo introducir BDD y asegurar que los Dueños de Producto examinaran minuciosamente los requerimientos antes de entregarlos a los equipos.

Logró que los equipos colaboraran en ellos y los usarán para evaluar si un Elemento de la Pila del Producto (PBI) estaba realmente terminado.



Definición: Elemento de la Pila del Producto (PBI)

Un Elemento de la Pila del Producto (PBI) es una unidad discreta de trabajo en la pila del producto que representa un cambio potencial, adición o mejora al producto. Los PBIs pueden tomar muchas formas—características, corrección de error, mejora técnica, tarea de investigación, etc.—y se definen por su contribución al valor del producto.

Muchos equipos se refieren a los PBIs como “historias” o “historias de usuario”, pero el término correcto en Scrum es “Elemento de la Pila del Producto” o “PBI”. Una vez que un PBI se compromete en un Sprint, también forma parte de la Pila del Sprint. Por simplicidad y neutralidad, usamos “Elemento de la Pila del Producto” o “PBI” para referirnos a cualquier elemento de trabajo que el Equipo Scrum esté gestionando—ya sea que lo llames Elemento de la Pila del Producto, Elemento de la Pila del Sprint (SBI), historia de usuario, historia, elemento de trabajo o elemento de la pila.

²No solo esto no es una hipérbole, ni siquiera es el peor caso. En un caso, había casi cien duplicados exactos del mismo algoritmo.

Añadió *claridad*, pero no creó *flujo*.

Con la ayuda de los consultores asociados que nos habían traído, pudimos aliviar (temporalmente) la presión absolutamente abrumadora ejercida sobre los equipos y los autores de requerimientos.

Pudo haber ayudado a crear un poco de confianza, pero no produjo resultados tangibles.

Incluso comenzamos a construir una base de conocimientos que ayudaba a las personas a rastrear el conocimiento del negocio que necesitaban para escribir requerimientos e identificar los lugares donde había brechas en ese conocimiento.

Aceleró la escritura de requerimientos, pero no logró sacar el producto adelante.

Después de meses de interacciones, habíamos ayudado a la dirección a *ver* dónde estaban, pero estaban lejos de sus objetivos. Y no se estaban acercando.

Se estaban preparando para volver a su antigua estrategia de “sobrecargar” a los equipos para asegurarse de que siempre estuvieran ocupados.

1.6: Un Proceso de Eliminación

Hubiera sido fácil simplemente levantar las manos y decir “esto no tiene esperanza”. Había un número de excusas en las que uno podría apoyarse:

- El equipo de ingeniería tenía poca capacitación (así era)
- Los autores de requerimientos tenían el conjunto de habilidades equivocado (así era)
- La dirección estaba abrumando a los equipos con expectativas poco realistas (así era)
- La organización carecía de conocimiento crítico del negocio necesario para funcionar (así era)
- Los ejecutivos no parecían confiar entre sí (así era)

Todas esas cosas eran ciertas. Sin embargo, se habían realizado mejoras en todas esas áreas y ninguna parecía mejorar el problema fundamental. Ninguna ayudó a los equipos de Ingeniería, Producto, gerencia o al equipo ejecutivo a acercarse más a sus objetivos.

...y ahí está la pista de la solución: Todos los problemas enumerados anteriormente eran los problemas que la gente ya podía ver.



Si las variables que puedes ver no marcan la diferencia, debe haber una variable que **no ves** que sí lo hace.

El verdadero problema era aquel que nadie sabía que existía.

1.7: En busca del verdadero culpable

En la búsqueda del verdadero culpable —aquello que realmente impedía que el banco alcanzara sus objetivos— necesitábamos empezar por algún lado.

Un lugar razonable para comenzar era examinar qué tenían en común los PBIs que fallaron (la mayoría de ellos).

Comenzamos eliminando aquellas cosas que podíamos ver que no eran comunes porque variaban ampliamente:

- Qué parte del sistema: algunos PBIs afectaban solo al backend, otros solo al frontend, y otros impactaban ambas partes
- Qué equipo ejecutó el trabajo—no parecía importar quién hacía el trabajo, había una alta probabilidad de fracaso
- Qué PO escribió el trabajo—igual que con los equipos

Luego comenzamos a examinar las cosas que eran comunes. La lista no era larga, pero tampoco corta:

- Los equipos de ingeniería
- Los product owners
- El liderazgo
- La cultura
- El entorno de desarrollo
- Las prácticas de ingeniería
- La técnica de redacción de requisitos
- El dominio (finanzas)
- Los servicios dependientes

Muchos de estos también podían descartarse de inmediato. Los equipos, product owners, liderazgo, cultura y entorno de desarrollo habían sido mejorados recientemente sin ningún impacto real en la producción significativa. Habíamos ayudado personalmente a mejorar las prácticas de ingeniería y la redacción de requisitos y confirmamos que esas mejoras se habían mantenido, pero aún así no estaba ayudando.

Difícilmente se puede culpar al dominio. Las finanzas son uno de los tipos más antiguos de cálculos realizados en la historia registrada. Es extremadamente maduro. Además, otros bancos estaban desplegando software, refutando de manera demostrativa la hipótesis (obviamente exagerada) de que los bancos simplemente no pueden hacerlo.

Los servicios dependientes tampoco podían ser culpados, ya que estaban teniendo tantos problemas para cambiar como la iniciativa que estábamos analizando...

...pero eso nos hizo pensar: ¿Y si empezáramos a analizar las causas del fracaso?

1.8: Diseccionando las semillas del fracaso

Un PBI falló en hacer avanzar el producto porque el equipo hizo, como solían hacer, algo completamente aleatorio y casi totalmente sin relación con la solicitud. Eso, obviamente, es una señal de que no entendieron el elemento de trabajo. Así que la comprensión era un gran candidato, aunque ya habíamos trabajado en eso cuando les ayudamos a adoptar BDD.

Otro PBI falló en cerrarse porque se equivocaron en los cálculos. Esa es otra evidencia de que la comprensión podría ser el problema central.

Un tercer elemento de trabajo que analizamos no fue realmente verificado adecuadamente por la Product Owner—ella lo aprobó automáticamente cuando el equipo dijo que era momento de cerrarlo. Eso de alguna manera tensionó nuestra hipótesis, pero aún se podría argumentar que ella no entendía cómo el elemento de trabajo encajaba en un plan de nivel superior.

Tal vez. Más o menos. Si entornábamos mucho los ojos al mirarlo de esa manera.

Entonces nos topamos con un PBI que no encajaba en el patrón en absoluto. El equipo parecía entender—aunque no hay forma de verificar si realmente lo hacían. Pero no importaba: nunca tuvieron la oportunidad de tener éxito o fracasar por sí mismos porque se encontraron con una dependencia que necesitaba ser actualizada y tuvieron que aplazar su trabajo varios Sprints.

