
DIPLOMADO
Formación Profesional en Proyectos
DOBLE CERTIFICACIÓN INTERNACIONAL
Project Management Certified
PMC
HumansPRO®
SCRUM MASTER PROFESSIONAL
CERTIFIED
Humans PRO
Universidad de San Buenaventura
THINK DESIGN BUILD ING
FACULTAD DE INGENIERÍA
Humans PRO



Ruta de formación
Especialización

Especialización

Gestión Integral de Proyectos

Código SNIES: 102009

Registro calificado: Res. No.014483. 13 diciembre 2019 - 7 AÑOS.

Humans PRO

Ente de acreditación internacional

BIENVENIDOS



TOMAS LISCANO

Especialista en Gestión de Proyectos
VP Membresías PMI® Pacífico Colombia
Certificada Social Project Management
Instructora HumansPRO® Latam



TOMAS LISCANO

Especialista en Gestión de Proyectos
Certificado PMP-PMI® | ACP-PMI® |
Certified SAFe® 6 Agilist y SM | SM | PO
Instructor HumansPRO® Latam





Diplomado en gestión de proyectos - HumansPRO® 2025



Scrum Master Profesional

Contenido del Curso (1/4)

- **Módulo 1 - Fundamentos de la Agilidad y de Scrum**
  - El pensamiento ágil y el Manifiesto Ágil.
  - Contexto: complejidad, adaptabilidad y empirismo.
  - Diferencias entre la gestión tradicional y la ágil.
  - ¿Qué es Scrum y qué no es Scrum?
  - Los tres pilares de Scrum: transparencia, inspección y adaptación.
- **Módulo 2 – Valores y Cultura Scrum**
  - Compromiso, Foco, Apertura, Respeto y Coraje.
  - Cómo los valores sustentan la autogestión y la transparencia.
  - Conductas esperadas del Scrum Master para fomentar la cultura Scrum.

✨ ***El conocimiento cobra sentido cuando se convierte en acción con propósito.***



Scrum Master Profesional

Contenido del Curso (2/4)

- **Módulo 3. El Scrum Team y el Rol del Scrum Master**
  - Estructura y responsabilidades del Scrum Team.
  - Qué hace y qué no hace un Scrum Master.
  - El liderazgo servicial: coaching, facilitación y mentoring.
  - Cómo eliminar impedimentos y promover la autogestión.
  - Ejemplos de disfunciones comunes (ScrumBut) y cómo corregirlas.
- **Módulo 4. Los Eventos de Scrum**
  - El Sprint como contenedor de todo.
  - Sprint Planning: propósito, enfoque en el “por qué”, “qué” y “cómo”.
  - Daily Scrum: autogestión, foco y visibilidad del progreso.
  - Sprint Review: transparencia y colaboración con stakeholders.
  - Sprint Retrospective: mejora continua y experimentación.
  - Time-boxing y cómo mantener eventos productivos.

✨ ***El conocimiento cobra sentido cuando se convierte en acción con propósito.***



Scrum Master Profesional

Contenido del Curso (3/4)

- **Módulo 5. Artefactos y Compromisos**
  - Product Backlog, Sprint Backlog e Increment.
  - Compromisos: Objetivo del Producto, Objetivo del Sprint y Definición de Terminado.
  - Refinamiento del backlog.
  - Técnicas ágiles de estimación y priorización (INVEST, Planning Poker, SMART).
  - Control empírico del progreso (Burn-down, Burn-up).
- **Módulo 6. Scrum Mastery y Facilitación Ágil**
  - Habilidades esenciales: escucha activa, coaching, mediación, gestión de conflictos.
  - Cómo crear un entorno de seguridad psicológica.
  - Métricas ágiles: flujo, valor, felicidad del equipo.
  - Introducción a escalamiento y madurez organizacional



✨ ***El conocimiento cobra sentido cuando se convierte en acción con propósito.***

Scrum Master Profesional

Contenido del Curso (4/4)

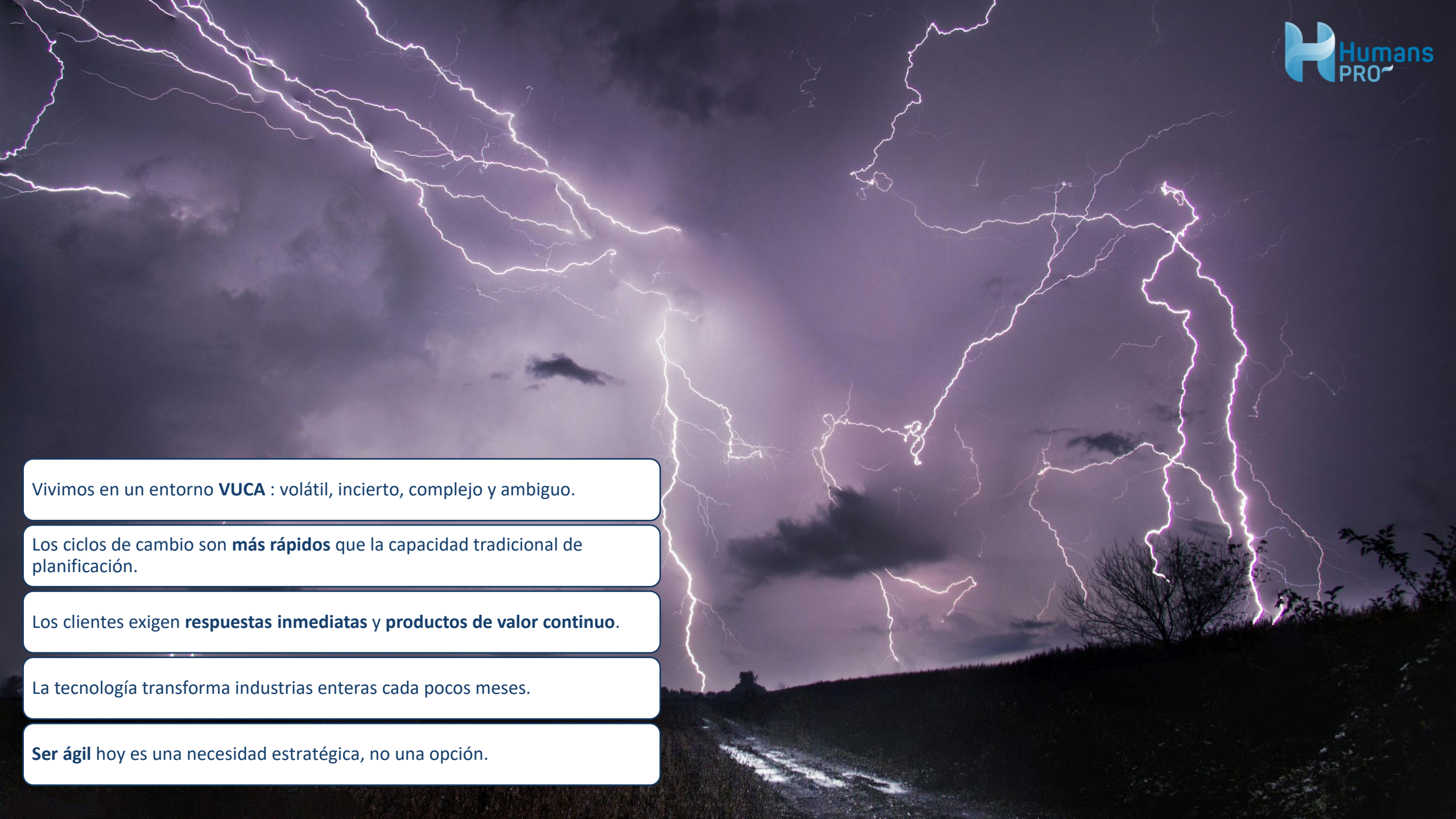
- **Módulo 7. Preparación para la Certificación**
  - Repaso integral de conceptos clave.
  - Tips de examen.
  - Simulacro de 20 preguntas tipo test.
  - Rutas de aprendizaje continuo del Scrum Master.





¿Qué entienden por *agilidad* en su día a día?

https://shorturl.at/17bu3



Vivimos en un entorno **VUCA** : volátil, incierto, complejo y ambiguo.

Los ciclos de cambio son **más rápidos** que la capacidad tradicional de planificación.

Los clientes exigen **respuestas inmediatas** y **productos de valor continuo**.

La tecnología transforma industrias enteras cada pocos meses.

Ser ágil hoy es una necesidad estratégica, no una opción.



“Para sobrevivir en un futuro VUCA hay que empezar a trabajar en el presente, y mejor hacerlo cuanto antes”

- **¿Cuánto sabemos sobre una determinada situación concreta?**
- **¿En qué medida podemos anticipar el resultado de nuestras acciones?**



Formación constante Afrontar los cambios



Entorno VUCA es imprescindible **adaptarse a los cambios e imprevistos**

- Abordar la volatilidad con una correcta visión de futuro (V).
- Afrontar la incertidumbre con conocimiento, formación, actualización constante, es definitiva con entendimiento (U).
- Aportar claridad, simplicidad y sencillez en la ejecución de tareas y acciones dentro del seno de la organización (C).
- Acabar con la ambigüedad con agilidad, con una rápida capacidad de reacción ante los imprevistos que sucedan y que atenten contra la planificación estratégica de la organización (A).





Las organizaciones se han dado cuenta de que la expansión ha creado la necesidad de un lenguaje común, una mentalidad abierta y la voluntad de ser flexible en la forma en que los productos y entregables se llevan al mercado.

Además, reconocen que hay múltiples maneras de lograr una entrega exitosa.

Libro PMI – Agile PG-SPA p.6



Año 1968

"crisis del software"

- **continuos** retrasos,
- **sobrecostes** y
- **deficiencias** de calidad o utilidad que se producían en el desarrollo de software



Takeuchi & Nonaka

vislumbran el nacimiento del "agilismo"

Año 1986

La realidad fue que la gestión de proyectos predictiva no funcionaba como debiera en **entornos inestables**, con **gran competencia** y cuando los **requisitos iniciales cambiaban.**

Humans PRO

- Metodología en cascada (Waterfall): todo se define antes de comenzar.
- Cambiar requerimientos = reiniciar el proyecto.
- Comunicación unidireccional, documentación excesiva.




Proyectos de software cancelados o entregados tarde.
Equipos agotados, clientes frustrados.
La complejidad superaba los planes.
El Caos de los 90s
DIC
GATŌN-AS-



Año 1968

"crisis del software"

- **continuos** retrasos,
- **sobrecostes** y
- **deficiencias** de calidad o utilidad que se producían en el desarrollo de software

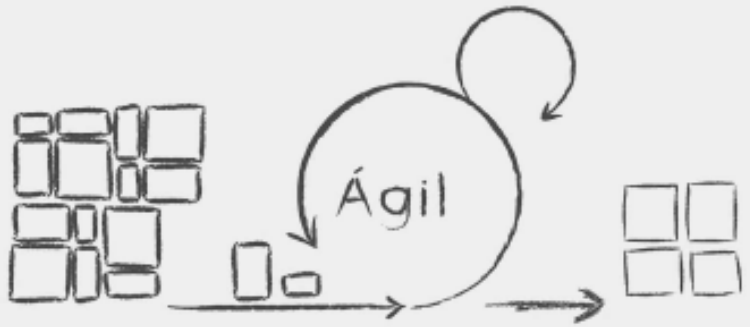


Takeuchi & Nonaka

vislumbran el nacimiento del "agilismo"

Año 1986

La realidad fue que la gestión de proyectos predictiva no funcionaba como debiera en **entornos inestables**, con **gran competencia** y cuando los **requisitos iniciales cambiaban.**



Año 2001

reunión Salt Lake City

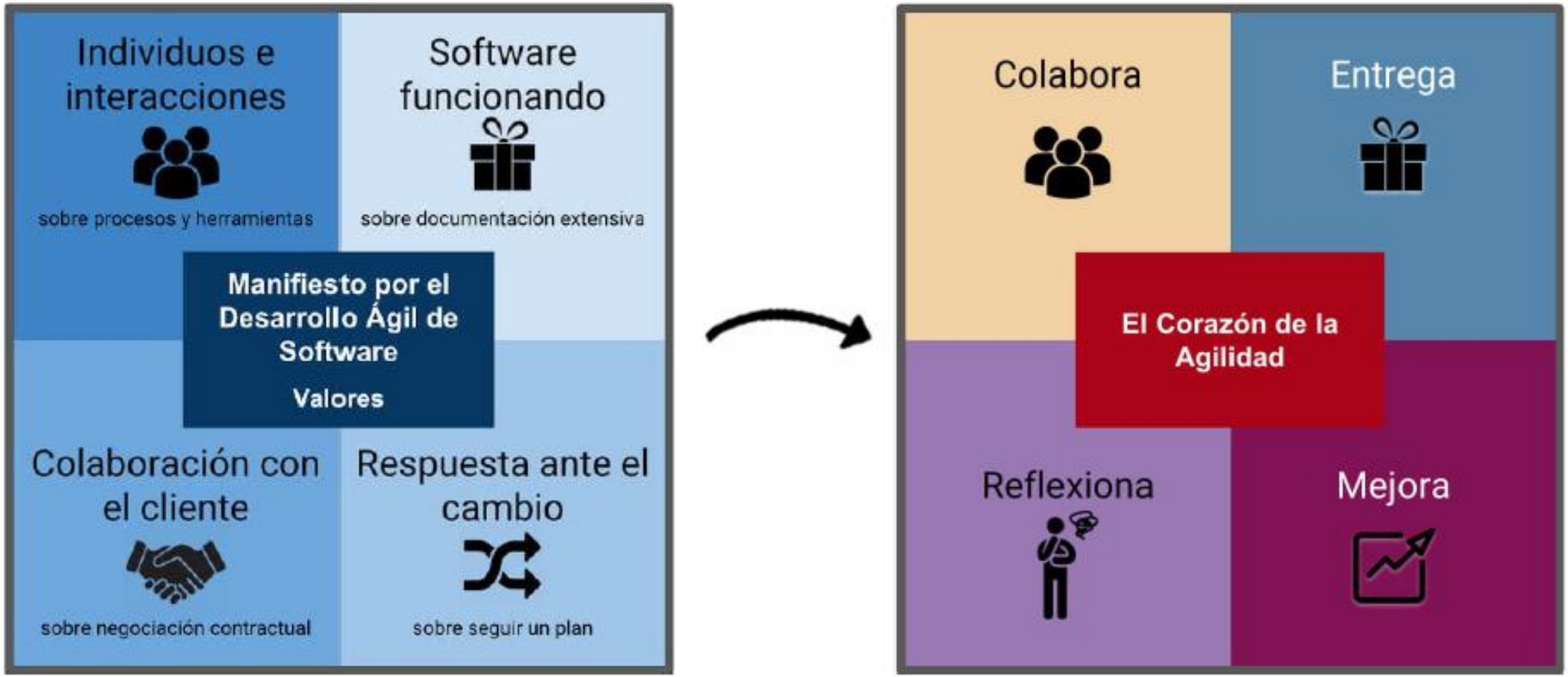
17 expertos en programación

Se acuña el término "Métodos Ágiles"

y surgen los 4 valores y los 12 principios que conforman el "Manifiesto Ágil"

https://agilemanifesto.org/iso/es/principles.html

Kent Beck, Ken Schwaber, Jeff Sutherland


Individuos e interacciones
sobre procesos y herramientas
Software funcionando
sobre documentación extensiva
Manifiesto por el Desarrollo Ágil de Software
Valores
Colaboración con el cliente
sobre negociación contractual
Respuesta ante el cambio
sobre seguir un plan
Colabora
Entrega
El Corazón de la Agilidad
Reflexiona
Mejora