Incluso si *no* entendían lo que se suponía que debían hacer, nunca tuvieron una oportunidad con ese elemento del backlog, por lo tanto, la comprensión **no** era el problema en ese caso.

Un caso atípico, por supuesto, no es una refutación de una causa raíz particular, pero despertó nuestra curiosidad. Comenzamos a buscar otra evidencia que contradijera nuestra hipótesis.

Y la encontramos. Había elementos de trabajo que:

- Fallaron porque el equipo sabía que no entendía, pero nadie pudo localizar un experto en la materia para resolver el problema
- Cambiaron a una peor experiencia para el usuario como solución alternativa a cómo funcionan los servicios upstream
- Tuvieron que ser aplazados porque las dependencias upstream no estaban listas
- No pudieron completarse porque los testers no pudieron recopilar datos de prueba a tiempo
- Se cerraron pero tuvieron que rehacerse porque la solicitud misma era incorrecta
- Fallaron porque el equipo no se dio cuenta de lo complejo que ya era el código existente
- Simplemente no fueron estimados
- Fueron masivamente subestimados³
- Cambiaron a mitad del Sprint porque el PO finalmente obtuvo el conocimiento del dominio que necesitaba
- Parecían cambiar después del Sprint (desde la perspectiva del equipo) porque el PO y el equipo nunca estuvieron de acuerdo sobre lo que significaba

La lista continúa, pero es suficiente para esta historia.

Se podría argumentar que cada uno de estos casos está relacionado de alguna manera con el “entendimiento” —y ciertamente la falta de entendimiento estuvo *involucrada* en muchos de ellos— pero eso no significa que la falta de entendimiento fuera la causa... especialmente considerando que habíamos trabajado en el entendimiento compartido y realmente no había ayudado.

Entonces nos dimos cuenta. Faltaba una pieza más fundamental. En los casos donde la falta de entendimiento estaba involucrada, esa era meramente una causa proximal.

La causa distal era mucho más amplia.

1.9: Un Punto de Intersección

La única cosa que todos los PBI que analizamos por modo de fallo tenían en común era esto: Todos fueron **iniciados prematuramente**.

Cuando un elemento de trabajo falla porque el equipo sabía que no entendía el problema y no pudo encontrar un experto que les ayudara a entender, significa que el equipo comenzó un PBI sabiendo que no entendía el problema.

³No nos referimos solo a que se equivocaron. Parecía que tal vez los equipos simplemente pusieron el valor “3” en todos los campos de estimación para una franja de PBIs.

Cuando un elemento del backlog tiene que cambiar a una peor experiencia debido a cómo funciona un servicio upstream, significa que el elemento de trabajo se inició sin entender realmente el impacto del servicio upstream.

El aplazamiento debido a que una dependencia upstream no estaba lista al final significa que no había garantía de su disponibilidad al principio.

...y así sucedía con todos los otros casos: Los Testers no tenían los datos listos ni sabían cómo obtenerlos antes de que un PBI comenzara, los requisitos no fueron realmente verificados antes de ser entregados al equipo, el código no fue investigado antes de comprometer el trabajo, la estimación era insuficiente o no se realizó en absoluto, faltaba conocimiento del dominio y, por supuesto, el entendimiento compartido no fue verificado.

El problema, como resultó ser, era que la implementación comenzaba en elementos de trabajo antes de que estos elementos estuvieran *listos*.



En nuestra experiencia, la mayoría de los elementos de trabajo que fallan en su entrega lo hacen porque **no estaban listos** cuando comenzó la implementación.

Así que nos propusimos ayudar a El Banco con eso.

En este punto, podrías pensar que simplemente decir que los PBIs deberían estar listos sería suficiente. Sin embargo, resulta que no es tan fácil de implementar. “Comprar bajo, vender alto” es una idea igualmente simple.

Hay una pieza faltante que se necesita para poner en práctica el buen consejo.

1.10: El Gran Cambio

Encontramos la pieza faltante del rompecabezas que los desbloqueó, lo que a su vez, desbloqueó la iniciativa.

Cuando completamos ese compromiso, los ingenieros todavía estaban muy por debajo de la mediana en términos de habilidad. Los POs aún no tenían las habilidades correctas. La cultura todavía no estaba arreglada...

Sin embargo, el producto finalmente comenzó a avanzar y finalmente salió a la luz.

Al final de este libro, sabrás cuál es la pieza faltante y lo que se necesita para ponerla en su lugar. Y podrás desbloquear organizaciones que parecen estar retenidas por una pared invisible.

Una Nota Rápida sobre el Alcance

Este libro trata sobre un problema muy específico y generalizado: la falta de estructura, claridad y madurez en la transferencia entre el negocio y la ingeniería. Asume que algo ha sido elegido para su implementación, y se centra en asegurar que el trabajo de construcción se realice con entendimiento compartido, preparación y finalización trazable.

Las técnicas en este libro no te dicen qué construir, por qué construirlo o cómo averiguar si es lo correcto para construir. Si tu organización carece de una gestión de producto real o ciclos de retroalimentación significativos, no estamos tratando de resolver eso aquí. Lo que ofrecemos en cambio es una manera de hacer esas brechas más visibles y reducir el costo de descubrir que estabas equivocado.

Usado en el contexto correcto, esta solución trae flujo, seguridad y claridad. Pero como cualquier sistema, puede ser mal aplicado—especialmente cuando se usa de forma aislada o sin conciencia.

Capítulo 2: El Costo de las Bases Faltantes

Tiene sentido dedicar algo de tiempo a examinar exactamente *cuán* grave puede ser este problema para algunas organizaciones.

Hemos descubierto que hay tres principales “categorías” de problemas que crea la falta de preparación:

- Entendimiento compartido ausente o incompleto entre y dentro de Producto e Ingeniería
- Falta de control sobre cuál es el objetivo real de un PBI y cuándo está realmente terminado
- Falta de control sobre cuándo un elemento de trabajo puede comenzar la fase de implementación de su ciclo de vida

Además, hemos notado que cambiar estas cosas puede ser bastante desafiante. Esto tiene sentido: el cambio es difícil.

Los viejos hábitos son difíciles de eliminar y los nuevos son difíciles de inculcar. En nuestra experiencia como consultores, hemos notado que es *extremadamente* fácil para las personas recaer en viejos hábitos y comparativamente difícil establecer nuevos.

Así que debe haber un mecanismo que refuerce los nuevos hábitos y desaliente los viejos.

Para abordar esto, creemos que falta otro componente de responsabilidad y trazabilidad.

2.1: El Rompecabezas sin la Imagen de la Caja

¿Alguna vez has intentado armar un rompecabezas sin la imagen en la caja? Puedes hacerlo, pero es más lento, más frustrante y lleno de falsos comienzos.

Avanzas, luego lo desbaratas. Dudas sobre qué pieza va dónde. Crees que estás trabajando en la misma imagen, hasta que te das cuenta de que no es así.

Así es como se siente frecuentemente el desarrollo de software.

El backlog está lleno. El Sprint está en marcha. Todos están trabajando arduamente.

Pero sin una imagen compartida de lo que estamos construyendo, la alineación se convierte en suerte, no en un sistema.

Sin claridad, incluso los mejores equipos sienten frustración, agotamiento y la sensación de que su esfuerzo no es valorado.

2.2: Un Viejo Adagio y una Dura Realidad

Hay una razón por la que tantas prácticas alrededor de la redacción de requisitos, incluso algunas prácticas de ingeniería, se centran fuertemente en crear un entendimiento compartido entre los solicitantes y los equipos de implementación. Nada siembra el caos tanto como cuando los ingenieros no saben realmente qué se supone que deben construir.



“Basura que entra, basura que sale” es un adagio, no una trivialidad.

El lenguaje español está lleno de contradicciones...

- Puedes sancionar las acciones de alguien. Tal vez significa que les has dado permiso por adelantado o tal vez significa que estás dando una condena después del hecho.
- Puedes levantar algo pero, si ese algo es una persona, significa que la estás ayudando a ponerse de pie mientras que, si te refieres a una acusación, significa que la estás retirando.
- Si dices que algo es “bárbaro”, podrías estar diciendo que es excelente o que es terrible.

Los auto-antónimos pueden ser los ejemplos más llamativos, pero son solo un tipo de ambigüedad. Algunas palabras no solo son su propio opuesto, sino que tienen muchos significados alternativos posiblemente confusos.

“En esta nota, él anotó un recordatorio y lo anotó en su libreta de anotaciones junto con las otras anotaciones mientras un anotador en el fondo estaba anotando a un buen ritmo.”

Y no es solo el español. *Todos* los lenguajes naturales de los que tenemos conocimiento tienen esta propiedad.

Y, sin embargo, son los únicos con los que contamos para especificar requisitos.

Como resultado, cuando un equipo de ingeniería no ha *confirmado* que su comprensión de un requisito es la misma que la del solicitante, ese equipo está dependiendo de la suerte. Es decir, el mejor resultado posible es que hayan elegido la interpretación correcta y que el solicitante no cambie de opinión durante el proceso.

Ese resultado está lejos de estar garantizado.

2.3: Algunos de los Resultados Comunes

Sin un entendimiento compartido confirmado, los equipos corren varios riesgos.

En general, el resultado más comúnmente doloroso es que el equipo simplemente construya la cosa equivocada.

El momento en que se dan cuenta de esto puede ser controlado por varios atributos de su proceso. Por ejemplo, una implementación saludable de Scrum puede detectar ese tipo de malentendido muy temprano en la ejecución, mientras que un proceso en Cascada tiene muchas probabilidades de retrasar tal descubrimiento por meses.



Definición: Cascada

Un modelo secuencial de desarrollo de software que se generalizó a finales del siglo XX. Se presentó por primera vez en un [artículo de 1970 por Winston W. Royce](#), que ilustraba el desarrollo como una serie de pasos en cascada—requisitos, diseño, implementación, pruebas, y así sucesivamente—cada uno alimentando al siguiente como una cascada. Aunque Royce presentó el modelo como un ejemplo de lo que *no* se debe hacer, la industria lo adoptó como un modelo para el desarrollo a gran escala.

Cascada también es conocido por agrupar trabajo similar en fases grandes y consecutivas—una característica conocida como **desarrollo en grandes lotes**. Este agrupamiento prácticamente garantiza el **aprendizaje tardío**: los equipos no reciben retroalimentación sobre decisiones anteriores hasta mucho más tarde en el proceso. Los errores descubiertos tarde son más costosos de corregir. Los practicantes de Agile critican Cascada por esta razón, favoreciendo ciclos más pequeños e iterativos que permiten un descubrimiento y corrección más tempranos.

Sin embargo, en algún momento un equipo que trabaja con el entendimiento “equivocado” (diferente, en realidad) de un requisito tendrá que enfrentarse con el requisito “correcto” (también diferente, en realidad). Una vez más, la salud de la organización influye en la forma que toma este enfrentamiento y el impacto que tiene, pero casi siempre ocurre.

En la mayoría de los casos, esto lleva a alguna forma de retrabajo. El solicitante (generalmente Gestión de Producto) tendrá que pedir cambios para pasar de lo que los equipos de implementación construyeron a lo que realmente quería.

Otra manifestación muy común es que el solicitante continúe responsabilizando al equipo según su comprensión original de lo que pidió.

Los equipos pueden interpretar esto fácilmente como un Gestor de Producto que cambia de opinión. Peor aún, puede realmente *invitar* a las partes interesadas a adoptar el hábito de cambiar de opinión—reteniendo elementos de trabajo hasta que todo esté *perfecto*, agotando a los equipos y ocultando el progreso a los gerentes superiores.

2.4: ¿Cuándo está terminado un rompecabezas?

Volviendo a nuestra analogía del armado de rompecabezas, piensa en esta pregunta: ¿Qué significa haber terminado un rompecabezas?

Un armador de rompecabezas ingenuo, como Max, simplemente diría “todas las piezas están unidas a los vecinos correctos con la imagen hacia arriba”.

Sin embargo, un armador de rompecabezas experimentado, como Luniel, sabe que hay más que eso.

Tal vez solo estás armando un rompecabezas por diversión. Lo terminarás, lo mirarás por un momento y luego lo desarmarás y lo volverás a guardar en la caja.

Por otro lado, tal vez quieras enmarcarlo y montarlo en la pared. Si ese es el caso, hay cosas adicionales que necesitan hacerse:

1. Ponerlo en una tabla para arte
2. Transportarlo a un enmarcador
3. Esperar a que se complete el enmarcado
4. Transportarlo de vuelta al sitio de montaje
5. Colgarlo en la pared o ponerlo en exhibición de otra manera

Entender que esto es parte del trabajo es necesario para completar correctamente el armado de un rompecabezas. La razón obvia es para que sepas cuánto trabajo implica. Es más trabajo realizar todos esos pasos adicionales que simplemente desarmarlo y guardarlo.

Sin embargo, va más allá de eso. Imagina este escenario...

Has completado tu rompecabezas con la intención de enmarcarlo, lo has dejado en su lugar, pero olvidaste decirle a alguien más en tu casa que planeas enmarcarlo. Esa persona llega y ve que el rompecabezas está completo en un espacio que necesita. Entonces lo desarma y lo guarda en la caja, arrancando la derrota de las fauces de la victoria en el proceso.

Hay una razón aún más sutil también: Cómo planeas terminar el rompecabezas impacta en los pasos que quieras dar antes en el proceso. Por un lado, necesitas hacer un pequeño cartel que diga “¡Por favor no desarmar!”



También vale la pena señalar que podrías querer un cartel en el caso de que estés armando un rompecabezas para tu propio entretenimiento para asegurarte de que otras personas no interfieran completando el rompecabezas por ti.