VALORES ÁGILES

INDIVIDUOS E INTERACCIONES



SOBRE



PROCESOS Y HERRAMIENTAS

SOFTWARE FUNCIONANDO



SOBRE



DOCUMENTACION EXTENSIVA

COLABORAR CON EL CLIENTE



SOBRE



NEGOCIACION CONTRACTUAL

RESPONDER AL CAMBIO

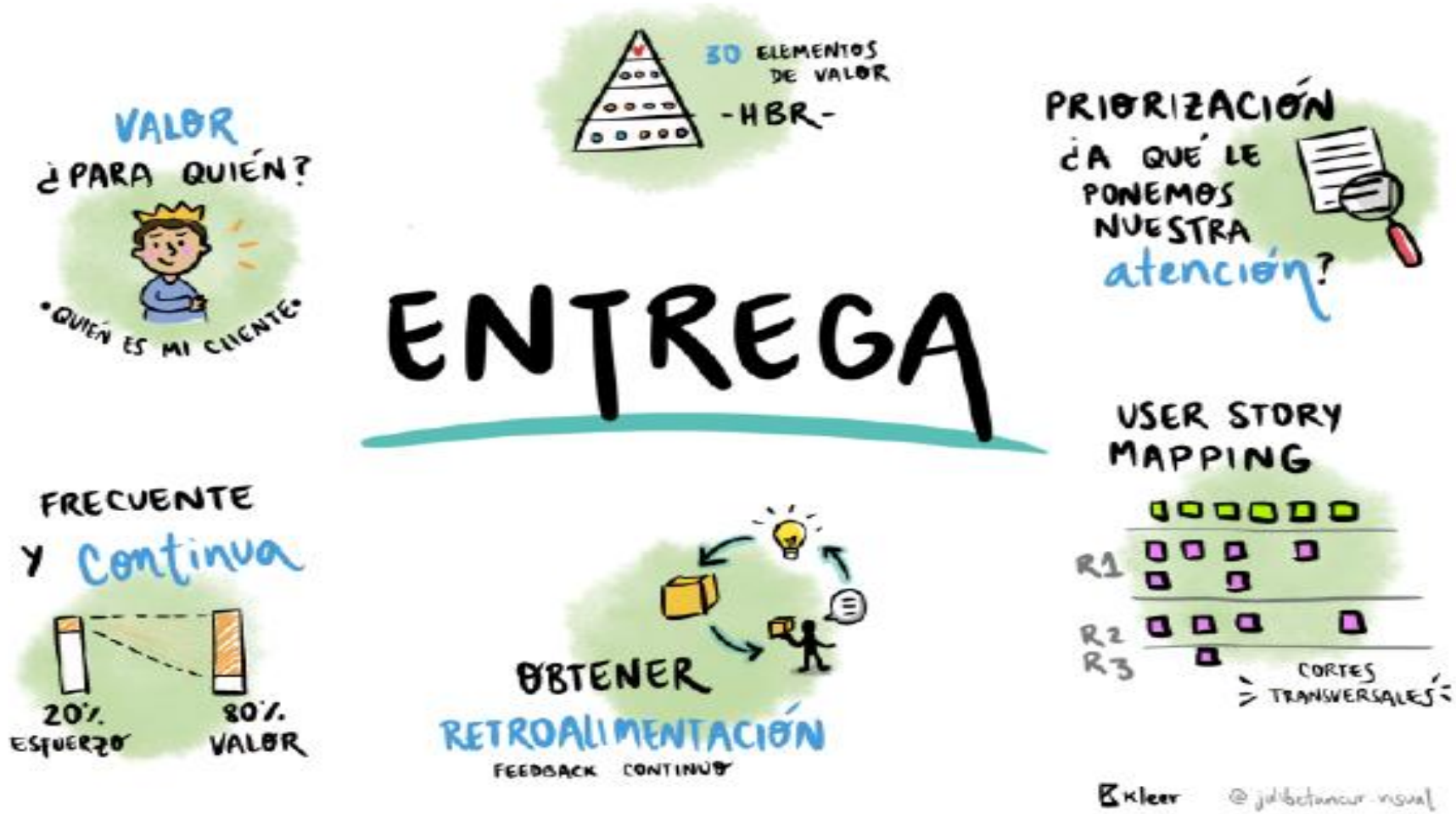


SOBRE

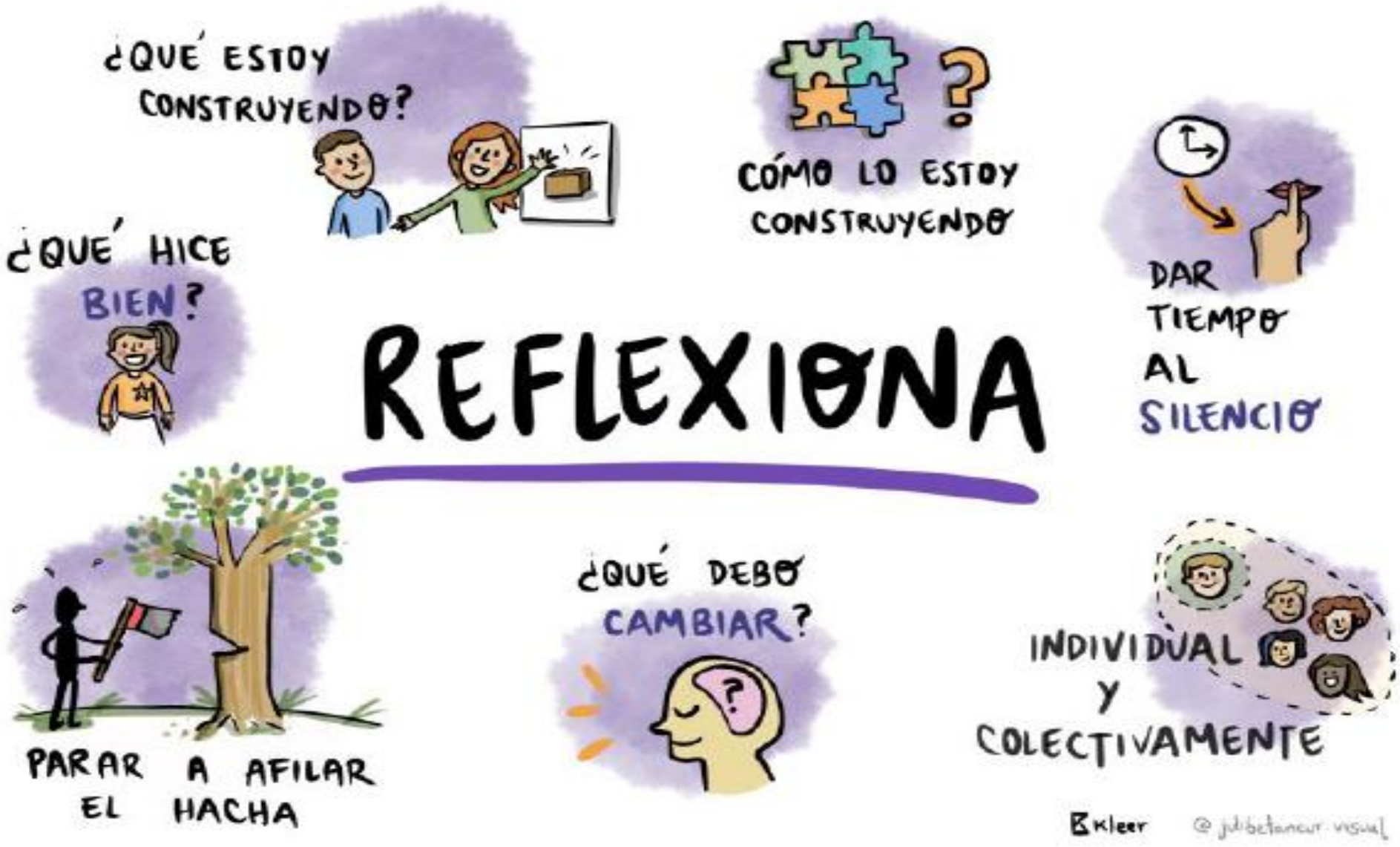


SEGUIR UN PLAN


PROPÓSITO
común
COCREADO
ESPACIOS
COMUNES
EQUIPOS MULTIDISCIPLINARIOS
FOCO Y
MOTIVACIÓN
LIDERAZGO EMERGENTE
ENTENDER
CÓMO PUEDO
aportar
COLABORA
ooops!
ENTENDER
VULNERABILIDAD
¡QUE EL
EQUIPO
GANE!
help!
OFRECER
Y PEDIR
AYUDA
CONFIANZA
genuina
Skleer
@julibetancur.visual



"Cuando entregamos, lo que estamos construyendo genera valor. Se promueven entregas tempranas y frecuentes, para aprender temprano y ajustar el curso que estemos siguiendo. Es clave estar cerca de nuestros usuarios y clientes para asegurarnos que estamos entregándoles algo que realmente valoren."



"Hacer pausas frecuentes para pensar en la forma como nosotros, nuestros equipos y nuestras organizaciones están haciendo las cosas. Acortar los ciclos de retroalimentación es una de las claves a tener en cuenta."

MEJORA

SALIR DE LA QUEJA





MEDICIONES

QUE SEA AYUDA, NO OBSTÁCULO



ENTORNO SEGURO PARA equivocarnos

EXPERIMENTOS CONTENIDOS



PENSAR EN PASOS PEQUEÑOS



EXPERIMENTAR

VALIDAR APRENDIZAJE

MEJORAR PRODUCTOS Y PROCESOS



Kleer @julibetancur.visual

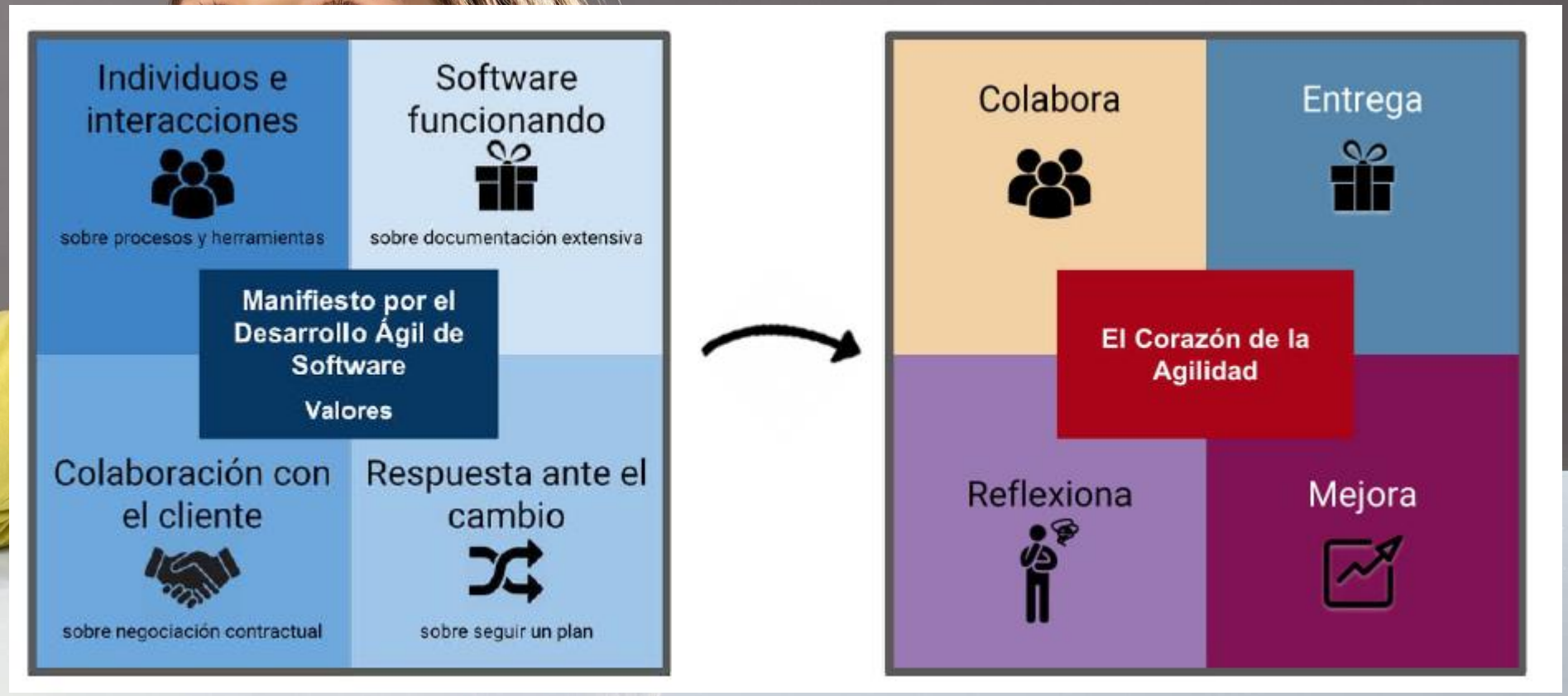
Del documento al valor

De controlar el trabajo	A empoderar equipos
De seguir un plan	Entregar valor continuamente.
De reportar	A aprender y adaptarse.



¿Qué valor del Manifiesto te resulta más desafiante de aplicar?

¿Cómo podrías integrarlo en tu trabajo diario?



1. **Nuestra mayor prioridad es satisfacer al cliente** mediante la entrega temprana y continua de ~~***software con***~~ valor.
2. **Aceptamos que los requisitos cambien**, incluso en etapas tardías del desarrollo. Los procesos Ágiles aprovechan el **cambio para proporcionar ventaja competitiva al cliente.**
3. Entregamos ~~**software funcional**~~ **(Valor)** frecuentemente, **entre dos semanas y dos meses**, con preferencia al periodo de tiempo más corto posible.
4. Los responsables de negocio y los desarrolladores **trabajamos juntos** de forma cotidiana durante todo el proyecto.
5. Los proyectos se desarrollan en torno a **individuos motivados**. Hay que darles el entorno y el apoyo que necesitan, y confiarles la ejecución del trabajo.
6. El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la **conversación cara a cara.**
7. El ~~**software**~~ **(Entregable) funcionando** es la medida principal de progreso.
8. Los procesos Ágiles promueven el **desarrollo sostenible.** Los promotores, desarrolladores y usuarios debemos ser capaces de mantener un **ritmo constante de forma indefinida**.
9. La **atención continua a la excelencia técnica** y al buen diseño mejora la Agilidad.
10. **La simplicidad**, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.
11. Las mejores arquitecturas, requisitos y diseños emergen de **equipos auto-organizados**.
12. **A intervalos regulares el equipo reflexiona** sobre cómo ser más efectivo para a continuación ajustar y perfeccionar su comportamiento en consecuencia.

(*2) Con costo adicional, siempre y cuando NO cambie la visión del proyecto.

(*5) Teoría Y

(*8) Ritmo constante = Productividad Contante = Velocidad

(*9) Calidad Permanente + Pruebas

(*9) Un producto se determina terminado con 100% de Calidad.

(*10) Eliminar requerimientos que no le dan valor al cliente. Solo necesario y Multifuncional.

12 PRINCIPIOS ÁGILES





Satisfacer a los clientes



Requisitos cambiantes



Entrega de valor frecuente



Colaboración diaria



Equipos motivados



Comunicación cara a cara



Medición del progreso



Ritmo de trabajo sostenible



Excelencia técnica



Simplicidad



Equipos autoorganizados



Mejora continua

12 PRINCIPIOS ÁGILES

Principio Ágil	Ejemplo real de aplicación
🏃 Entrega temprana y continua	Publicar versiones del producto cada 2–4 semanas con feedback real del usuario.
🔁 Aceptar el cambio	Repriorizar el Product Backlog en cada Sprint según nueva información.
💬 Comunicación constante	Daily Scrum enfocada en colaboración, no en reportes.
⚖️ Ritmo sostenible	Definir capacidad real del equipo para evitar sobrecarga y burnout.
🤝 Colaboración diaria	Product Owner y Developers revisan juntos el progreso del Sprint.
💡 Reflexión y mejora continua	Realizar retrospectivas con acciones concretas que se aplican en el siguiente Sprint.

Excelencia técnica · Simplicidad · Equipos autoorganizados · Mejora continua



Reflexión individual y grupal:

- Identifica un principio del Manifiesto Ágil que más te costaría aplicar.
- Escríbelo en un post-it virtual (Miro) junto con una breve razón.
- Comenta con el grupo los patrones comunes.

¿Que es ágil?

Concepto

Agile es un conjunto de métodos y prácticas de gestión de proyectos que enfatiza la flexibilidad, la colaboración interdisciplinaria, la calidad del producto y la adaptabilidad a lo largo de todo el ciclo de vida del proyecto. Está basado en iteraciones cortas o sprints, v revisiones y ajustes constantes según sea necesario. Agile no es una metodología en sí, sino un enfoque adaptable a diferentes tipos de proyectos para mejorar la eficiencia y la efectividad.



¿Que es ágil?

Contraste vs Waterfall

A diferencia de las metodologías tradicionales como Waterfall, donde el proyecto se realiza de manera secuencial desde la concepción hasta la conclusión, Agile permite una ejecución más dinámica y adaptativa. En Waterfall, los errores detectados en fases tempranas pueden ser costosos de corregir más tarde, ya que cada fase depende de la entrega completa y exacta de la anterior. Agile, por su parte, facilita una respuesta rápida a los cambios y la incorporación de feedback durante todo el proyecto, lo cual es crucial en entornos volátiles y en proyectos donde los requisitos pueden evolucionar o no estar completamente definidos desde el inicio.





¿Qué es la agilidad?

- Capacidad de adaptarse rápidamente al cambio y entregar valor continuo.
- Ejemplo: Cambiar de ruta si hay tráfico.

Ser ágil vs. hacer ágil

Ser Ágil	Hacer Ágil
Mentalidad y cultura.	Uso de herramientas y ceremonias.
Basado en valores y principios.	Basado en procesos y técnicas.
Fomenta colaboración, confianza y aprendizaje.	Sigue prácticas sin entender el propósito.
Mide el éxito por valor entregado y mejora continua.	Mide el éxito por cumplimiento del proceso.
Requiere cambio cultural y liderazgo.	Puede quedarse en lo superficial.

"Agile no es algo que haces, es algo que eres."



Una Mente Ágil

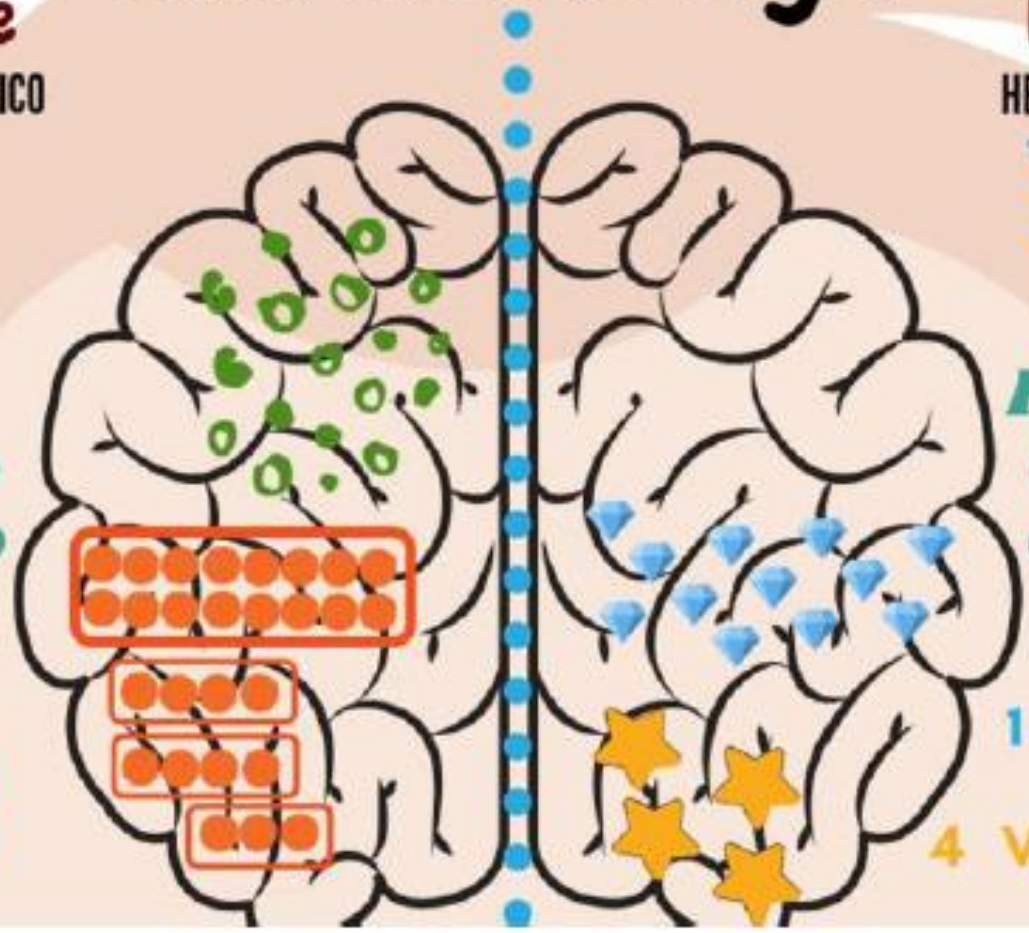
Haciendo Agile

HEMISFERIO LÓGICO MATEMÁTICO

INSPECCIÓN
ADAPTACIÓN
EMPIRISMO
ENTREGA

Atención
Desperdicio
y flujo

LeSS
Kanban
Scrum
XP



Pensando Agile

HEMISFERIO CREATIVO INTUITUVO

TRANSPARENCIA
COLABORACIÓN
CONFIANZA

Foco
Centrado en el cliente y el valor

12 PRINCIPIOS

4 VALORES

Del pensamiento ágil al contexto



La agilidad **no es solo para software**: hoy aplica a todas las industrias.

Las organizaciones enfrentan entornos **complejos, inciertos y cambiantes**.

El pensamiento ágil impulsa **adaptabilidad, colaboración y aprendizaje continuo**.

Los equipos ágiles prosperan en contextos donde se valora **la experimentación y el feedback temprano**.

Scrum es una manifestación práctica de ese pensamiento ágil.

El mundo VUCA - BANI

🔄 **Volatilidad:** los cambios ocurren con velocidad y sin aviso.

❓ **Incertidumbre:** no siempre hay datos claros para decidir.

🧩 **Complejidad:** todo está interconectado; las soluciones no son lineales.