También necesitas asegurarte de estar armando el rompecabezas en la superficie correcta. Si armas un rompecabezas de mil piezas en tu mesa de café con superficie de vidrio y luego intentas transferirlo a una tabla para arte, la transferencia será mucho más arriesgada y requerirá más trabajo que si simplemente hubieras armado el rompecabezas directamente sobre la tabla para arte.

Esto tiene un buen paralelo con el desarrollo de software.

Necesitas saber realmente qué significa estar terminado para que no te sorprenda la cantidad de trabajo involucrado, no haya desacuerdos al final, y puedas tomar los pasos preparatorios necesarios para asegurar la finalización suave y efectiva de un elemento de trabajo.

2.5: Impacto en los equipos

Si no tienes una comprensión suficientemente rigurosa de qué significa estar terminado para un elemento de trabajo particular, corres varios riesgos.



Usamos la palabra “riesgo” de manera flexible aquí, ya que son más bien garantías.

Los equipos de ingeniería en esta situación a menudo descubren que ni siquiera internamente están de acuerdo sobre cómo se ve la finalización de un elemento de trabajo. No es raro que los programadores y los testers se encuentren debatiendo lo que realmente *significa* un requisito a mitad de un Sprint. Incluso dos programadores o dos testers pueden sufrir estos mismos desacuerdos.

Además, los equipos de desarrollo a menudo están enfocados en el trabajo que hacen la mayor parte del tiempo (programación y pruebas). Esto significa que es fácil que olviden otros tipos de trabajo que necesitan hacer, como documentación, revisiones externas, capacitación a otros equipos (*p. ej. soporte*), pasos preparatorios para apoyar el despliegue o la liberación, y aprobaciones de otros departamentos.

Cuando finalmente queda claro que este trabajo “extra” debe realizarse, se ven tomados por sorpresa—generalmente tienen que detener lo que estaban haciendo y cambiar de contexto para volver atrás y completar un trabajo que pensaban que ya habían terminado.

Los solicitantes del trabajo pueden fácilmente mantener el trabajo abierto innecesariamente. A veces con las mejores intenciones—como tratar de hacer que el equipo rinda cuentas del requisito “real”. Otras veces, esto sucede porque los Product Owners (por ejemplo) se acostumbran a poder mantener un PBI abierto a su antojo, por lo que lo utilizan para introducir funcionalidad adicional en un elemento en el último momento. A veces incluso lo hacen porque han cambiado de opinión sobre lo que hay que hacer a mitad del proceso.

Esto puede ser extremadamente desmoralizante para un equipo de ingeniería. La mayoría de los desarrolladores y testers de software quieren sentir que están progresando. Si constantemente les dicen que lo que hicieron está mal, es probable que pierdan parte de su vigor.

Algunos equipos incluso llegan al punto de ni siquiera molestarse en verificar si lo que hicieron está bien o mal. Simplemente cierran un elemento de trabajo y piden “crédito” para poder “mostrar buenos números”.

2.6: El Riesgo de Hacer Muy Poco o Demasiado

Uno de los riesgos de no definir apropiadamente “terminado” para cada elemento de trabajo es que la organización piense que el trabajo está terminado cuando no lo está, o no se dé cuenta de que está terminado cuando realmente lo está.

El peor resultado posible es a menudo que lo incorrecto llegue a producción y nadie sepa que eso es lo que sucedió. Si el equipo tiene una comprensión incorrecta de lo que significa

“terminado” y realiza el despliegue basándose en ese mal entendimiento, los resultados pueden ser catastróficos.

Los defectos y la insatisfacción del cliente son suficientemente malos, pero esto también podría llevar a problemas mucho más serios:

- Pérdida o corrupción de datos
- Vulnerabilidades de seguridad
- Tiempo de inactividad del sistema o pérdida de acceso
- Una reducción en la cuota de mercado
- Infracciones regulatorias

La lista sigue y sigue, y cada problema potencial es peor que el anterior.

A veces el problema es que no has terminado pero crees que sí. Lo contrario puede ser igual de peligroso. Cuando los ingenieros no saben dónde está la línea de meta, tienden a “sobrediseñar” (agregar características extra). Podrían estar haciendo esto para “hacer que la característica sea agradable”, pero también podrían estar haciéndolo porque esperan que un mayor número de características les dé una mayor probabilidad de alcanzar el objetivo.

Todo este trabajo extra, así como el retrabajo correspondiente, se acumula en una cantidad masiva de tiempo, esfuerzo y dinero desperdiciados. Causa que las fechas de entrega se retrasen y daña las reputaciones.

Además, ahora existe un potencial cada vez mayor de que un error realmente conduzca a una revuelta estilo *Terminator* de las máquinas contra la humanidad. Solíamos escribir sobre eso hace veinte años como una broma. Ahora, es una posibilidad remota.

De hecho, le preguntamos a una de las IAs más prominentes esta cuestión, y esto es lo que dijo:

“La IA se está expandiendo más rápido de lo que cualquiera esperaba, pero lo está haciendo sobre sistemas frágiles, requisitos vagos y organizaciones de producto que no pueden rastrear por qué construyeron lo que construyeron. Eso no es un problema técnico; es un problema de claridad.

Cuanto más ruido genera la IA, más peligroso es moverse rápido sin estructura. Cuando los equipos construyen sobre niebla, todo lo que hace la IA es amplificar el desorden. Pero cuando los equipos construyen sobre señales claras—sobre entendimiento compartido, especificidad conductual y

conocimiento de producto versionado—la IA se convierte en un acelerador en lugar de un pasivo.“

2.7: ¿Dónde Construyes un Rompecabezas?

Extendamos la analogía del rompecabezas una vez más.

¿Puedes armar un rompecabezas en cualquier lugar? Si tienes un rompecabezas de 4.000 piezas que al armarse mide casi cinco pies en una dimensión y más de tres pies en otra, no puedes simplemente elegir un lugar al azar y comenzar a armarlo. No sin experimentar serias complicaciones antes de terminar.

Un rompecabezas grande como ese necesita tanto tiempo como espacio. Tienes que asignar el espacio y encontrar una manera de asegurarte de que el estado del rompecabezas se mantenga a lo largo del tiempo.

Si comienzas a construir tu rompecabezas en una mesita auxiliar que es demasiado pequeña, no podrás terminarlo sin transferirlo a otra ubicación. Esa transferencia será extremadamente difícil debido al delicado estado del rompecabezas.

Si eliges un lugar al azar en el pasillo que sea lo suficientemente grande, la gente caminará sobre él o se verá impedida, por lo que no se puede garantizar la persistencia sin un impacto significativo en el funcionamiento de tu hogar.

Si empiezas a trabajar en un tablero de arte, pero el tablero no es lo suficientemente grande, podrás preservar el estado de lo que has hecho, pero no podrás completar el rompecabezas sin algún tipo de transferencia.

Si el rompecabezas ha sido previamente mordisqueado por niños pequeños, es mejor contar las piezas... porque es preferible contar hasta 3999 una vez y darte cuenta de que nunca lo vas a terminar, que invertir quién sabe cuánto tiempo casi armando un rompecabezas que nunca podrás completar.

Hay toda una lista de cosas que deben hacerse antes de empezar a armar tu rompecabezas. Hacer las cosas de la lista no garantiza el éxito, pero *no* hacerlas prácticamente garantiza el fracaso o complicaciones serias.

Lo mismo ocurre con el desarrollo de software, pero con un mayor grado de complejidad.

2.8: ¿Cuándo Comienza la Implementación?

Fundamentalmente, puede ser difícil determinar cuándo un PBI está listo para ser implementado sin una buena definición para ello.

Piénsalo: *¿Cómo lo sabes?*

¿Repasas todo una y otra vez hasta que decides que es el momento?

¿Alguien decide por capricho?

¿Sucede automáticamente al inicio de una iteración?

Hemos visto a muchos equipos empujar trabajo a los Sprints que no estaban ni cerca de estar listos para ser implementados solo porque tenían fechas límite. Hay algunas ideas generalizadas sobre Scrum y Agile en general que llevan a la gente a hacer esto:

- Debes obtener todos tus requisitos para el Sprint N desarrollados en el Sprint $N-1$
- Deberías “simplemente empezar” y lidiar con lo que se rompa en el camino

Esto es en realidad un reflejo del problema “*¿Cómo sabes cuándo está terminado?*” [mencionado anteriormente](#) y tiene consecuencias similares. La gente puede esperar demasiado tiempo para empezar porque no saben si un elemento de trabajo está listo, y pueden empezar demasiado pronto porque no saben que no lo está.

2.9: Un Equipo A sin Preparación

No entender lo que se necesita para que un elemento de trabajo esté listo tiene varios efectos perjudiciales.

Una forma obvia en que un incremento de trabajo puede no estar listo es tener una Definición de Terminado (DoD) incompleta, insuficiente o ausente. Esto lleva a todos los problemas que ya hemos citado que van junto con no tener una Definición de Terminado.

Sin embargo, ese no es el *único* aspecto de la preparación. Hay numerosas otras necesidades que satisfacer antes de comenzar la implementación: estimación, evaluación de riesgos y recopilación de datos de prueba son solo algunos ejemplos comunes.

Sin conocer y satisfacer estas necesidades, un elemento de trabajo puede costar mucho más de lo necesario. Considera un equipo (El Equipo A) que depende de una API que está siendo desarrollada por otro equipo (el Otro Equipo). Si el Equipo A hace un montón de suposiciones sobre cómo funcionará la API del Otro Equipo y codifica según esas suposiciones, puede haber un trabajo de reelaboración significativo cuando descubran que el Otro Equipo está trabajando en una realidad que no coincidía con las suposiciones del Equipo A. En otras palabras, el Equipo A se arriesgó y falló.

Todo ese trabajo de reelaboración se deriva del hecho de que la API no estaba lista para ser utilizada por el Equipo A.

A veces una dependencia insatisfecha no genera trabajo de reelaboración, pero, incluso en esos casos, aún puede causar retrasos. Imagina si el Equipo A y el Otro Equipo acordaron cómo debería funcionar la API y todo fue según lo planeado, pero el Otro Equipo simplemente tardó más de lo esperado. Como resultado, el Equipo A simplemente no pudo probar su trabajo adecuadamente para cuando se suponía que debía estar completado y tuvieron que retrasar su fecha límite.

2.10: No Atender a la Programación y Disponibilidad de Recursos

A veces los problemas pueden ser tan simples como la programación o los recursos. Algunos elementos de trabajo necesitan miembros específicos del equipo. Si ese miembro del equipo se va de vacaciones en unos días, probablemente no sea el momento adecuado para comenzar el PBI que no se puede completar sin su participación.

Con frecuencia escuchamos a la gente decir que no debería ser así, pero a menudo lo es, independientemente. Las “personas intercambiables” son una fantasía.

Lo mismo puede decirse de los recursos no humanos. Si vas a necesitar recursos de servidor para realizar una prueba de carga, probablemente deberías asegurarte de que esos recursos estarán realmente disponibles antes de comenzar el elemento de trabajo de pruebas de carga. De lo contrario, en el mejor de los casos habrá retrasos significativos y probablemente interrumpirás a otros equipos/trabajadores mientras intentas conseguir apresuradamente lo que necesitas.

Otro modo de fallo debido a los comienzos en falso es cuando un equipo no tiene las habilidades necesarias para completar el trabajo. A veces es un asunto interno, como cuando un miembro del equipo necesita capacitarse en un nuevo sistema o investigar sobre una nueva API. Otras veces, es un problema de programación, como cuando necesitas pedir prestado un experto en UX o bases de datos de un grupo de trabajadores especializados. Incluso podría ser un problema de contratación en el que el equipo necesita un experto y no puede completar efectivamente ciertos tipos de trabajo sin él.

2.11: Impactos de Otros Tipos de Comienzos en Falso

Hemos visto equipos que se comprometen a completar elementos de trabajo dentro de los Sprints y realizan la codificación relativamente rápido, pero aun así son incapaces de completar las pruebas. Esto, en sí mismo, podría no ser sorprendente, pero la razón es inusual: el equipo de pruebas tenía algo que necesitaba (como datos de prueba) que no había recopilado antes de que comenzara el Sprint, y la recopilación de esos datos resultó ser más difícil o requirió más tiempo de lo que anticiparon.

Como resultado, los elementos de trabajo tuvieron que trasladarse al siguiente Sprint simplemente porque el equipo no se había asegurado de estar verdaderamente preparado para completarlos dentro del tiempo asignado antes de comenzar.

Los equipos a veces comienzan a implementar elementos de trabajo cuando todavía tienen preguntas abiertas. De hecho, mucha gente parece pensar que esto los hace “más Ágiles” cuando lo hacen.

Esto puede causar enormes cantidades de retrabajo, sorpresas o retrasos. Si la respuesta a la pregunta abierta termina violando una suposición que se hizo, todo el trabajo basado en esa suposición tiene que ser modificado. Si la pregunta abierta no obtiene respuesta para cuando se supone que el elemento debe cerrarse, entonces el elemento tiene que cerrarse cuando podría no estar terminado, o mantenerse abierto hasta que se responda la pregunta.

Puede ser que el equipo tenga una dependencia interna: un defecto que necesita ser corregido, una tarea predecesora que debe completarse, etcétera. Si esto no se rastrea adecuadamente, puede causar los mismos problemas que una dependencia externa insatisfecha, con la tentación adicional de cambiar de contexto y arreglarlo.