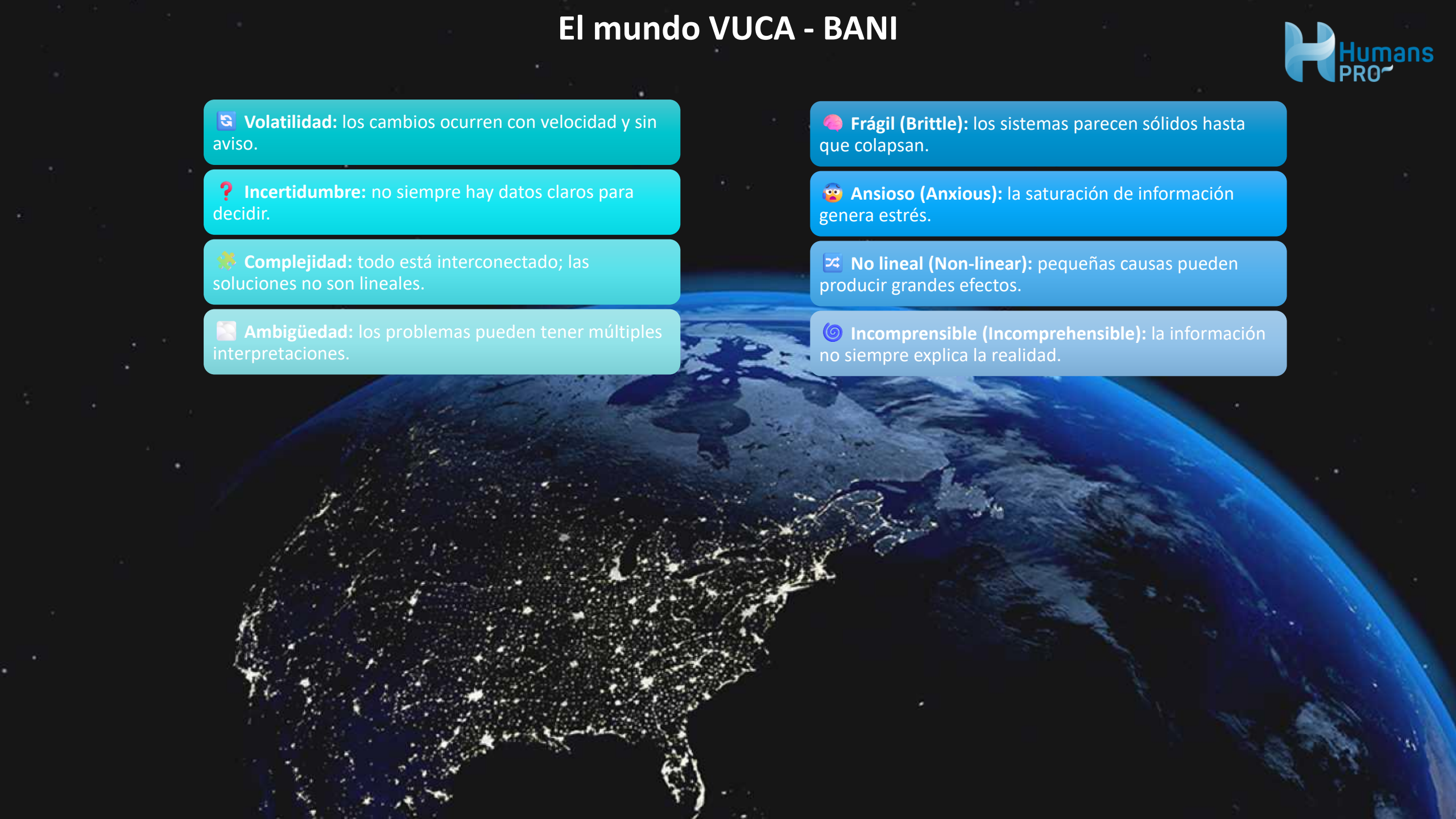
🌫️ **Ambigüedad:** los problemas pueden tener múltiples interpretaciones.

🧠 **Frágil (Brittle):** los sistemas parecen sólidos hasta que colapsan.

😨 **Ansioso (Anxious):** la saturación de información genera estrés.

🔀 **No lineal (Non-linear):** pequeñas causas pueden producir grandes efectos.

🌀 **Incomprensible (Incomprehensible):** la información no siempre explica la realidad.



El Modelo Cynefin (Dave Snowden, 1999)

Dominio	Características	Respuesta recomendada
Simple / Claro	Causa-efecto evidente. Buenas prácticas conocidas.	Seguir el proceso estándar.
Complicado	Requiere análisis o expertos. Varias soluciones posibles.	Investigar, analizar, decidir.
Complejo	Relaciones impredecibles. El conocimiento surge al experimentar.	**Probar – aprender – adaptar (Scrum).**
Caótico	No hay relación clara causa-efecto. Crisis, urgencia.	Actuar rápidamente para estabilizar.



🔬 “El empirismo: base de Scrum”

Empirismo = aprender haciendo y ajustando según la realidad.

En entornos complejos, **no podemos predecir**; solo **experimentar, inspeccionar y adaptar**.

Scrum crea **bucles cortos de aprendizaje** llamados *Sprints*.

Cada Sprint es una oportunidad para **inspeccionar el progreso** y **adaptar el rumbo**.

Las decisiones se basan en **datos observables**, no en intuiciones o jerarquías.

Humans PRO

PMI

- Enfoque proyecto
- Requiere un jefe
- Entregas productos completos
- Estimación de tiempo, alcance y costo



AGILE

- Enfoque en producto
- Empoderamiento equipo
- Entregas tempranas
- Participación permanente del cliente

Tradicional Vs Ágil

Tradicional

- Plan fijo
- Documentación exhaustiva
- Entregas al final
- Cliente pasivo

Ágil

- Plan adaptativo
- Comunicación continua
- Entregas incrementales
- Cliente activo




¿Dónde falla el enfoque tradicional en nuestros proyectos?



Por qué falla la gestión tradicional

Principales causas de falla en la gestión tradicional:

1. **Planificación rígida:** se asume que todo puede predecirse desde el inicio.
2. **Exceso de documentación:** se prioriza el papeleo sobre la entrega de valor.
3. **Comunicación en silos:** poca colaboración entre equipos y con el cliente.
4. **Resistencia al cambio:** las modificaciones se perciben como errores, no como aprendizaje.
5. **Enfoque en el control, no en el valor:** el éxito se mide por cumplir el plan, no por satisfacer al usuario.
6. **Feedback tardío:** se descubre que algo está mal solo al final del proyecto.



“La agilidad no elimina la planificación; la hace continua y adaptable.”

De esto…	A esto…
• Planificación exhaustiva antes de actuar. • Control jerárquico y microgestión. • Éxito = cumplimiento del plan. • Equipos que ejecutan órdenes.	• Iterar y ajustar sobre la marcha. • Liderazgo servicial y empoderamiento. • Éxito = valor entregado y aprendizaje continuo. • Equipos que deciden, se autoorganizan y aprenden.

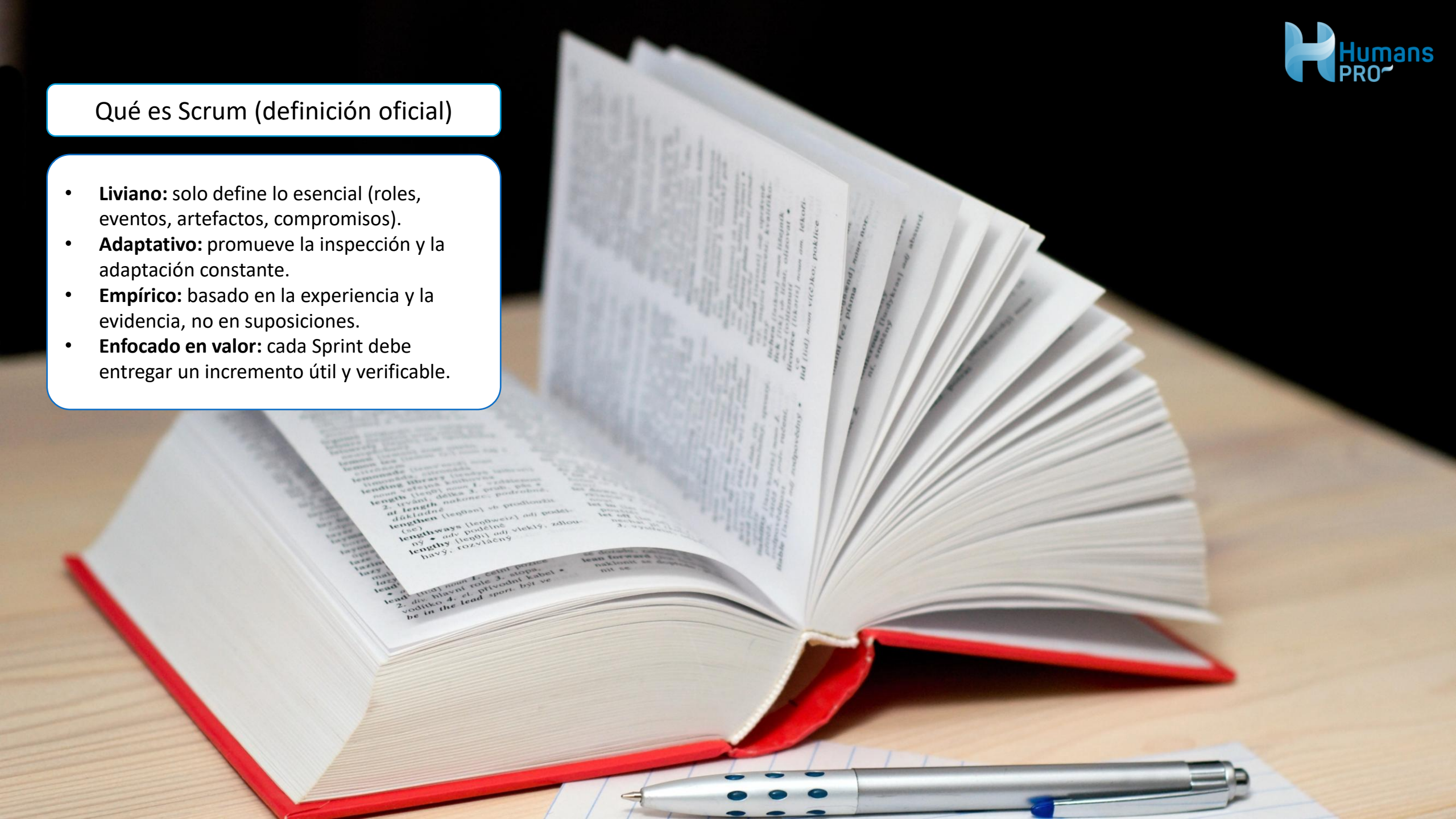
Qué es Scrum (definición oficial)

Definición oficial (Scrum Guide 2020):
"Scrum es un marco de trabajo liviano que ayuda a las personas, equipos y organizaciones a generar valor a través de soluciones adaptativas para problemas complejos."



Qué es Scrum (definición oficial)

- **Liviano:** solo define lo esencial (roles, eventos, artefactos, compromisos).
- **Adaptativo:** promueve la inspección y la adaptación constante.
- **Empírico:** basado en la experiencia y la evidencia, no en suposiciones.
- **Enfocado en valor:** cada Sprint debe entregar un incremento útil y verificable.





Scrum no es…

❌ **Una metodología rígida**
→ No prescribe cómo hacer el trabajo; establece un marco para aprender y adaptarse.
❌ **Solo para desarrollo de software**
→ Se aplica en marketing, educación, diseño, investigación y más.
❌ **Una herramienta o software (como Jira o Trello)**
→ Son medios de soporte, no Scrum en sí.
❌ **Una fórmula mágica para el éxito**
→ Scrum revela problemas; el equipo debe resolverlos.
❌ **Un conjunto de reuniones obligatorias**
→ Los eventos existen para inspeccionar y adaptar, no por cumplimiento formal.
❌ **Microgestión disfraza de colaboración**
→ Scrum fomenta la autogestión, no el control diario.

Los equipos Scrum son auto-organizados y multifuncionales.



Dueño del Producto o
Product Owner (PO)

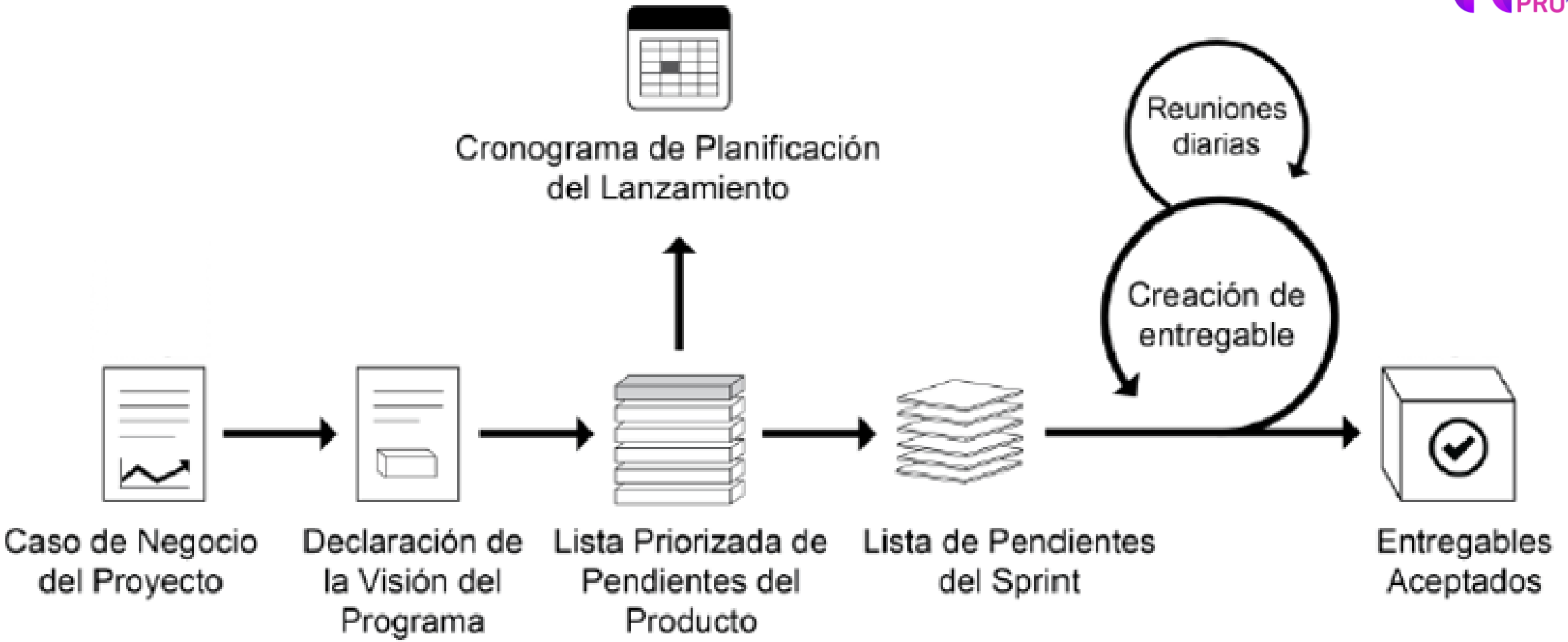


Equipo de Desarrollo o
Development Team o
Scrum Development (SD)



Scrum Master (SM)

PO + SD + SM = Scrum Core Team



Desviación normal entre planeado y real 5% a 10%

Project Charter
Alcance,
Tiempo y Costo

Lo hace el PO lo estima el SM

Incluye posible fecha de cada Sprint

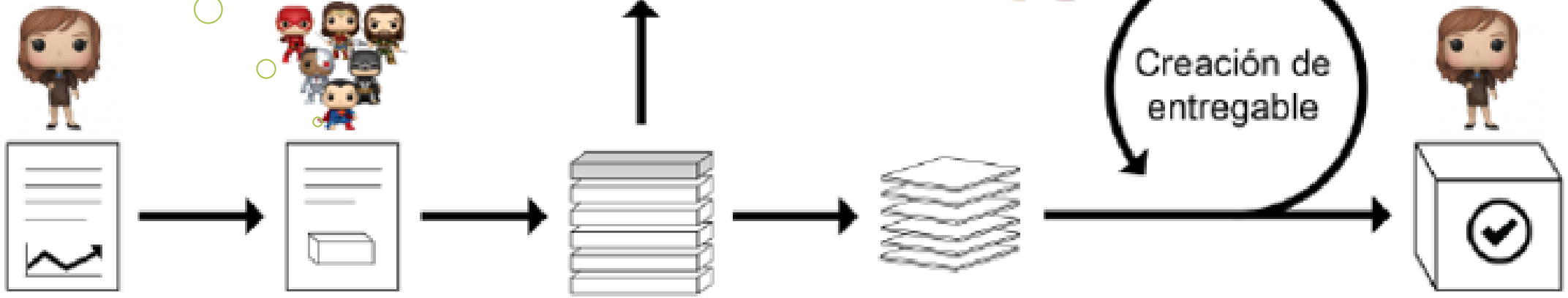
Cronograma de Planificación del Lanzamiento

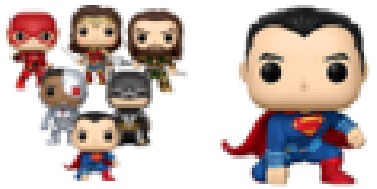
Reuniones diarias

Creación de entregable









Beneficios
Presupuesto
Problemas

Desviación normal entre planeado y real 5% a 10%

SPRINT + ROLES




Vamos al miro y miremos qué es y qué nos es Scrum

Los tres pilares del marco Scrum

Transparencia	Inspección	Adaptación
• Todos deben tener visibilidad del trabajo, los objetivos y el progreso. • Ejemplo: Tableros visibles, “Definition of Done”, reuniones abiertas.	• Revisar frecuentemente el progreso y los resultados. • Ejemplo: Daily Scrum, Sprint Review.	• Ajustar el rumbo en función del aprendizaje y la retroalimentación. • Ejemplo: Retrospectivas y replanificación en cada Sprint.

Pilar	Ejemplos reales en Scrum
Transparencia	- Tablero visible (Scrum Board, Jira, Miro). - Definition of Done compartida. - Backlog accesible para todos.
Inspección	- Daily Scrum para revisar progreso. - Sprint Review con stakeholders. - Refinamientos frecuentes del backlog.
Adaptación	- Retrospectivas para mejorar. - Ajustes en la planificación de sprint. - Cambios de estrategia según resultados.



Conclusiones clave del módulo

1. La **agilidad** es una **mentalidad**, no una metodología.
2. El **Manifiesto Ágil** cambió la forma de trabajar en entornos de cambio constante.
3. Scrum se basa en **empirismo**: aprender, inspeccionar y adaptar.
4. Los **tres pilares** —transparencia, inspección y adaptación— hacen visible la realidad del trabajo.
5. El rol del **Scrum Master** empieza aquí: **guiar al equipo hacia el aprendizaje continuo**.