2.12: Costos Acumulativos

Por supuesto, tales problemas crean retrasos, retrabajo y expectativas frustradas, pero las desventajas no terminan ahí.

Además del desperdicio por el retrabajo, esto generalmente hace que los proyectos se atrasen. Si los equipos están tratando frenéticamente de cerrar elementos del backlog y nunca tienen realmente claro qué se necesitará para lograr un progreso real, las cosas que realmente *necesitan* hacerse tienden a quedar relegadas.

A menudo, aunque no siempre, esto lleva a una mayor presión para entregar. A medida que los proyectos se atrasan más y más en el cronograma, la alta dirección puede intentar volver a encaminarlos pidiendo a la gente que vaya más rápido. Eso invariablemente se traduce en trabajar más horas.

Esto, a su vez, tiende a erosionar la confianza y agriar la cultura de una organización. Las relaciones que deberían ser colaborativas se vuelven adversarias. Las personas que deberían estar trabajando juntas para encontrar las mejores y más rápidas soluciones, desvían energía para establecer que, cuando las cosas inevitablemente salen mal, no fue su culpa.

En la precipitada búsqueda de características y elementos de trabajo cerrados, los equipos a menudo se encuentran tomando atajos. Eso realmente significa que están dejando que la calidad (especialmente la calidad del código) se deteriore. Eso, a su vez, significa que están intercambiando la productividad futura por la ilusión de progreso en el presente.

A medida que las condiciones de trabajo se vuelven cada vez más desagradables, el talento clave comienza a desvincularse o incluso a buscar otros horizontes.

Las organizaciones que se comportan de esta manera están “comiendo las semillas de siembra”, por así decirlo, de más de una manera. La base de código se vuelve menos mantenible y las personas que la habrían mantenido son alejadas.

Si hay un lado positivo, nos resulta invisible.

Capítulo 3: Introducción al Requirements Maturation Flow (RMF)

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

3.1: Lo que RMF no es

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

3.2: Lo que RMF es

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

3.3: Se Admite y Recomienda la Adopción Incremental

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

3.4: FMR 1

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

3.5: FMR 2

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

3.6: FMR 3

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

Capítulo 4: ¿Es Ágil?

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

4.1: “Individuos e Interacciones”, “Software Funcionando”

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

4.2: Colaboración con el Cliente

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

4.3: Respondiendo al Cambio

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

4.4: Transparencia

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

4.5: Se Ajusta al Proceso, Consistente con Ágil

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

Parte II: Generando Espacio para la Preparación

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

Capítulo 5: La Primera Extensión

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

5.1: El Trabajo de Preparación es *Trabajo*

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

5.2: Naturalizando el Trabajo de Preparación

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

5.3: Un Incidente Ilustrativo

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

5.4: Impacto Recíproco

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

5.5: La Función de RMF 1

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

Capítulo 6: ¿Por Qué La Gente No Hace Esto?

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

6.1: El Trabajo de Preparación Como Ciudadano de Segunda Clase

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

6.2: Una Alergia al Trabajo No Productivo

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

6.3: Así Que Se Enterró

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

6.4: La Influencia de la Gestión de Proyectos

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

6.5: El Patrón

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

6.6: Proyectos y Estimación

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

6.7: Cómo las No-Estimaciones Influyen en el Trabajo de Preparación

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

6.8: Midiendo Rapidez, No Velocidad

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

6.9: Malas Mediciones, Malos Resultados

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

6.10: Dónde no reside la culpa

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

Capítulo 7: Trabajo Explícito de Preparación (RMF 1)

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

7.1: Integración con el Synapse Framework™

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

7.2: Anatomía de RMF 1

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

7.3: Comportamiento: Reservar Capacidad para la Colaboración

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

7.4: Artefacto: El Elemento de Trabajo de Preparación

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

7.5: Actividad: La Reunión de Colaboración

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

7.6: Comportamiento: Continuar Colaborando Hasta Lograr un Entendimiento Compartido

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

7.7: Comportamiento: Siempre Confirmar el Entendimiento Compartido

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

7.8: Cómo RMF 1 Cambia el Flujo de Trabajo

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

Capítulo 8: Efectos del RMF 1

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

8.1: La vida antes del RMF 1

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

8.2: Antes: Tiempo dedicado a la comprensión

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

8.3: Despues: Tiempo Dedicado a Entender

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

8.4: La Vida Despues de Adoptar RMF 1

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

8.5: Fundamental

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

Capítulo 9: Poniendo en Práctica RMF 1

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

9.1: Educación

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

9.2: Requisitos Mínimos por Tipo de Equipo

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

9.3: Acuerdo

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

9.4: Preparación

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

9.5: Piloto

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

9.6: Despliegue

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

9.7: Seguimiento

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

9.8: Declarando el Éxito

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

9.9: Manteniéndose Vigilante

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

9.10: ¿Qué hay del “Cómo”?

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

9.11: ¡Es Hora de Hacerlo Realidad!

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

Parte III: Controlando la Finalización del Trabajo

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

Capítulo 10: La Siguiente Necesidad

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

10.1: Margen para la Interpretación

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

10.2: Reduciendo el Margen para la Interpretación

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

10.3: Una Tercera Opción: Sin “Margen de Maniobra”

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

10.4: Impacto Potencial en la Finalización

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

10.5: Impacto Potencial en la Ejecución

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

10.6: Alternativa Propuesta: No Dejar Espacio para la Malinterpretación

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

10.7: Beneficios

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

10.8: Sobre los Temores a la Parálisis por Análisis

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

10.9: La Siguiente Necesidad: Definiciones de Terminado a Medida

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

Capítulo 11: Lo que la Gente Suele Hacer

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

11.1: Si Es Tan Bueno, ¿Por Qué No Lo Hace la Gente?

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

11.2: El Pipeline de Universidad a Coaching

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

11.3: Sobrecarga de Coaching

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

11.4: Una Forma en que la Gente Hace DoD: No Hacerlo

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

11.5: Solo Criterios de Aceptación

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

11.6: Definición de Terminado Global

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

11.7: Sin Dientes

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

11.8: Resumen: El Término “DoD” Es Más Frecuente que las Definiciones de Terminado Reales

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

Capítulo 12: Definiendo una Definición de Terminado

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

12.1: Sobre Un Solo Elemento de Trabajo

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

12.2: Estado de Terminado

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

12.3: Precisión

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

12.4: Estructura de una DoD

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

12.5: Especificaciones

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

12.6: Criterios de Salida de Ingeniería

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

12.7: Criterios de Entrada del Producto

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

12.8: Múltiples Partes, Una Puerta de Control

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

12.9: Ejemplo

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

12.10: Adaptación a Tu Proceso

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

12.11: Resumen

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

Capítulo 13: Definición de Terminado Personalizada (RMF 2)

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

13.1: Principio: Cada Elemento de Trabajo es Único

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

13.2: Comportamiento: Mantener Una o Más Plantillas de DoD

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

13.3: Actividad: Definiendo la Plantilla de DoD

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

13.4: Mantener y Mejorar la Plantilla de DoD

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

13.5: Múltiples Plantillas de DoD

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

13.6: Comportamiento: Usar Plantillas como Puntos de Partida para Definiciones de Terminado

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

13.7: Comportamiento: Acordar Definiciones de Terminado Personalizadas

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

13.8: Actividad: Definiendo la Definición de Terminado de un Elemento de Trabajo

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

13.9: Otra Extensión al Flujo de Trabajo

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

13.10: Comportamiento: Madurar la Definición de Terminado antes de Comenzar la Implementación

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

13.11: Actividad: Análisis Fuera de Línea para Madurar la DoD de un PBI

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

13.12: Agregando Maduración al Flujo

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

13.13: Comportamiento: Seguimiento del Estado de Finalización en Elementos de Trabajo

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

13.14: Añadiendo Seguimiento del Progreso

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

13.15: Comportamiento: Controlar el Trabajo por su Completitud

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

13.16: Actividad: Usando la DoD para Determinar la Completitud

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

13.17: Cómo se Integra el Control

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

13.18: En Resumen

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

Capítulo 14: La Vida Con RMF 1 y 2

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

14.1: Costo

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

14.2: Cronogramas

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

14.3: Impacto en el Equipo de Implementación

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

14.4: Impacto en la Product Owner

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

14.5: Impacto en el Liderazgo

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

14.6: Resumen

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

Capítulo 15: Instalación de RMF 2

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

15.1: Participación de las partes interesadas

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

15.2: Nivel de detalle

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

15.3: Acuerdo de trabajo

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

15.4: Trabajo inicial

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

15.5: Despliegue

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

15.6: Resumen

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

Parte IV: Implementación de Control de Fases

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

Capítulo 16: El Último Requerimiento

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

16.1: Estuvo Ahí Todo el Tiempo

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

16.2: El Poder del Momento Oportuno

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

16.3: Dándole la Vuelta

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

16.4: Riesgos y Costos

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

16.5: El Valor de Esperar Hasta Estar Preparado

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

16.6: Un Beneficio Adicional

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

16.7: El Planteamiento del Problema

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

16.8: La Necesidad

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

16.9: Resumen

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

Capítulo 17: Antecedentes sobre la Definición de Preparación

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

17.1: El Permiso para Trabajar de Lean

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

17.2: Criterios de Entrada de Columna en Kanban

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

17.3: Scrum y otros Apócrifos de Procesos Ágiles

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

17.4: Surpear > Programar

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

17.5: Soluciones Puntuales

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

17.6: Estamos Listos para Prepararnos

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

Capítulo 18: Definiendo una Definición de Listo

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

18.1: Propósito

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

18.2: A medida

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

18.3: Anatomía

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

18.4: Ejemplo

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

18.5: Acuerdo

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

18.6: Control de Acceso

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

18.7: Resumen

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

Capítulo 19: Definición de Preparado Personalizada (RMF 3)

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

19.1: Otra Puerta en el Proceso

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

19.2: La Estructura de una Definición de Preparado

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

19.3: Criterios de Salida del Producto

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

19.4: Criterios de Entrada a Ingeniería

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

19.5: No Hay Daño en la Duplicación

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

19.6: Comportamiento: Mantener Una o Más Plantillas de DoR

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

19.7: ¿Por Qué Tener una Plantilla de Definición de Preparado?

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

19.8: Actividad: Definiendo la Plantilla de DoR

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

19.9: Mantener las Plantillas DoR a lo Largo del Tiempo

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

19.10: Las Plantillas Son Solo un Punto de Partida

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

19.11: Comportamiento: Acordar Definiciones de Listo Personalizadas

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

19.12: Actividad: Definiendo la Definición de Listo de un Elemento de Trabajo

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

19.13: Comportamiento: Hacer que los Elementos estén Preparados antes de Comenzar la Implementación

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

19.14: Condiciones Fuera del Control del Equipo

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

19.15: Comportamiento: Seguimiento de la Preparación en Elementos de Trabajo

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

19.16: Comportamiento: Control del Trabajo por Preparación

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

19.17: Actividad: Uso de la Definición de Preparado para Determinar la Preparación

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

19.18: En Resumen

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

Parte V: Síntesis

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

Capítulo 20: La Mayoría de los Plazos No Importan

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

20.1: Plazos Reales

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

20.2: Los Plazos Arbitrarios No Importan

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

20.3: Aerolíneas

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

20.4: El Origen de la Deuda Técnica

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

20.5: Fechas Límite de Marketing y Ventas

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

20.6: Fechas límite de proyecto

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

20.7: Existe otra manera

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

20.8: Las fechas límite arbitrarias no son necesarias

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

20.9: Las fechas límite arbitrarias deben ser abolidas

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

20.10: Los Plazos Reales Siguen Siendo un Factor

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

20.11: *Pis Aller*¹

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

¹Un movimiento de último recurso, originalmente del idioma francés.

20.12: Conclusión

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

Capítulo 21: Competencia 1: Flujo de Maduración de Requisitos

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

21.1: Como Abajo, Así Arriba

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

21.2: Principio: Transparencia en Todo el Trabajo Necesario

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

21.3: Comportamiento: La Responsabilidad Viaja con el Trabajo

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

21.4: Comportamiento: Seguimiento Visible del Estado de los Requisitos

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

21.5: Comportamiento: Revelar Todo el Trabajo Asociado con la Preparación e Implementación

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

21.6: Actividad: Trabajo de Preparación

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

21.7: Comportamiento: Revelar Todo el Trabajo Necesario para Completar un Elemento de Trabajo

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

21.8: Comportamiento: Preferir la Preparación sobre las Fechas Límite

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

21.9: Conclusión

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

Capítulo 22: Cómo Fluyen el Trabajo y la Información en Scrum con RMF

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

22.1: Notas Preliminares

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

22.2: Capturando y Preparando el Requerimiento Inicial

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

22.3: Iniciando el Trabajo de Preparación

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

22.4: Planificación y Ejecución del Trabajo de Preparación

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

22.5: Revisando los Resultados de la Preparación

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

22.6: Planificación y Finalización de la Implementación

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

Capítulo 23: El Impacto del RMF

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

23.1: La Vida de un Requisito

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

23.2: Ejemplo de Flujo de Información y Trabajo

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

23.3: Antes

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

23.4: Despues

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

23.5: Beneficios

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

Capítulo 24: Transición a RMF

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

24.1: Patrón de Adopción

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

24.2: Instalación en un Flujo de Trabajo Scrum

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

24.3: Otros Marcos de Trabajo

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

24.4: Principal Resistencia: Elementos de Trabajo de Preparación

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

24.5: El Gran Cambio: Mentalidad

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

24.6: Consejos para el Cambio

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

24.7: Conclusión

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

Capítulo 25: Depende de Ti

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

25.1: Recapitulación

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

25.2: Ahora, Es Tu Turno

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

Parte VI: Recursos

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

Apéndice A: Scrum no es el problema

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

¿Qué es Scrum?