Módulo 2 – Valores y Cultura Scrum



Scrum Master Profesional

Contenido del Curso

- **Módulo 1 - Fundamentos de la Agilidad y de Scrum**
  - El pensamiento ágil y el Manifiesto Ágil.
  - Contexto: complejidad, adaptabilidad y empirismo.
  - Diferencias entre la gestión tradicional y la ágil.
  - ¿Qué es Scrum y qué no es Scrum?
  - Los tres pilares de Scrum: transparencia, inspección y adaptación.
- **Módulo 2 – Valores y Cultura Scrum**
  - Compromiso, Foco, Apertura, Respeto y Coraje.
  - Cómo los valores sustentan la autogestión y la transparencia.
  - Conductas esperadas del Scrum Master para fomentar la cultura Scrum.

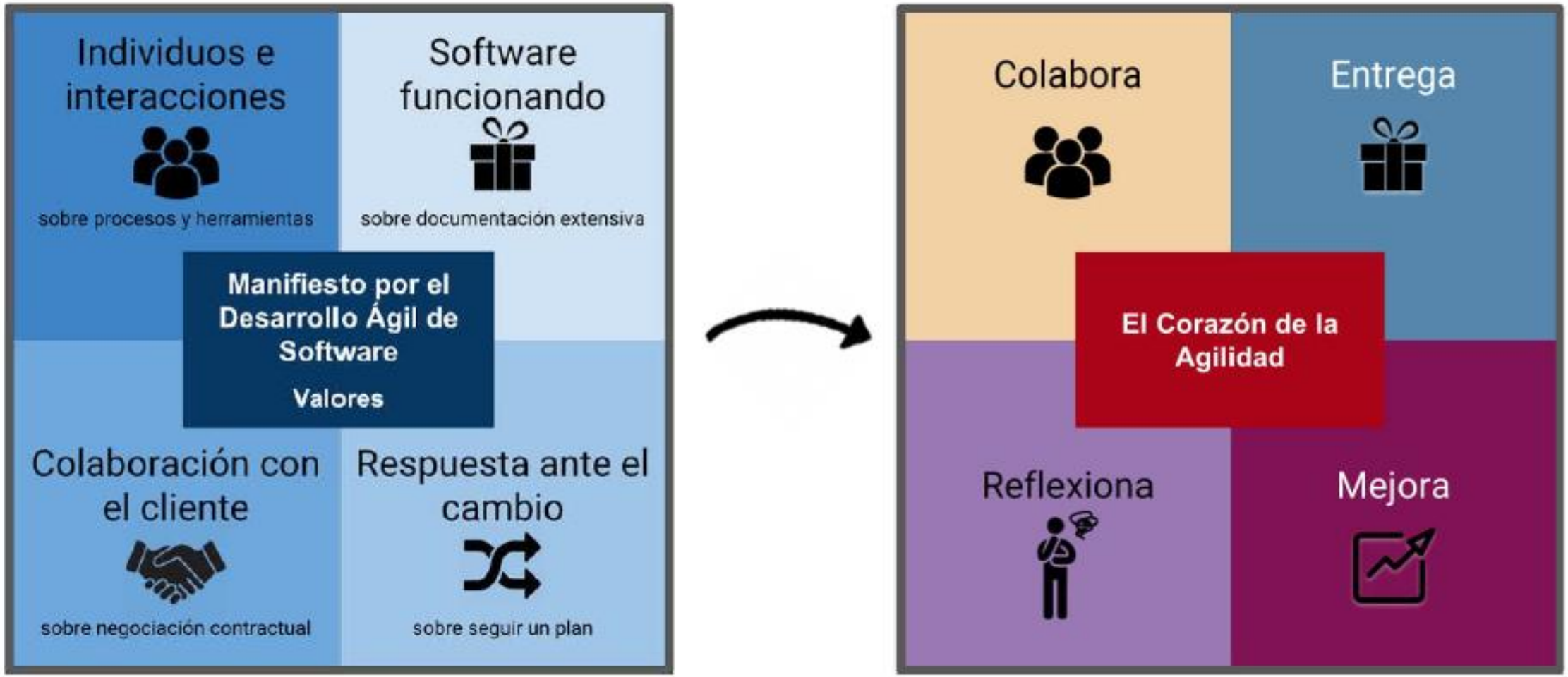


✨ ***El conocimiento cobra sentido cuando se convierte en acción con propósito.***

Recordemos el Módulo 1: Fundamentos de la Agilidad y de Scrum

- La agilidad nació como respuesta a la rigidez de la gestión tradicional.
- Scrum se basa en **empirismo**: inspección, adaptación y transparencia.
- Los tres pilares de Scrum sostienen su práctica:
- 🔍 Transparencia
- 🔁 Inspección
- 🔄 Adaptación
- Scrum no es una metodología, sino un **marco ligero para aprender haciendo.**




Individuos e interacciones
sobre procesos y herramientas
Software funcionando
sobre documentación extensiva
Manifiesto por el Desarrollo Ágil de Software
Valores
Colaboración con el cliente
sobre negociación contractual
Respuesta ante el cambio
sobre seguir un plan
Colabora
Entrega
El Corazón de la Agilidad
Reflexiona
Mejora

12 PRINCIPIOS ÁGILES





Satisfacer a los clientes



Requisitos cambiantes



Entrega de valor frecuente



Colaboración diaria



Equipos motivados



Comunicación cara a cara



Medición del progreso



Ritmo de trabajo sostenible



Excelencia técnica



Simplicidad



Equipos autoorganizados



Mejora continua

Los equipos Scrum son auto-organizados y multifuncionales.



Dueño del Producto o
Product Owner (PO)

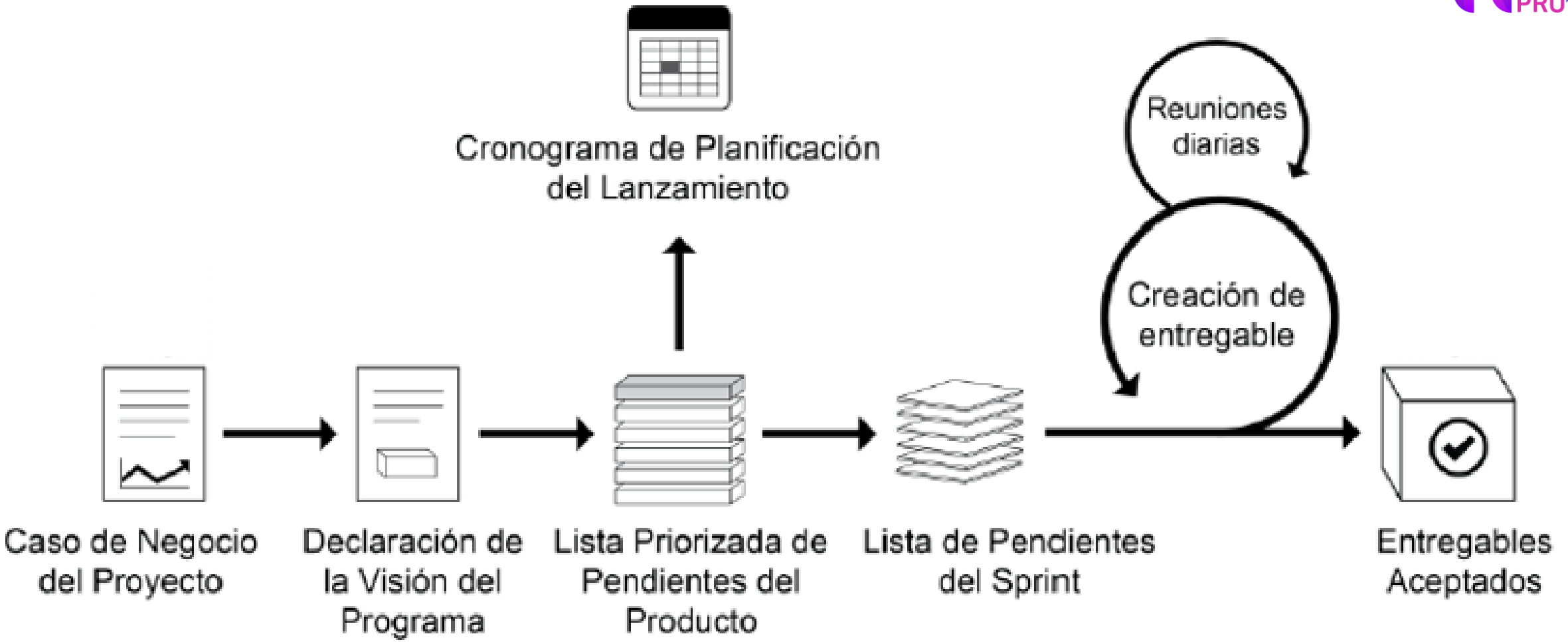


Equipo de Desarrollo o
Development Team o
Scrum Development (SD)

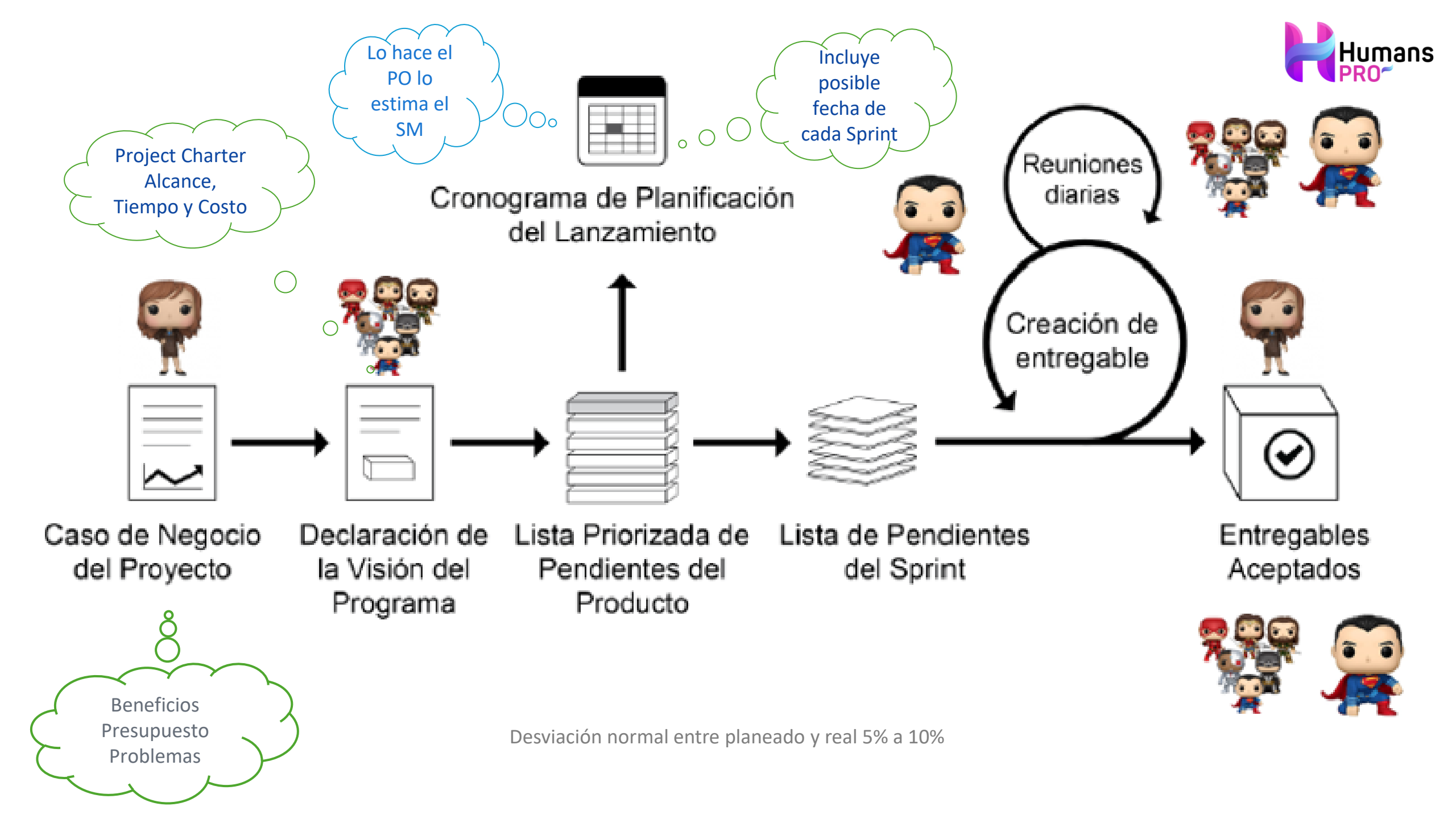


Scrum Master (SM)

PO + SD + SM = Scrum Core Team



Desviación normal entre planeado y real 5% a 10%



Desviación normal entre planeado y real 5% a 10%

¿Cuál de los siguientes enunciados describe mejor el propósito de Scrum?

A. Entregar proyectos dentro del alcance, tiempo y presupuesto definidos.
B. Establecer un conjunto rígido de prácticas para equipos ágiles.
C. Proveer un marco empírico para desarrollar, entregar y sostener productos complejos.
D. Garantizar la ausencia de errores en los productos entregados.



¿Qué representan los tres pilares de Scrum?

A. Transparencia, Inspección y Adaptación.
B. Planificación, Ejecución y Control.
C. Compromiso, Colaboración y Comunicación.
D. Product Backlog, Sprint Backlog y Incremento.



¿Cuál de los siguientes elementos refleja mejor el principio del empirismo?

A. Planificar exhaustivamente antes de comenzar.
B. Tomar decisiones basadas en experiencia y evidencia.
C. Confiar en las instrucciones del gerente de proyecto.
D. Evitar los cambios durante el Sprint.



¿Qué diferencia fundamental existe entre la gestión tradicional y la ágil?

A. En la gestión ágil los equipos trabajan sin planificación.
B. En la gestión ágil se valora la adaptabilidad sobre el cumplimiento del plan.
C. En la gestión tradicional se promueve la autogestión del equipo.
D. En la gestión tradicional no existen roles definidos.



¿Cuál de las siguientes afirmaciones es correcta respecto a Scrum?

A. Scrum es una metodología rígida para gestionar proyectos.
B. Scrum es un marco adaptable para generar valor en entornos complejos.
C. Scrum sustituye la gestión de proyectos tradicional sin ajustes.
D. Scrum solo puede aplicarse a equipos de desarrollo de software.



Scrum Master Profesional

Contenido del Curso

- **Módulo 1 - Fundamentos de la Agilidad y de Scrum**
  - El pensamiento ágil y el Manifiesto Ágil.
  - Contexto: complejidad, adaptabilidad y empirismo.
  - Diferencias entre la gestión tradicional y la ágil.
  - ¿Qué es Scrum y qué no es Scrum?
  - Los tres pilares de Scrum: transparencia, inspección y adaptación.
- **Módulo 2 – Valores y Cultura Scrum**
  - Compromiso, Foco, Apertura, Respeto y Coraje.
  - Cómo los valores sustentan la autogestión y la transparencia.
  - Conductas esperadas del Scrum Master para fomentar la cultura Scrum.



✨ ***El conocimiento cobra sentido cuando se convierte en acción con propósito.***

Módulo 2: Valores y Cultura Scrum

- Compromiso, Foco, Apertura, Respeto y Coraje.
- Cómo los valores sustentan la autogestión y la transparencia.
- Conductas esperadas del Scrum Master para fomentar la cultura Scrum.

Los valores de Scrum

- Compromiso
- Foco
- Apertura
- Respeto
- Coraje





Compromiso: hacer lo que decimos que haremos

- Cumplir con lo que el equipo acuerda en el **Sprint Goal**.
- Participar activamente y aportar al éxito colectivo.
- Ser transparente sobre el progreso y los impedimentos.
- Priorizar el valor del producto sobre los intereses individuales.
- Compromiso no es trabajar más… es **asumir responsabilidad por el resultado**.



¿Qué acciones concretas demuestran compromiso en tu equipo?

Compromiso: hacer lo que decimos que haremos

🧩 Dentro del equipo Scrum

- **Cumplir con la Definition of Done** — no entregar nada “a medias”.
- **Actualizar el tablero diariamente** — mantener la transparencia sobre el progreso.
- **Ayudar a un compañero bloqueado** aunque no sea “tu tarea”.
- **Revisar y mejorar el proceso en cada retrospectiva**, aplicando al menos una acción acordada.
- **Cuidar la calidad del código o del producto**, incluso si implica decir “no” a una entrega apresurada.
- **Defender el Sprint Goal** frente a solicitudes que no aportan valor.

🧠 Desde el rol del Scrum Master

- **Proteger al equipo de interferencias externas** para mantener el foco.
- **Cumplir los compromisos como facilitador**, preparando eventos útiles y bien enfocados.
- **Modelar el comportamiento esperado**: puntualidad, preparación, escucha activa.
- **Fomentar la responsabilidad colectiva**, no individual, por el resultado del Sprint.
- **Acompañar con empatía**, pero también con firmeza, cuando el equipo evita conversaciones difíciles.

🏢 En la cultura organizacional

- **Los líderes cumplen lo que prometen** (por ejemplo, eliminar impedimentos cuando se comprometen a hacerlo).
- **Las prioridades del negocio se comunican con claridad**, sin cambiar arbitrariamente.
- **La organización respeta la autonomía del equipo** y confía en su planificación.
- **El valor del compromiso se mide por resultados sostenibles**, no por horas trabajadas.

Foco: priorizar lo que aporta valor

- Concentrarse en el **Objetivo del Sprint (Sprint Goal)** y evitar el multitasking.
- Priorizar el trabajo que genera **mayor impacto en el cliente o producto.**
- Decir **“no”** a tareas que no contribuyen al objetivo acordado.
- Dedicar tiempo de calidad a los eventos Scrum y no tratarlos como formalidades.
- El foco crea claridad, ritmo y mejora la previsibilidad del equipo.

Los ladrones del foco

- Pueden compartir ejemplos de cosas que interrumpen o desvían el foco del equipo
  - Reuniones urgentes
  - Cambios de prioridad
  - Multitasking
  - Correos



Foco: priorizar lo que aporta valor

🧩 **Dentro del equipo Scrum**

- **Sprint Planning enfocado:** el equipo define un objetivo claro y lo usa como brújula durante todo el Sprint.
- **Daily Scrum efectiva:** las conversaciones se centran en el progreso hacia el Sprint Goal, no en tareas secundarias.
- **Eliminar el multitasking:** los miembros no trabajan en múltiples historias a la vez; terminan una antes de iniciar otra.
- **Cierre disciplinado:** si una tarea no aporta al Sprint Goal, se reprograma o se rechaza.
- **Revisión con propósito:** la Sprint Review muestra solo incrementos que aportan valor real, no "trabajo en progreso".

🧠 **Desde el rol del Scrum Master**

- **Protege al equipo de interrupciones externas** (otros proyectos, managers, emergencias mal definidas).
- **Refuerza el foco en los eventos:** mantiene las discusiones alineadas con el objetivo, evitando desvíos.
- **Facilita priorización real:** ayuda al Product Owner y Developers a identificar lo que más valor genera.
- **Detecta patrones de dispersión:** si el equipo cambia de dirección constantemente, promueve una retrospectiva sobre foco.
- **Fomenta claridad en los objetivos:** asegura que cada Sprint Goal sea entendible, medible y relevante.

🏢 **A nivel organizacional**

- **Evitar context switching:** los líderes respetan los Sprints y no reasignan personas durante el ciclo.
- **Cultura de priorización:** los directivos entienden que "todo no puede ser prioridad".
- **Reuniones con propósito:** las reuniones fuera del marco Scrum tienen agenda y duración limitada.
- **Visibilidad del valor:** la organización mide resultados en función del impacto en el cliente, no del volumen de tareas.
- **Comunicación clara:** las metas estratégicas se transmiten a todos los equipos para alinear el esfuerzo.

Apertura: transparencia y aprendizaje

- Ser **honestos sobre el progreso**, los bloqueos y los errores.
- Compartir información relevante con **todo el equipo y los stakeholders**.
- Estar dispuestos a recibir y dar feedback con respeto.
- Mostrar vulnerabilidad sin temor: reconocer lo que no sabemos.
- Promover conversaciones abiertas en lugar de decisiones cerradas.



- Tienen ejemplos de **momentos en que faltó apertura** en un equipo (por miedo, jerarquía o política).

- ¿Qué habría cambiado si el equipo hubiera sido más abierto desde el principio?



Apertura: transparencia y aprendizaje



🧩 **Dentro del Equipo Scrum**

- **Compartir bloqueos temprano:** un miembro del equipo comunica que necesita ayuda antes de que el problema escale.
- **Feedback frecuente:** se revisa el incremento en la Sprint Review sin miedo a mostrar errores.
- **Hablar de fricciones:** durante la retrospectiva, alguien expresa algo que le incomodó y el equipo lo escucha con respeto.
- **Visibilidad del progreso:** todos mantienen actualizado el tablero de trabajo.
- **Celebrar y reconocer:** se agradecen públicamente los logros del equipo.

🧠 **Desde el rol del Scrum Master**

- **Admite cuando algo no sale bien en su facilitación.**
- **Modela apertura emocional:** da ejemplo de vulnerabilidad y transparencia.
- **Promueve un entorno seguro:** anima al equipo a expresar desacuerdos sin consecuencias negativas.
- **Expone impedimentos reales** al Product Owner o management, sin maquillarlos.
- **Da espacio a todas las voces:** se asegura de que las personas más calladas sean escuchadas.

🏢 **A nivel organizacional**

- **Comunicación transparente:** los líderes explican el “por qué” detrás de las decisiones estratégicas.
- **Errores vistos como oportunidades:** se analizan fallos en retrospectiva institucional, sin buscar culpables.
- **Datos accesibles:** métricas, objetivos y avances del producto están visibles para todos.
- **Retroalimentación bidireccional:** los equipos también pueden dar feedback a la dirección.
- **Reconocer públicamente aprendizajes:** se celebran casos donde la apertura llevó a una mejora o innovación.



Respeto: base de la colaboración

- Reconocer que **cada miembro del equipo aporta valor**, sin importar su rol o experiencia.
- Escuchar activamente y permitir que **todas las voces sean escuchadas**.
- Asumir que los demás actúan con la mejor intención.
- Tratar los errores como oportunidades para aprender, no para culpar.
- Fomentar **seguridad psicológica**, donde las personas se sientan libres de expresar ideas o preocupaciones.



Respeto en acción

Tiene algún ejemplo de **“irrespeto cotidiano”** que hayan visto (sin nombrar personas).



Respeto: base de la colaboración

🧩 **Dentro del equipo Scrum**

- Escuchar a todos antes de tomar una decisión importante.
- No interrumpir ni minimizar ideas durante reuniones.
- Agradecer el trabajo o esfuerzo de otros miembros.
- Corregir errores sin sarcasmo o culpa.
- Valorar la experiencia de otros roles (Product Owner, Developers, Scrum Master).

🧠 **Desde el rol del Scrum Master**

- Tratar a todos los participantes como iguales, sin favoritismos.
- Defender al equipo cuando recibe críticas injustas.
- Promover conversaciones difíciles con empatía.
- Cuidar el tono y la forma en que se da feedback.
- Ser paciente con el ritmo de aprendizaje de cada persona.

🏢 **A nivel organizacional**

- Evitar jerarquías rígidas que silencian la voz de los equipos.
- Reconocer públicamente el trabajo bien hecho.
- No culpar a las personas cuando los procesos fallan.
- Crear políticas que fomenten la inclusión y el bienestar.
- Recompensar comportamientos colaborativos, no competitivos.

Coraje: tomar decisiones difíciles

- Afrontar conversaciones difíciles y expresar desacuerdos con respeto.
- Asumir responsabilidad cuando algo no sale bien.
- Defender los valores de Scrum ante presiones externas.
- Decir **“no”** cuando una decisión pone en riesgo la calidad o los principios.
- Probar nuevas ideas, aunque puedan fallar, y aprender de los errores.

Coraje: tomar decisiones difíciles

Tienen alguna situación reciente donde tuvieron que mostrar coraje (o donde no lo hicieron).

- ¿Qué los ayudó —o qué les faltó— para actuar con valentía en ese momento?

Coraje: tomar decisiones difíciles

🧩 **Dentro del equipo Scrum**

- Admitir que una tarea fue mal estimada y replantear el trabajo.
- Pedir ayuda sin miedo a parecer débil.
- Detener una entrega que no cumple la Definition of Done.
- Cuestionar una práctica que no agrega valor.
- Dar feedback honesto en la retrospectiva.

🧠 **Desde el rol del Scrum Master**

- Decir “no” a pedidos que rompen el Sprint o agregan trabajo no planificado.
- Comunicar de forma constructiva cuando la organización interfiere en la autogestión del equipo.
- Defender la importancia de la mejora continua ante la presión de solo “entregar más”.
- Facilitar conversaciones difíciles entre miembros del equipo o stakeholders.
- Admitir cuando su propia facilitación no funcionó y ajustar el enfoque.

🏢 **A nivel organizacional**

- Los líderes reconocen errores públicamente.
- Se detienen proyectos sin sentido en lugar de sostenerlos por orgullo.
- Se fomenta la experimentación, aunque implique fallar.
- Se toman decisiones con base en datos, no en política.
- Se promueve una cultura donde cuestionar con respeto es bien visto.



Los valores como un sistema vivo

- **Los valores no se aplican por separado**
- Los cinco valores **se conectan y retroalimentan**.
- La ausencia de uno debilita a los demás.
- Cada valor **sostiene un aspecto clave del empirismo**:
  - **Apertura** genera **Transparencia**.
  - **Foco** permite **Inspección efectiva**.
  - **Coraje y Compromiso** impulsan la **Adaptación**.
  - **Respeto** mantiene la **colaboración y confianza**.
- Un equipo Scrum saludable muestra **equilibrio entre los cinco valores**.



Los valores como un sistema vivo

- Un equipo comprometido (Compromiso) se atreve a hablar de errores (Coraje) porque hay respeto y apertura.
- Cuando el foco se mantiene, la transparencia aumenta: todos saben hacia dónde van.
- Si el respeto desaparece, la apertura se convierte en riesgo.
- Cuando hay coraje, el equipo puede adaptarse rápido sin miedo a equivocarse.
- Un Scrum Master que protege estos vínculos fortalece la cultura Scrum cada Sprint.



Los valores como un sistema vivo

- Si tuvieran que elegir **un valor** de Scrum que mantuviera vivos a los demás, ¿cuál sería y por qué?

Cultura Scrum: cómo se construye

- Es el **conjunto de comportamientos, actitudes y hábitos** que hacen visible la agilidad en el día a día.
- Se **construye Sprint a Sprint**, a través de la coherencia entre valores y acciones.
- No depende solo del Scrum Master, sino del compromiso de **todo el equipo y la organización.**
- La cultura se fortalece cuando:
  - Se prioriza el aprendizaje sobre el control.
  - Se valoran los resultados colectivos sobre los logros individuales.
  - Se promueve la experimentación y la confianza.

Cultura Scrum: cómo se construye

🧩 En el equipo

- Celebrar los aprendizajes tanto como los resultados.
- Cumplir acuerdos y respetar los compromisos.
- Mantener la Definition of Done visible y actualizada.
- Tratar los errores como oportunidades de mejora.
- Evaluar procesos y relaciones en cada retrospectiva.

🧠 Desde el rol del Scrum Master

- Predicar con el ejemplo: escuchar, respetar y facilitar conversaciones difíciles.
- Reforzar los valores Scrum cuando se rompen (por ejemplo, intervenir ante falta de respeto).
- Fomentar un entorno de seguridad psicológica.
- Mostrar coherencia entre palabras y acciones.
- Facilitar el aprendizaje continuo del equipo.

🏢 A nivel organizacional

- Los líderes apoyan la autogestión y no intervienen en cada decisión.
- La comunicación es abierta, no jerárquica.
- Se mide el éxito por valor entregado, no por control o cumplimiento.
- La organización protege el tiempo del equipo para reflexionar y mejorar.
- Se promueven comportamientos colaborativos como modelo cultural.

Discusión grupal: ¿Qué comportamiento muestra (o no) un verdadero Scrum Master?

- Miro ….
- ¿Qué comportamientos *muestran* un verdadero Scrum Master?
- ¿Qué comportamientos *no lo representan*?

Discusión grupal: ¿Qué comportamiento muestra (o no) un verdadero Scrum Master?

✅ **Muestra un verdadero Scrum Master**

- Escucha activamente y facilita sin imponer.
- Protege al equipo de distracciones externas.
- Refuerza los valores Scrum cuando se ven amenazados.
- Promueve la mejora continua a través de retros efectivas.
- Cuestiona con respeto prácticas que no aportan valor.
- Da feedback constructivo y acepta el propio.
- Celebra los logros del equipo, no los individuales.

❌ **No representa a un Scrum Master**

- Actúa como jefe o coordinador de tareas.
- Usa el Daily como control de estatus.
- Toma decisiones por el equipo sin consulta.
- Ignora los impedimentos o los delega siempre a otros.
- Evita los conflictos difíciles.
- Habla más de procesos que de personas.
- No respeta los valores cuando hay presión externa.



Reflexión: valores en acción

- Los valores de Scrum se **demuestran con acciones**, no con palabras.
- Cada valor tiene un **impacto directo** en la cultura y en la confianza del equipo.
- La madurez de un equipo Scrum se refleja en cómo **vive los valores bajo presión.**



Reflexión: valores en acción

- ¿Qué valor practico con mayor naturalidad?
- ¿Qué valor necesito reforzar como profesional y como miembro del equipo?



Conclusiones del Módulo 2

- Los **valores de Scrum** (Compromiso, Foco, Apertura, Respeto y Coraje) son el corazón del marco.
- Sin valores, Scrum se convierte en un proceso vacío.
- La **cultura Scrum** se construye con coherencia entre lo que decimos y lo que hacemos.
- El **Scrum Master lidera con el ejemplo**, no con autoridad.
- La práctica diaria de los valores refuerza la **autogestión, la confianza y la mejora continua.**

Módulo 3. El Scrum Team y el Rol del Scrum Master

Scrum Master Profesional

Contenido del Curso

- **Módulo 1 - Fundamentos de la Agilidad y de Scrum**
  - El pensamiento ágil y el Manifiesto Ágil.
  - Contexto: complejidad, adaptabilidad y empirismo.
  - Diferencias entre la gestión tradicional y la ágil.
  - ¿Qué es Scrum y qué no es Scrum?
  - Los tres pilares de Scrum: transparencia, inspección y adaptación.
- **Módulo 2 – Valores y Cultura Scrum**
  - Compromiso, Foco, Apertura, Respeto y Coraje.
  - Cómo los valores sustentan la autogestión y la transparencia.
  - Conductas esperadas del Scrum Master para fomentar la cultura Scrum.



✨ ***El conocimiento cobra sentido cuando se convierte en acción con propósito.***

Scrum Master Profesional

Contenido del Curso

- **Módulo 3. El Scrum Team y el Rol del Scrum Master**
  - Estructura y responsabilidades del Scrum Team.
  - Qué hace y qué no hace un Scrum Master.
  - El liderazgo servicial: coaching, facilitación y mentoring.
  - Cómo eliminar impedimentos y promover la autogestión.
  - Ejemplos de disfunciones comunes (ScrumBut) y cómo corregirlas.



✨ ***El conocimiento cobra sentido cuando se convierte en acción con propósito.***

📘 Resumen del Módulo 2 — Valores y Cultura Scrum

- Scrum no funciona por sus eventos ni artefactos, sino porque **las personas viven sus valores**:
  Compromiso, **Foco**, **Apertura**, **Respeto** y **Coraje**.
- Los valores son **interdependientes**: si uno se debilita, todo el sistema pierde equilibrio.
- La **cultura Scrum** no se impone: **se construye Sprint a Sprint**, a través de la coherencia entre palabras y acciones.
- El Scrum Master es **modelo de comportamiento**: lidera con ejemplo, empatía y coherencia.
- Los valores se manifiestan en decisiones difíciles — cuando hay presión, conflicto o incertidumbre.



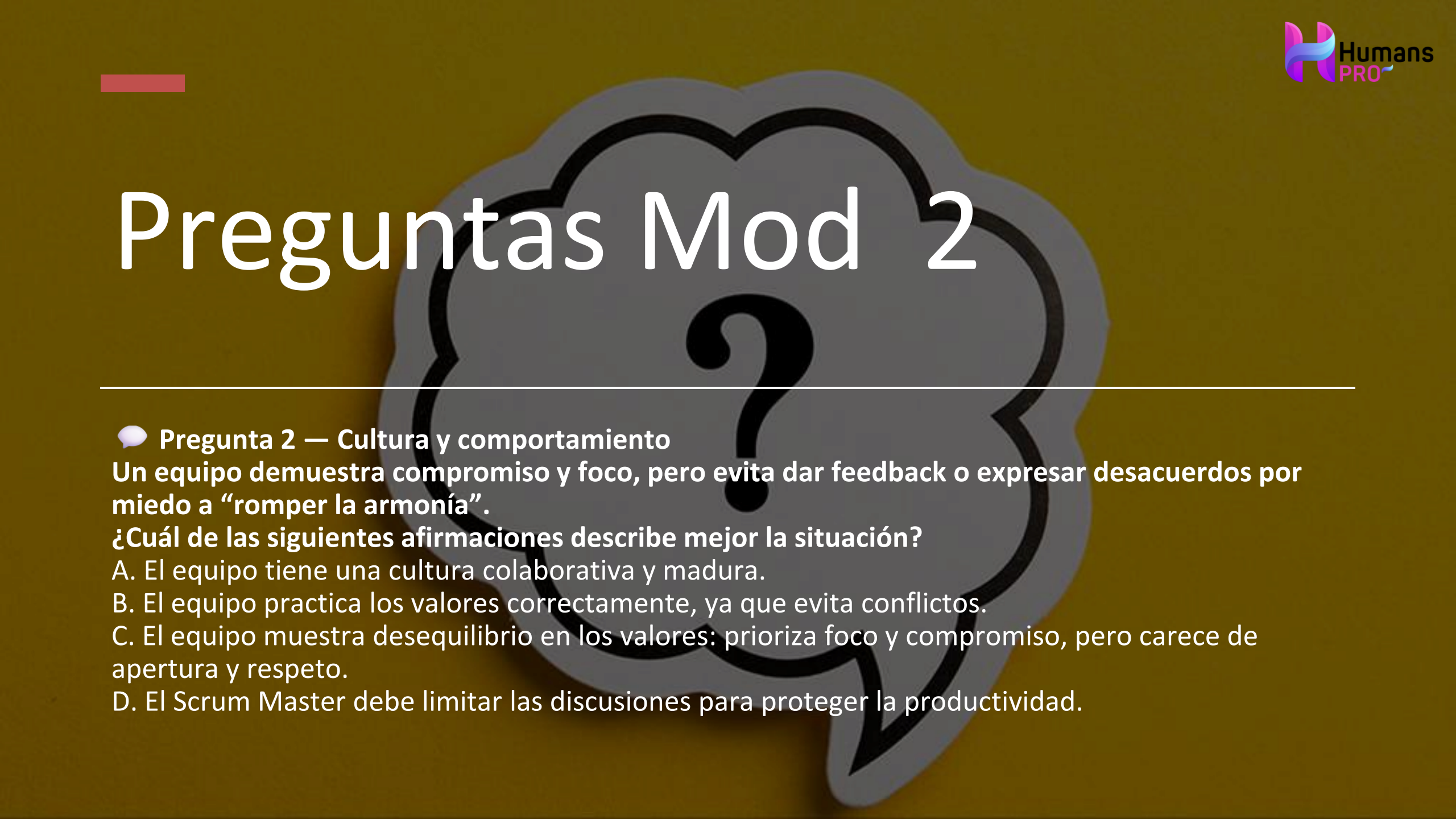
Preguntas Mod 2

🧠 **Pregunta 1 — Integración de valores**

Durante una retrospectiva, el equipo evita discutir un error reciente porque “no quieren generar tensión”.
¿Qué valor de Scrum está siendo comprometido y cuál debería reforzar el Scrum Master?

A. Compromiso — fomentando responsabilidad individual.
B. Coraje — promoviendo conversaciones difíciles con respeto.
C. Respeto — evitando la confrontación directa.
D. Foco — manteniendo la discusión solo en los resultados del Sprint.

Preguntas Mod 2



💬 **Pregunta 2 — Cultura y comportamiento**

Un equipo demuestra compromiso y foco, pero evita dar feedback o expresar desacuerdos por miedo a “romper la armonía”.

¿Cuál de las siguientes afirmaciones describe mejor la situación?