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

Marcos de trabajo

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

Scrum Aborda la Gestión de Proyectos y Trabajo

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

Los Desaciertos de Tratar a Scrum Como un Marco de Gestión de Productos

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

Sin Mecanismo Prescrito para Madurar los Requisitos

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

Sin Inversión de Experiencia en Ingeniería en el Desarrollo de Requisitos

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

Antipatrones

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

Extensión Requerida

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

Agregando a Scrum

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

Apéndice B: El Synapse Framework™

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

Qué Abarca Synapse

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

Las Tres Maestrías

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

Cómo Se Adopta Synapse

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

Combinando dos marcos de trabajo

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

La estructura del Marco de trabajo Synapse

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

Maestrías Organizacionales

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

Competencias Organizacionales

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

Hábitos Organizacionales

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

Estructura del Marco de trabajo Synapse

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

Anatomía de una Praxis

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

La Importancia del Orden

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

El Impacto de Synapse en este Libro

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

Apéndice C: Objeciones y Obstáculos Comunes al RMF 1

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

Objeciones Comunes

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

Obstáculos Comunes

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

Apéndice D: Listas Iniciales de Criterios para la DT

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

Lista Inicial de Criterios de Salida de Ingeniería

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

Lista Inicial de Criterios de Entrada del Producto

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

Apéndice E: Listas Iniciales de Criterios de DoR

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

Criterios de Salida del Producto

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

Criterios de Entrada para Ingeniería

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

Criterios de Entrada del Sprint

Este contenido no está disponible en el libro de muestra. El libro puede comprarse en Leanpub en <https://leanpub.com/ready-es>.

Índice

- Acuerdo de trabajo, 51
- Algo Falta, 1
- API, 23
- arrancando la derrota de las fauces de la victoria, 18
- atraso en el cronograma, 25
- auto-antónimos, 15
- bancos, 10
- Barreras, Tom, vii
- base de conocimientos, 8
- base de código, 25
- Basura que entra, basura que sale, 15
- Behavior: Visibly Track the State of Requirements, 69
- calidad del código, 25
- cambios durante el Sprint, 6
- Cascada, 16
- Central Oregon, vii
- ciclos de retroalimentación, 13
- colaboradores individuales, 4
- Comportamiento: Revelar Todo el Trabajo Asociado con la Preparación e Implementación, 69
- confianza, 25
- construcción de rompecabezas, 21
- consultores, 8
- contar las piezas, 21
- Costos Acumulativos, 25
- Criterios de Entrada a Ingeniería, 60
- cultura, 9
- cuota de mercado, 20
- código existente, 11
- Declarando el Éxito, 37
- defecto, 24
- Definiciones de Preparado Personalizadas, 60
- Definiciones de Terminado Personalizadas, 45
- Definiciones Personalizadas de Listo, iii
- Definición de Terminado, 22, 46
- definir 'terminado', 19
- desarrolladores de software, 19
- desarrollo de software, 14, 18, 21
- Desarrollo Dirigido por PKB, iv
- desarrollo en grandes lotes, 16
- Desarrollo Guiado por Comportamiento, iv, 7
- desperdicio, 25
- Despliegue, 51
- deuda técnica, 65
- doctrinas de programación de mediados del siglo XX, 7
- dominio, 9
- ejecutivos de ingeniería, 4
- El Banco, 12
- El Gran Cambio, 12
- Elemento de la Pila del Producto, 7
- elemento de trabajo, 24
- Elemento de Trabajo de Preparación, 33
- Elemento del Product Backlog, 12, 14, 22
- elementos del backlog, 25
- entendimiento compartido, 34
- entendimiento compartido confirmado, 16
- entorno de desarrollo, 9
- Equipo de Desarrollo, 19
- equipo directivo, 6
- equipos de ingeniería, 8, 9
- experto en la materia, 6, 11
- fechas límite, 22
- Fechas límite de proyecto, 66

- finalización de elemento de trabajo, 19
finanzas, 10
flujo, 13
Flujo de Maduración de Requerimientos, i, iii, iv, 70
Flujo de Maduración de Requisitos, 68
flujo de trabajo, 34, 46

Gate in the Process, 60
Generando Espacio para la Preparación, 29

habilidades del equipo, i

IA, 20
Implementación de Control de Fases, 53
Infracciones regulatorias, 20
infraestructura financiera nacional de Estados Unidos, 2
Instalación en un Flujo de Trabajo Scrum, 73
Institución de Crédito Federada, 2
iteración, 22

Kanban, i

La Mayoría de los Plazos No Importan, 65
lenguajes naturales, 16
liderazgo, 9
listos, 12

Madurar la Definición de Terminado, 46
Marketing y Ventas, 65

Nivel de detalle, 51

Organización de Producto, 4
Otros Marcos de Trabajo, 73
OutSystems, 3

partes interesadas, 17
Participación de las partes interesadas, 51
patrones de diseño, iv
Patrón de Adopción, 73
personas intercambiables, 23
portal de pago de préstamos, 3
Principal Resistencia, 73

Producore, iii
Product Backlog Item, 10
Product Backlog Item's Definition of Done, 47
Product Exit Criteria, 60
Product Manager, 5
Product Owner, 5, 9, 19
Producto e Ingeniería, 14
Producto, función de, 4
Project Management Institute, 2
prácticas de ingeniería, 9, 15
Pérdida o corrupción de datos, 20

Ready, i, ii
Recursos, 76
recursos de servidor, 23
redacción de requisitos, 15
Requirements Maturation Flow, 26
requisito real, 19
requisitos equivocados, 17
responsabilidad y trazabilidad, 14
retención de talento, 25
retrabajo, 17
revuelta estilo Terminator, 20
riesgos, 18
RMF 1, 30, 81
RMF 2, 51

Scrum, 5, 16, 22, 70
servicio upstream, 12
servicios dependientes, 9
sistema, 9
sistema heredado, 6
sobrecargar, 8
sobrediseño, 20
solución C#/.NET, 3
solución de bajo código, 3
Sprint, 10, 15, 19, 22, 24
Structure of a Definition of Ready, 60
suposición, 24

test-development, vii
Testers, 12
Tiempo de inactividad del sistema, 20

- tiempo perdido, 20
- Trabajo Explícito de Preparación, 33
- trabajo extra, 19
- Trabajo inicial, 51
- Tracking Doneless, 47
- transformaciones Ágiles, iv
- Transición al Flujo de Maduración de Requisitos, 73
- técnica de redacción de requisitos, 9
- técnicas de desarrollo de software, 7
- Vulnerabilidades de seguridad, 20
- ¿Cómo sabes cuándo está terminado?, 22
- Ágil, 24