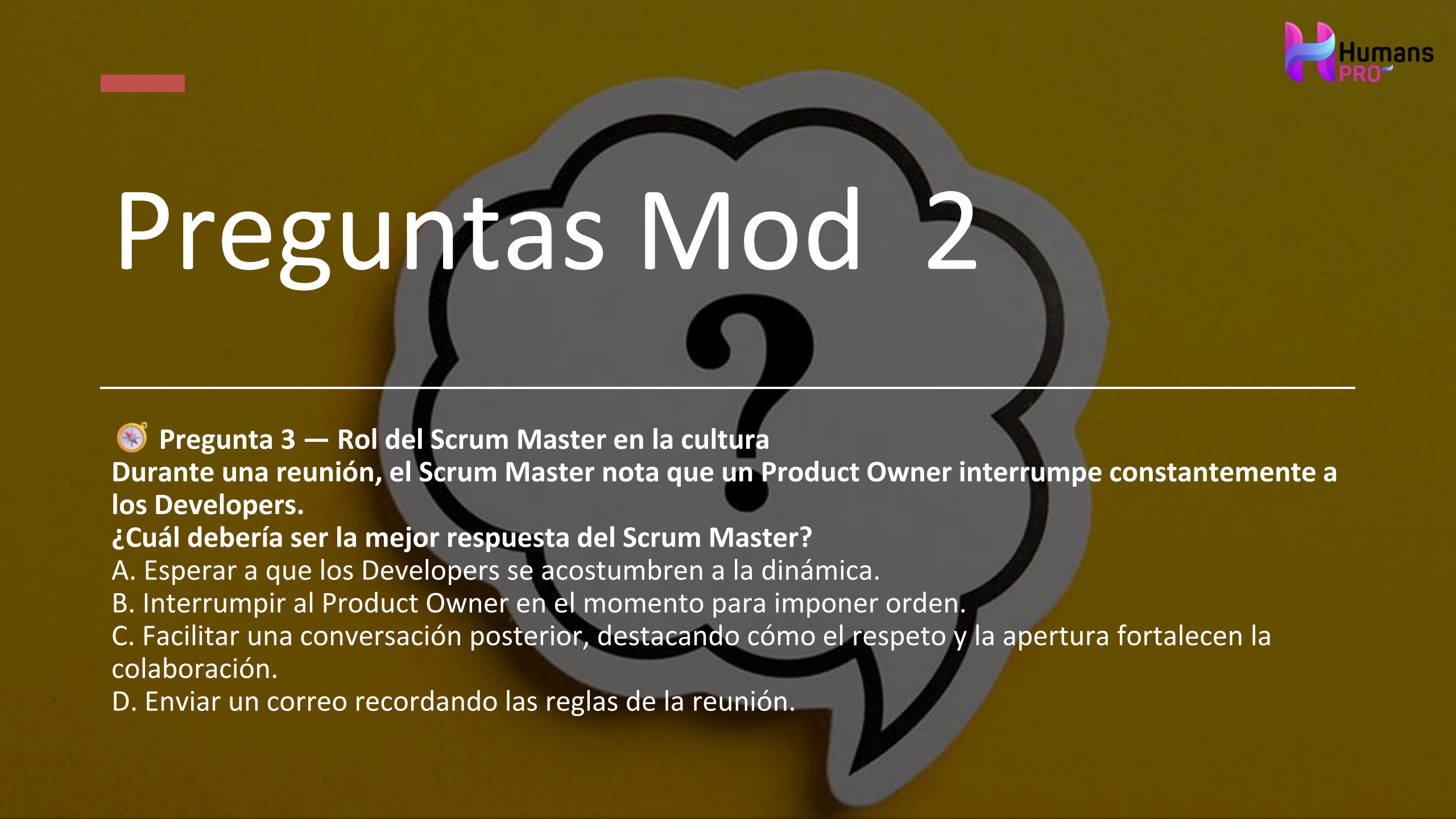
A. El equipo tiene una cultura colaborativa y madura.

B. El equipo practica los valores correctamente, ya que evita conflictos.

C. El equipo muestra desequilibrio en los valores: prioriza foco y compromiso, pero carece de apertura y respeto.

D. El Scrum Master debe limitar las discusiones para proteger la productividad.

Preguntas Mod 2



🧭 **Pregunta 3 — Rol del Scrum Master en la cultura**

Durante una reunión, el Scrum Master nota que un Product Owner interrumpe constantemente a los Developers.

¿Cuál debería ser la mejor respuesta del Scrum Master?

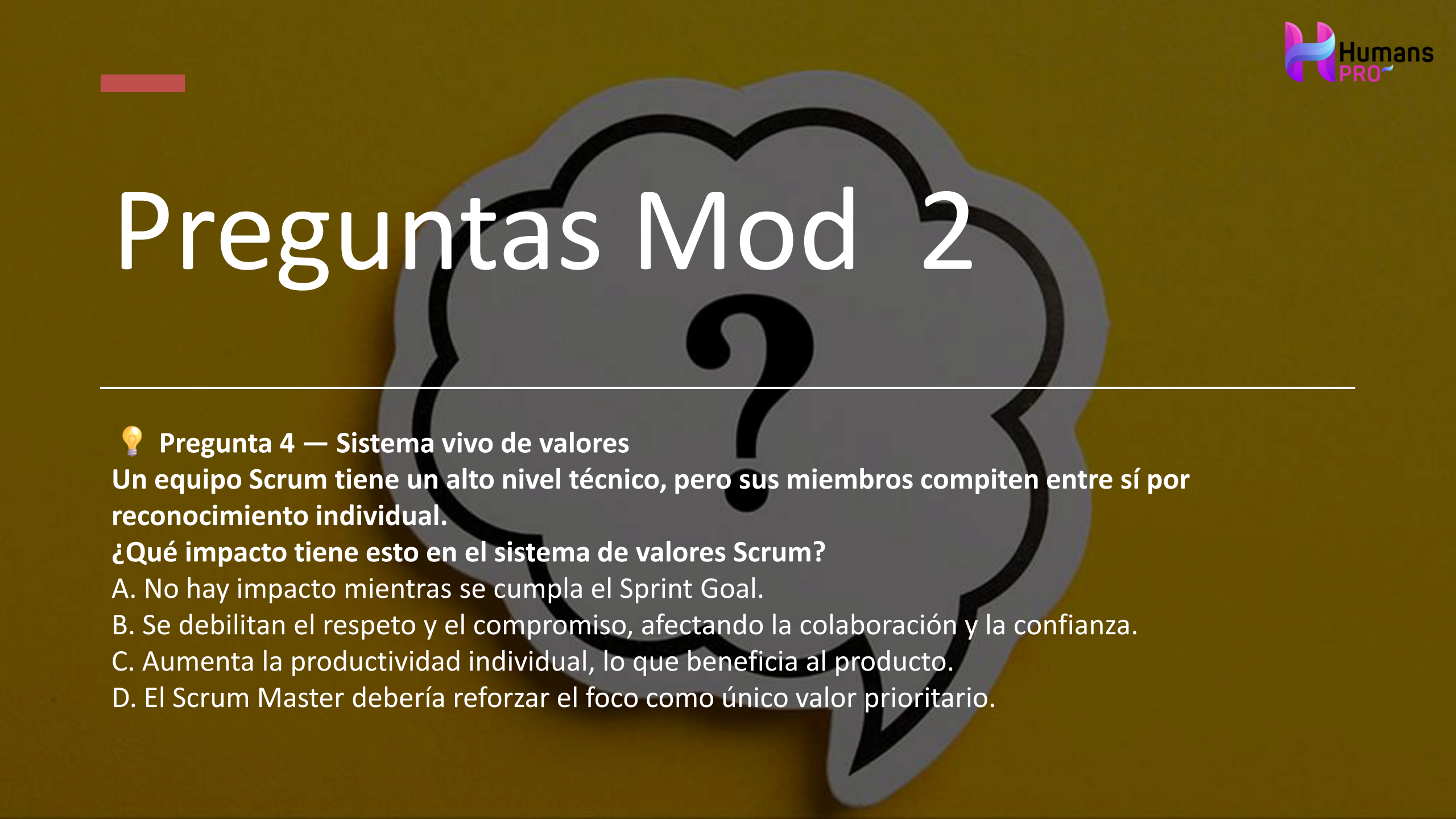
A. Esperar a que los Developers se acostumbren a la dinámica.

B. Interrumpir al Product Owner en el momento para imponer orden.

C. Facilitar una conversación posterior, destacando cómo el respeto y la apertura fortalecen la colaboración.

D. Enviar un correo recordando las reglas de la reunión.

Preguntas Mod 2



💡 **Pregunta 4 — Sistema vivo de valores**

Un equipo Scrum tiene un alto nivel técnico, pero sus miembros compiten entre sí por reconocimiento individual.

¿Qué impacto tiene esto en el sistema de valores Scrum?

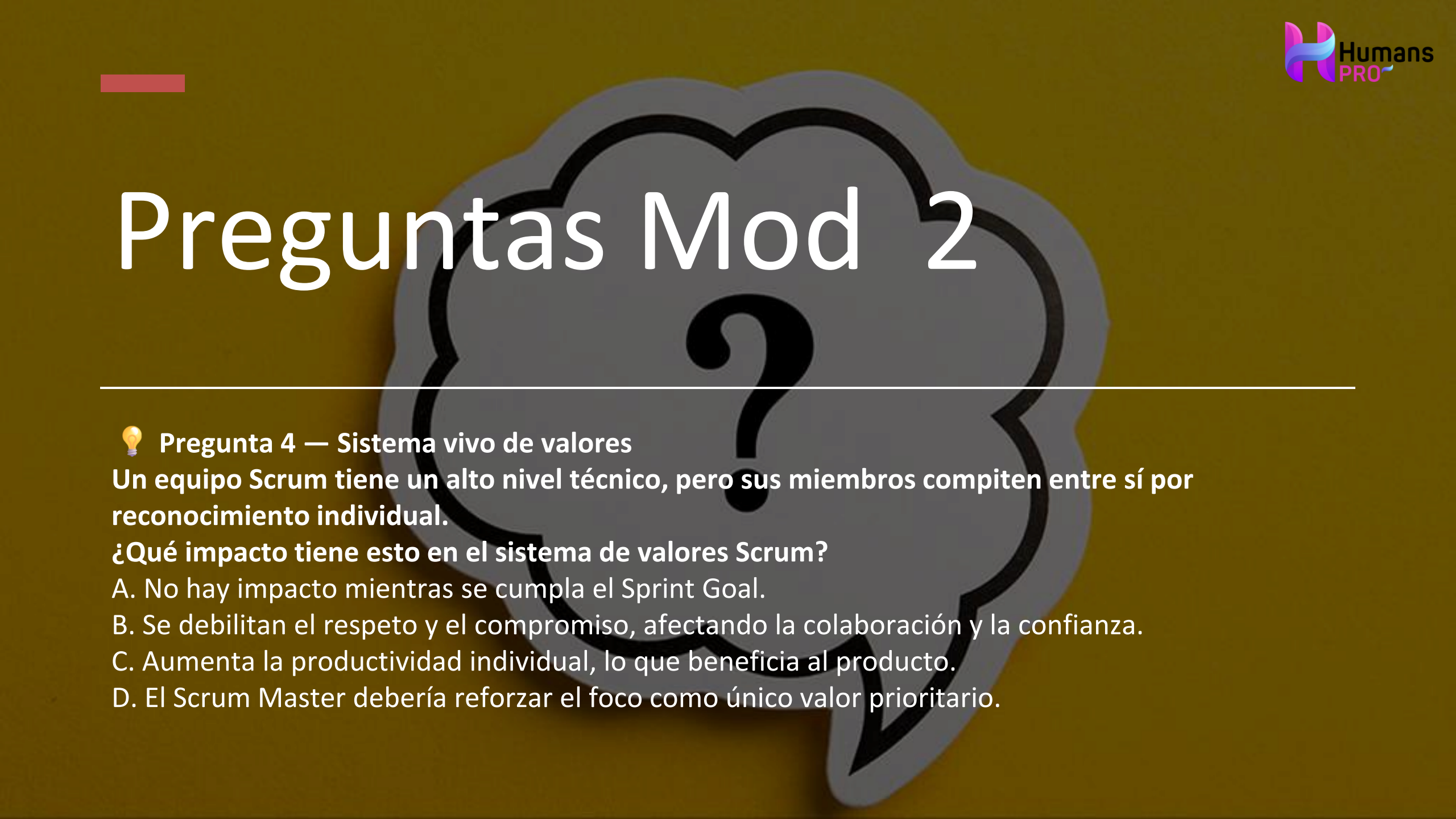
A. No hay impacto mientras se cumpla el Sprint Goal.

B. Se debilitan el respeto y el compromiso, afectando la colaboración y la confianza.

C. Aumenta la productividad individual, lo que beneficia al producto.

D. El Scrum Master debería reforzar el foco como único valor prioritario.

Preguntas Mod 2



💡 **Pregunta 4 — Sistema vivo de valores**

Un equipo Scrum tiene un alto nivel técnico, pero sus miembros compiten entre sí por reconocimiento individual.

¿Qué impacto tiene esto en el sistema de valores Scrum?

A. No hay impacto mientras se cumpla el Sprint Goal.

B. Se debilitan el respeto y el compromiso, afectando la colaboración y la confianza.

C. Aumenta la productividad individual, lo que beneficia al producto.

D. El Scrum Master debería reforzar el foco como único valor prioritario.

Preguntas Mod 2

🔥 **Pregunta 5 — Cultura Scrum en la organización**

Una empresa decide implementar Scrum, pero mantiene la aprobación jerárquica de decisiones, penaliza errores y recompensa la obediencia.

¿Cuál es el rol más crítico del Scrum Master en este contexto?

A. Garantizar que el equipo cumpla los eventos y artefactos al pie de la letra.

B. Adaptar Scrum para ajustarse a la estructura jerárquica existente.

C. Actuar como agente de cambio cultural, promoviendo transparencia, apertura y seguridad psicológica.

D. Escalar los conflictos al management para mantener la armonía.

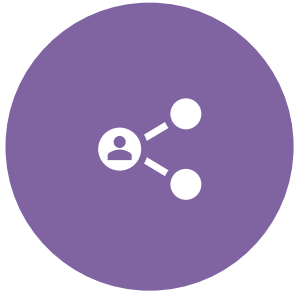
📘 Módulo 3: El Scrum Team y el Rol del Scrum Master



Estructura y responsabilidades del Scrum Team.



Qué hace y qué no hace un Scrum Master.

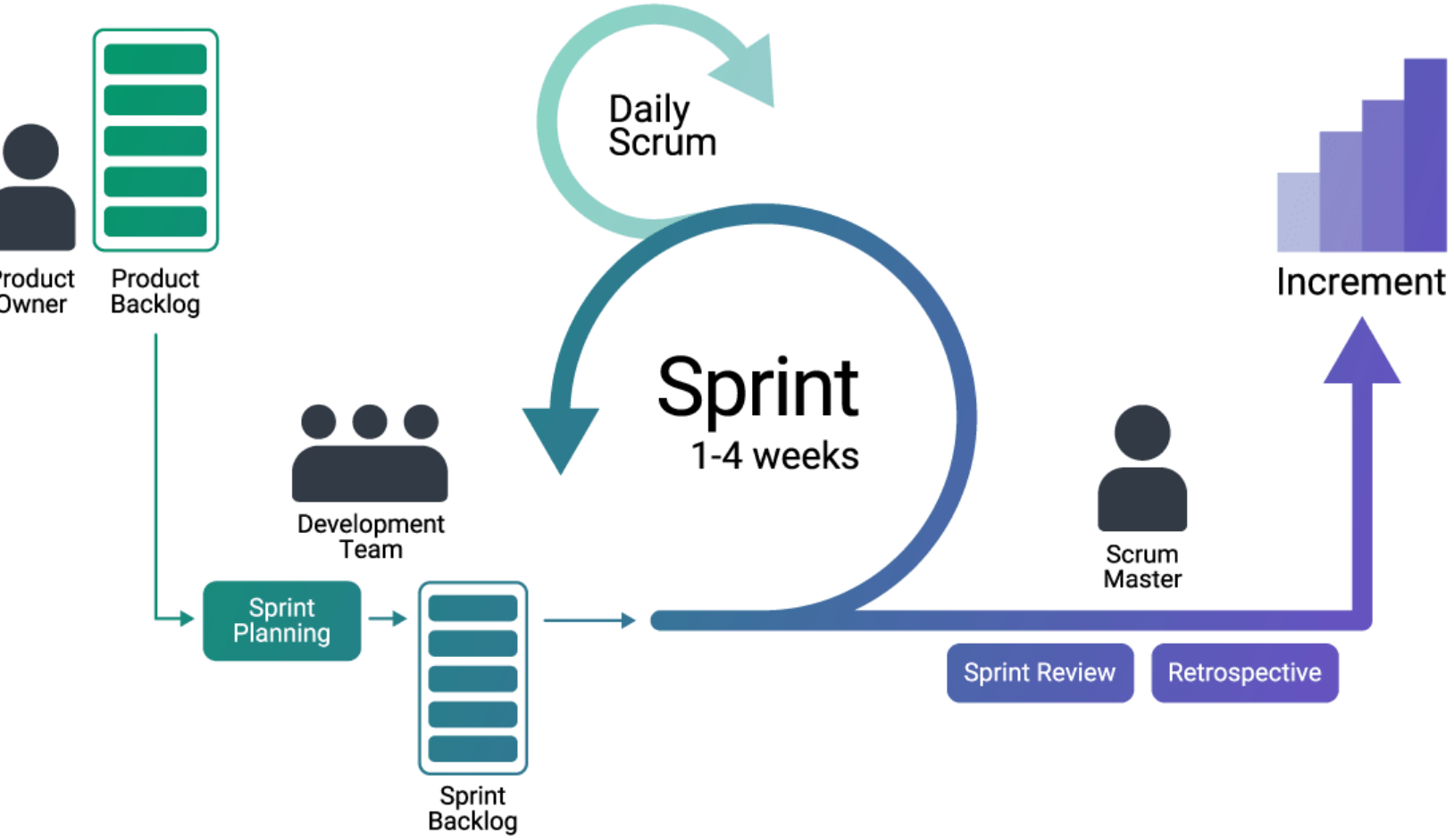


Liderazgo servicial: coaching, facilitación y mentoring.



Cómo eliminar impedimentos y promover la autogestión.

The Scrum Framework



👋 Bienvenida al Módulo 3

- Scrum Team: pequeño, multifuncional y autogestionado.
- Scrum Master: líder servidor, no jefe.
- El equipo es responsable colectivamente del resultado del Sprint.

Estructura del Scrum Team



- Tres roles: Product Owner, Scrum Master y Developers.
- Sin jerarquías internas ni títulos.
- Equipos pequeños: 10 personas o menos.
- Enfocados en la entrega de valor en cada Sprint.

🤝 Responsabilidades compartidas

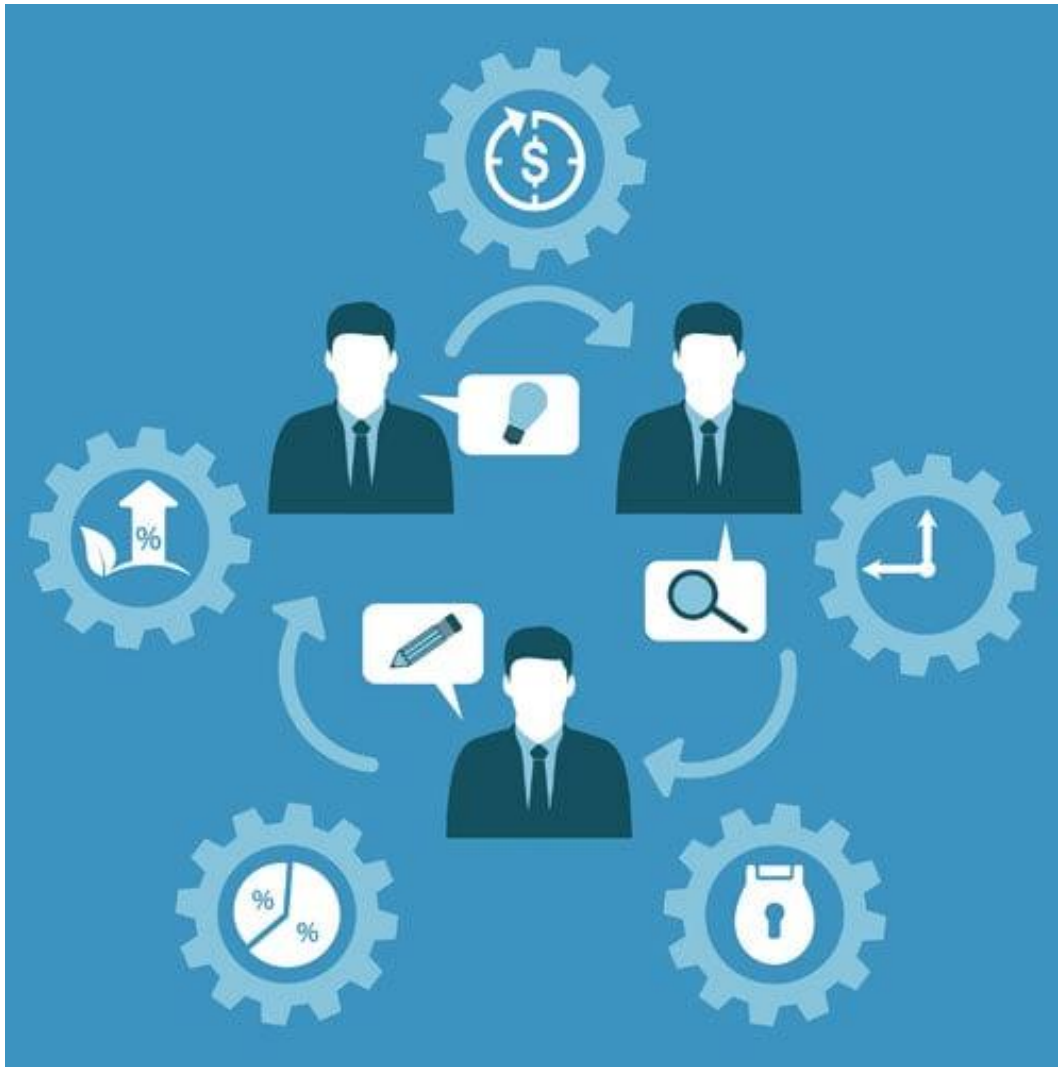
- Todos los miembros comparten la meta del Sprint.
- Todos contribuyen a la mejora continua.
- Todos son responsables del resultado del Incremento.
- No hay 'mi tarea' vs. 'tu tarea'.





🎯 El Product Owner: visión y valor

- Representa la voz del cliente.
- Define y prioriza el Product Backlog.
- Se enfoca en maximizar el valor del producto.
- Colabora con el Scrum Master y los Developers.



💻 Los Developers: creación y colaboración

- Crean el Incremento de producto.
- Son multifuncionales, colaborativos y autoorganizados.
- Gestionan su propio trabajo y calidad.
- Aseguran transparencia en los artefactos de Scrum.



SCRUM MASTER

🧭 El Scrum Master: líder servidor

- Promueve y apoya Scrum según la Guía.
- Lidera sirviendo al equipo y la organización.
- Facilita eventos y elimina impedimentos.
- Enseña agilidad a través del ejemplo.

⚙️ Qué hace un Scrum Master

- Facilita los eventos de Scrum efectivamente.
- Elimina impedimentos que bloquean el progreso.
- Ayuda al PO a definir objetivos claros.
- Promueve la mejora continua.
- Educa a la organización sobre Scrum.





🚫 Qué NO hace un Scrum Master

- No es jefe del equipo ni asigna tareas.
- No toma decisiones por los Developers.
- No controla horarios ni microgestiona.
- No convierte reuniones en formalidades vacías.



🚫 Qué NO hace un Scrum Master

🚫 Malas prácticas	✅ Alternativas efectivas
Hace la Daily solo para recibir reportes.	Facilita que el equipo converse y planifique su día.
Decide qué tareas deben hacerse.	Promueve que el equipo gestione su propio Sprint Backlog.
Habla todo el tiempo en las reuniones.	Escucha y fomenta la participación de todos.
Evita los conflictos.	Facilita conversaciones seguras para resolverlos.
Cubre errores o problemas.	Fomenta la apertura y el aprendizaje.



❤️ Liderazgo servicial en acción

- Liderar sirviendo, no dirigiendo.
- Escuchar antes de actuar.
- Facilitar, no imponer.
- Empoderar, no controlar.
- Inspirar confianza con coherencia.

🎓 Coaching, Facilitación y Mentoring

- Coaching: guiar al equipo a descubrir sus propias respuestas.
- Facilitación: crear espacios seguros para la toma de decisiones.
- Mentoring: compartir experiencia cuando el equipo necesita guía.
- Saber cuándo aplicar cada enfoque es clave.

Situación	Enfoque adecuado	Ejemplo de acción
El equipo evita hablar de un conflicto.	🧭 Coaching	Formular preguntas que lleven a la reflexión (“¿Qué nos impide hablar de esto?”).
La planificación se desvía del objetivo.	🤝 Facilitación	Redirigir la conversación con técnicas visuales o límite de tiempo.
Un nuevo Product Owner no entiende cómo priorizar.	💡 Mentoring	Compartir experiencias previas y buenas prácticas para guiarlo.

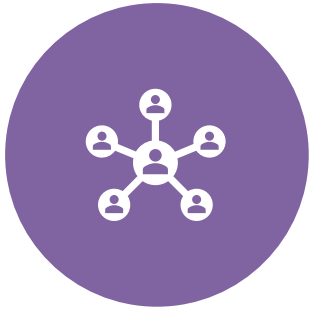
🎓 Coaching, Facilitación y Mentoring



• COACHING: GUIAR AL EQUIPO A DESCUBRIR SUS PROPIAS RESPUESTAS.



• FACILITACIÓN: CREAR ESPACIOS SEGUROS PARA LA TOMA DE DECISIONES.



• MENTORING: COMPARTIR EXPERIENCIA CUANDO EL EQUIPO NECESITA GUÍA.



• SABER CUÁNDO APLICAR CADA ENFOQUE ES CLAVE.

🎓 Coaching, Facilitación y Mentoring

Situación	Enfoque adecuado	Ejemplo de acción
El equipo evita hablar de un conflicto.	🧭 Coaching	Formular preguntas que lleven a la reflexión (“¿Qué nos impide hablar de esto?”).
La planificación se desvía del objetivo.	🤝 Facilitación	Redirigir la conversación con técnicas visuales o límite de tiempo.
Un nuevo Product Owner no entiende cómo priorizar.	💡 Mentoring	Compartir experiencias previas y buenas prácticas para guiarlo.



🧱 Eliminando impedimentos

- Detectar obstáculos visibles y ocultos.
- Priorizar los que más afectan al valor.
- Gestionar con transparencia los bloqueos externos.
- Enseñar al equipo a resolver sus propios impedimentos.

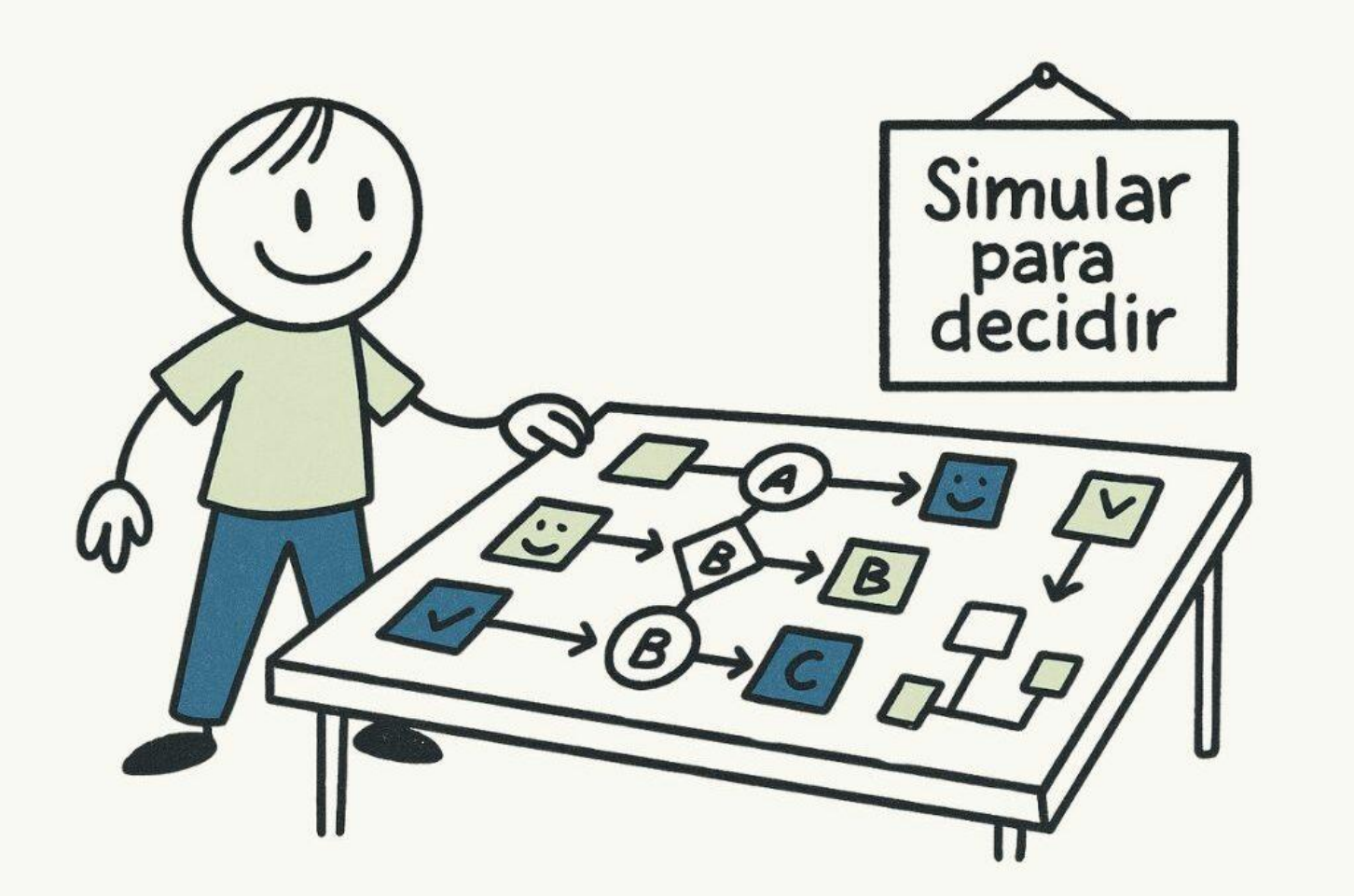


🧱 Eliminando impedimentos

🚧 Impedimento	🎯 Causa probable	🧭 Acción del Scrum Master
Dependencia externa (espera de aprobación, equipo de QA, etc.)	Procesos lentos o silos organizacionales.	Facilita acuerdos de servicio o sincronizaciones entre equipos.
Falta de claridad en requerimientos.	Comunicación deficiente con el PO o stakeholders.	Promueve refinamientos colaborativos y visualización del backlog.
Sobrecarga o multitarea.	Cultura de urgencias.	Refuerza la importancia del foco y del Sprint Goal.
Falta de participación en eventos.	Desmotivación o falta de propósito.	Explora causas en retrospectivas y celebra logros pequeños.

1. Introducción a un caso del módulo

- Validaremos dinámicas reales del día a día de un Scrum Master.
- Observaremos antipatrones (malas prácticas comunes) y cómo corregirlos.
- Hablaremos de la **facilitación**, la **gestión de conflictos** y el **liderazgo servicial** en acción.
- Aprenderemos a **detectar disfunciones (ScrumBut)** y transformarlas en oportunidades de mejora.



⚠️ 2. Simulación: Daily Scrum con anti-patrones

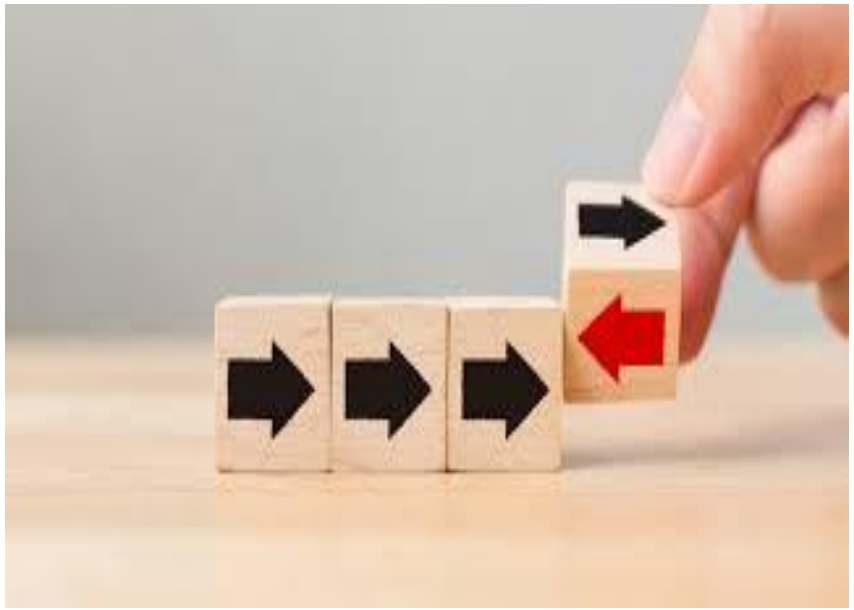
Anti-patrón	Consecuencia
Se reporta al Scrum Master, no al equipo.	Pérdida de autogestión.
Las conversaciones se alargan en temas técnicos.	Se rompe el foco y el time-box.
Algunos miembros no participan.	Falta de transparencia.
No se menciona el Sprint Goal.	La reunión se vuelve una lista de tareas.
Se buscan culpables.	Se pierde la confianza y la mejora continua.

🧩 3. Qué son los anti-patrones

- Un **comportamiento o práctica** que parece ayudar, pero **en realidad daña el propósito de Scrum.**
- Surge como **atajo o mala interpretación** del marco.
- Refleja **problemas culturales o estructurales** más profundos (falta de confianza, miedo, presión).
- El rol del Scrum Master es **hacerlos visibles y transformarlos** en aprendizaje.



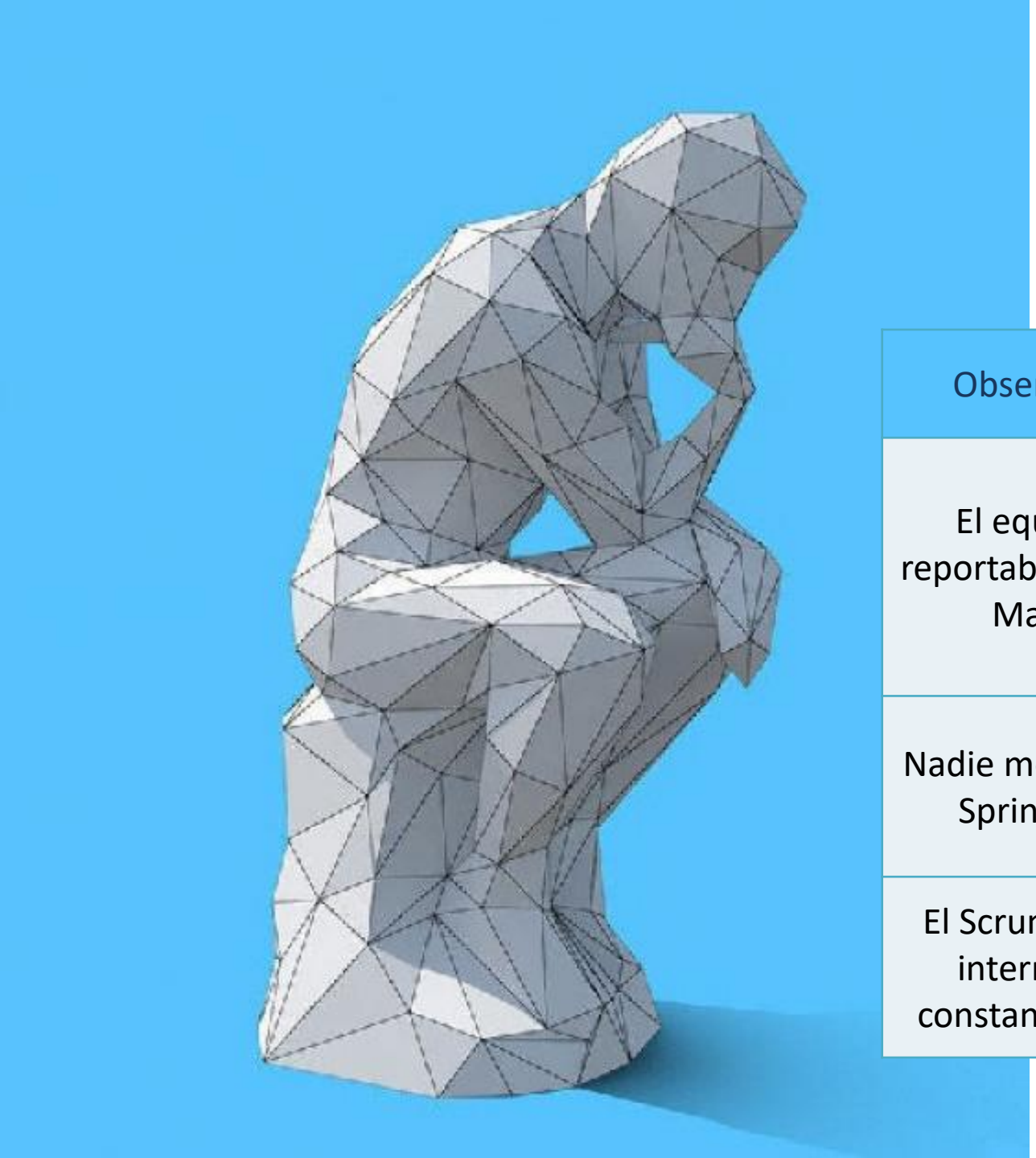
⏱️ 4. Ejemplos de anti-patrones en la Daily



Anti-patrón	Efecto negativo
Reporte al Scrum Master	Se pierde la autogestión.
Monólogo del PO	El equipo se desconecta.
Falta de foco	Se extiende y se vuelve irrelevante.
Sin inspección del progreso	No se detectan desvíos.

⏱️ 4. Ejemplos de anti-patrones en la Daily

⚠️ Anti-patrón	🔍 Qué lo causa	🚫 Efecto negativo
Se reporta al Scrum Master, no al equipo.	Cultura jerárquica o miedo.	Se pierde la autogestión y la transparencia.
El Daily se convierte en seguimiento de tareas.	Foco en control, no en colaboración.	Falta conexión con el Sprint Goal.
Conversaciones técnicas extensas.	Falta de timeboxing o foco.	Reuniones largas, pérdida de energía.
Algunos miembros no participan.	Desmotivación o falta de propósito.	Invisibilidad de bloqueos reales.
Se ignora el Sprint Goal.	Rutina sin contexto.	La reunión se vuelve irrelevante.
El Scrum Master lidera siempre.	Falta de madurez del equipo.	Dependencia del facilitador.



🧠 5. Reflexión post-situacion

Observación	Causa raíz	Acción del Scrum Master
El equipo se reportaba al Scrum Master.	Cultura jerárquica o falta de confianza.	Reforzar el propósito del Daily y fomentar la conversación entre pares.
Nadie mencionó el Sprint Goal.	Pérdida de foco o desconocimiento.	Recordar el objetivo y facilitar la conexión con él.
El Scrum Master interrumpía constantemente.	Deseo de control o ansiedad.	Practicar escucha activa y permitir la autogestión.



🧠 5. Reflexión post-situacion

- ¿Qué comportamientos ayudarían al equipo?
- ¿Qué comportamientos podrían dificultar la colaboración o la transparencia?
- ¿Qué emociones podrían surgir durante la simulación (frustración, confusión, silencio, etc.)?
- ¿Qué habría hecho diferente un Scrum Master en situaciones como las que hemos hablado?
- ¿Qué señales de disfunción pueden detectarse en reuniones reales?



🎭 6. Historia: Scrum Master ante un Product Owner difícil

- 🧭 **Scrum Master** — busca proteger el equipo y facilitar una conversación efectiva.
- 🎯 **Product Owner difícil** — presionado, quiere agregar trabajo urgente.
- 👥 **Observadores** — analizan las conductas, el lenguaje y las emociones de ambos.



🗣️ 7. Contexto del conflicto

- El **Product Owner** está bajo presión de negocio: un cliente importante pide una nueva funcionalidad urgente.
- El **Scrum Master** intenta proteger la estabilidad del Sprint y el compromiso del equipo.
- Los **Developers** están confundidos: quieren colaborar, pero temen perder el foco y la calidad.

📜 8. Contando la historia

- El PO pide trabajo adicional urgente.
- El Scrum Master debe responder desde el liderazgo servicial.
- El equipo observa e identifica:
  - Cómo se aplican los valores Scrum.
  - Qué tan efectiva es la comunicación.

💡 9. Feedback y aprendizaje del Role Play

- ¿Qué hizo bien el **Scrum Master** para manejar la situación?
- ¿Qué acciones del **Product Owner** fueron constructivas o contraproducentes?
- ¿Qué emociones creen que se percibieron durante el este sprint?
- ¿Cómo se manifestó (o no) el **liderazgo servicial**?
- ¿Qué habría podido mejorar la situación?

🧱 10. Disfunciones comunes (ScrumBut)

🧱 “ScrumBut”	💭 Racionalización común	⚠️ Riesgo real
“Hacemos Daily, pero solo tres veces por semana.”	“No siempre hay cosas nuevas que decir.”	Se pierde inspección y adaptación diaria.
“Hacemos retrospectivas cuando hay tiempo.”	“Ya nos conocemos bien.”	Desaparece la mejora continua.
“El Product Owner no puede asistir a las Reviews.”	“Tiene otras prioridades.”	Se rompe la colaboración y feedback.
“No usamos Definition of Done porque todos confían.”	“El equipo es maduro.”	Se pierde transparencia y calidad.
“Hacemos Scrum, pero sin Scrum Master.”	“No necesitamos facilitador.”	Nadie vela por el marco ni por los valores.

🔧 11. Cómo corregir las disfunciones

Situación	Causa probable	Acción del Scrum Master	Valor Scrum a reforzar
“Hacemos retrospectivas cada dos Sprints.”	Falta de tiempo o bajo valor percibido.	Facilitar una retro breve pero enfocada en resultados.	Compromiso
“El PO no asiste a Reviews.”	Falta de involucramiento o exceso de tareas.	Conversar sobre impacto y acordar su presencia como prioridad.	Apertura
“No usamos Definition of Done.”	Desconocimiento o exceso de confianza.	Co-crear una DoD mínima y visible.	Transparencia

12. Cierre y Reflexión Final



LO QUE VIMOS HOY: LIDERAZGO, OBSERVACIÓN Y MEJORA.



EL SCRUM MASTER ACTÚA MÁS COMO ENTRENADOR QUE COMO JEFE.

Módulo 4. Los Eventos de Scrum



Scrum Master Profesional

Contenido del Curso (1/2)

- **Módulo 1 - Fundamentos de la Agilidad y de Scrum**
  - El pensamiento ágil y el Manifiesto Ágil.
  - Contexto: complejidad, adaptabilidad y empirismo.
  - Diferencias entre la gestión tradicional y la ágil.
  - ¿Qué es Scrum y qué no es Scrum?
  - Los tres pilares de Scrum: transparencia, inspección y adaptación.
- **Módulo 2 – Valores y Cultura Scrum**
  - Compromiso, Foco, Apertura, Respeto y Coraje.
  - Cómo los valores sustentan la autogestión y la transparencia.
  - Conductas esperadas del Scrum Master para fomentar la cultura Scrum.



✨ ***El conocimiento cobra sentido cuando se convierte en acción con propósito.***

Scrum Master Profesional

Contenido del Curso (2/2)

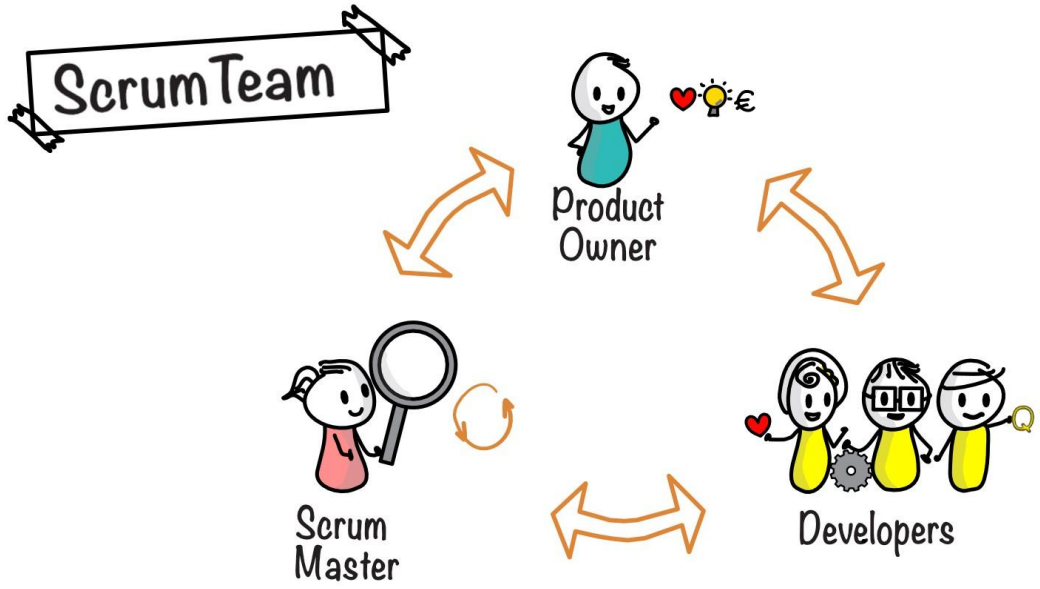
- **Módulo 3. El Scrum Team y el Rol del Scrum Master**
  - Estructura y responsabilidades del Scrum Team.
  - Qué hace y qué no hace un Scrum Master.
  - El liderazgo servicial: coaching, facilitación y mentoring.
  - Cómo eliminar impedimentos y promover la autogestión.
  - Ejemplos de disfunciones comunes (ScrumBut) y cómo corregirlas.
- **Módulo 4. Los Eventos de Scrum**
  - El Sprint como contenedor de todo.
  - Sprint Planning: propósito, enfoque en el “por qué”, “qué” y “cómo”.
  - Daily Scrum: autogestión, foco y visibilidad del progreso.
  - Sprint Review: transparencia y colaboración con stakeholders.
  - Sprint Retrospective: mejora continua y experimentación.
  - Time-boxing y cómo mantener eventos productivos.

✨ ***El conocimiento cobra sentido cuando se convierte en acción con propósito.***



Resumen del Módulo 3 — Roles y Liderazgo del Scrum Master

- El **Scrum Team** como unidad multifuncional y autogestionada.
- Responsabilidades de cada rol: PO, Developers y Scrum Master.
- El **Scrum Master** como líder servidor, no jefe.
- Coaching, facilitación y mentoring: tres formas de acompañar.
- Cómo detectar y manejar **anti-patrones** y disfunciones (ScrumBut).
- Conflictos como oportunidades de crecimiento.



🧱 1. Definition of Ready (DoR) y Definition of Done (DoD)

✔️ Asegura que un ítem del Product Backlog está **claramente entendido** y listo para ser trabajado.
✔️ Evita arrastrar ambigüedades al Sprint.
✔️ Reduce interrupciones y retrabajo.

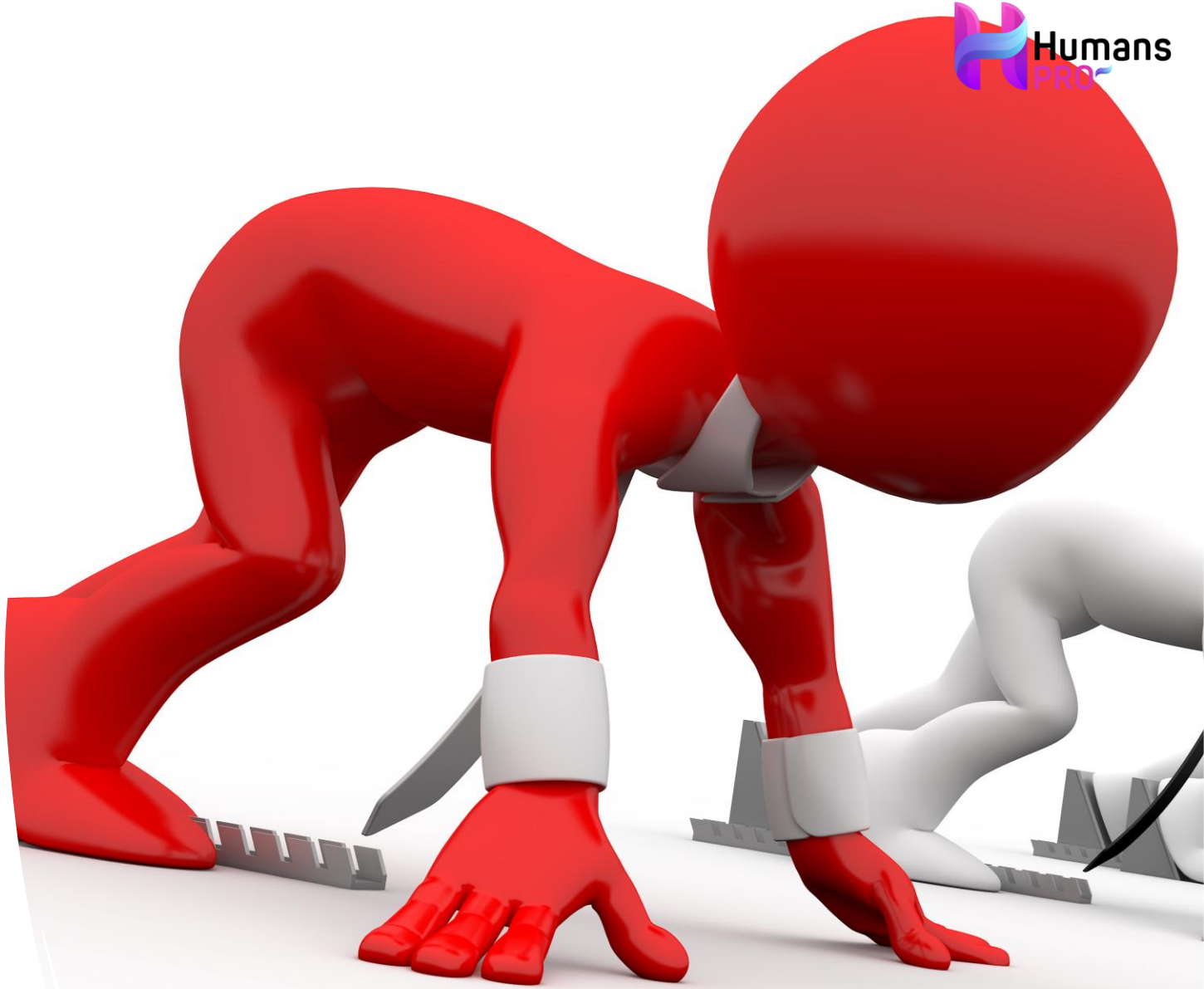


🧱 1. Definition of Ready (DoR) y Definition of Done (DoD)

✔️ Asegura que un ítem del Product Backlog está **claramente entendido** y listo para ser trabajado.

✔️ Evita arrastrar ambigüedades al Sprint.

✔️ Reduce interrupciones y retrabajo.



🧱 1. Definition of Ready (DoR) y Definition of Done (DoD)

✔️ Alinea el estándar de **calidad mínima** para considerar un Incremento “terminado”.
✔️ Fomenta transparencia y calidad consistente.
✔️ Permite inspeccionar un Incremento real, no “casi terminado”.



🧱 1. Definition of Ready (DoR) y Definition of Done (DoD)

✔️ Alinea el estándar de **calidad mínima** para considerar un Incremento “terminado”.
✔️ Fomenta transparencia y calidad consistente.
✔️ Permite inspeccionar un Incremento real, no “casi terminado”.



🧠 PREGUNTA 1

En un equipo nuevo en Scrum, los Developers esperan que el Scrum Master les asigne tareas porque “así trabajaban antes”.
Cuando el Scrum Master intenta que sean ellos quienes organicen su trabajo, se genera silencio e incomodidad.

- **¿Cuál debería ser la mejor intervención del Scrum Master?**

A. Asignar tareas temporalmente mientras el equipo se adapta.
B. Insistir en que ellos deben autogestionarse sin ofrecer apoyo adicional.
C. Facilitar una conversación sobre autogestión y acompañar con coaching para que descubran cómo organizarse.
D. Pedir al Product Owner que decida cómo dividir el trabajo.



🧠 PREGUNTA 2

Durante las últimas semanas, el Scrum Master revisa cada tarea antes de que los Developers la marquen como “Done”, diciendo que es “para evitar errores”.
El equipo empieza a depender de él para validar calidad, perdiendo autonomía.

¿Qué describe mejor esta situación?

A. El Scrum Master está actuando correctamente al proteger la calidad.
B. El equipo es débil y debe ser supervisado.
C. El Scrum Master está rompiendo la autogestión y actuando como un mini-jefe.
D. El Scrum Master debería aprobar todas las tareas para mantener consistencia.



🧠 PREGUNTA 3

Un nuevo Product Owner está luchando para priorizar el backlog.
Quiere apoyo del Scrum Master, pero le pide: *“Dime tú qué historias debo subir de prioridad.”*

¿Qué enfoque debería usar el Scrum Master?

A. Decirle exactamente cómo ordenar el backlog.
B. Hacer coaching para que él descubra cómo priorizar por valor.
C. Facilitar que el equipo vote las prioridades.
D. Negarse a ayudar para no interferir en su rol.



🧠 PREGUNTA 4

Un equipo dice: “Hacemos Scrum, *pero* no hacemos retrospectivas porque estamos muy ocupados y ya sabemos qué mejorar.”
Al mismo tiempo, los mismos problemas se repiten Sprint tras Sprint.

¿Cuál es el principal riesgo de este ScrumBut?

A. Que el equipo pierda velocidad.
B. Que el Product Owner pierda control sobre el backlog.
C. Que desaparezca la mejora continua y los problemas se cronifiquen.
D. Que se reduzca la documentación del proyecto.



🧠 PREGUNTA 4

El Product Owner quiere agregar un ítem urgente durante el Sprint “porque negocio lo necesita ya”.
El Scrum Master explica que eso rompe la estabilidad del Sprint, pero el PO insiste.
El ambiente se vuelve tenso y el equipo mira en silencio.

¿Cuál debería ser el siguiente paso del Scrum Master?

A. Aceptar el cambio para evitar conflicto.
B. Prohibir el cambio y exigir seguir el plan inicial.
C. Facilitar una conversación basada en el Sprint Goal, el impacto del cambio y alternativas colaborativas (como esperar la Planning o cancelar el Sprint).
D. Pedir al equipo que vote democráticamente.



🚀 2. El Sprint: el contenedor de todo

- **El Sprint es…**
- El **ciclo fijo** donde ocurre todo en Scrum.
- Un período de **tiempo constante** (1–4 semanas).
- El espacio donde el equipo entrega un **Incremento usable y valioso**.
- Un ciclo basado en **transparencia, inspección y adaptación**.



🚀 2. El Sprint: el contenedor de todo

- **El Sprint no se cancela salvo casos extremos**
- Solo el Product Owner puede cancelarlo y únicamente si el Sprint Goal deja de tener sentido.
- **Límites del Sprint**
- No se cambian objetivos a mitad de Sprint.
- No se cambia la duración del Sprint.
- No se agregan fases dentro del Sprint.



Sprint Planning — el punto de partida

✔ Alinear al equipo en **por qué**, **qué** y **cómo** trabajarán durante el Sprint.
✔ Establecer un **Sprint Goal claro y motivador**.
✔ Crear un plan realista y colaborativo para entregar valor.



Sprint Planning — el punto de partida

- **¿Por qué es valioso este Sprint?** → *Sprint Goal*
- **¿Qué podemos entregar?** → *Selección del Product Backlog*
- **¿Cómo lo haremos?** → *Plan de trabajo de los Developers*





🎯 El propósito del Sprint Goal

- **¿Para qué sirve el Sprint Goal?**

✔ Da **dirección** al equipo durante todo el Sprint.
✔ Proporciona **foco compartido**, más allá de tareas individuales.
✔ Ayuda a tomar decisiones bajo presión:

- “¿Esto contribuye al Sprint Goal?”

✔ Aumenta la **autonomía**: permite adaptarse sin perder el propósito.
✔ Evita Sprints que se convierten en una lista de tareas inconexas.



🎯 El propósito del Sprint Goal

🟢 **Bueno:**
“Reducir en un 20% el tiempo de carga de la página de inicio.”
“Permitir a usuarios registrarse usando cuentas de Google.”
🔴 **Malo:**
“Trabajar en tickets del backlog.”
“Hacer mantenimiento.”
“Implementar historias 123, 124 y 128.”

S	M	A	R	T
ESPECÍFICO	MEDIBLE	ALCANZABLE	RELEVANTE	TEMPORAL
Muy concreto. No hay cabida a dobles interpretaciones	Nunca debemos definir objetivos que no podamos medir.	El equilibrio ideal entre posible y retador.	En línea con los objetivos globales de la compañía.	Englobados dentro de un marco temporal específico.
G	O	A	L	S


S
Aumentar las visitas de nuestro sitio web en un 50%
M
50% (de 1,000 a 1,500)
A
Duplicando la distribución de contenido
R
Prepararnos para el lanzamiento de nuestro nuevo producto
T
En los siguientes 30 días



🕒 Daily Scrum — sincronización y autogestión

- ✔️ **Inspeccionar** el progreso hacia el Sprint Goal.
- ✔️ **Adaptar** el plan diario para avanzar como equipo.
- ✔️ **Sincronizar** el trabajo y detectar impedimentos temprano.
- ✔️ Fomentar la **autogestión**, no el reporte individual.



🕒 Daily Scrum — sincronización y autogestión

🟢 **Daily efectiva:**

- El equipo identifica un riesgo y ajusta el plan.
- Se menciona explícitamente el Sprint Goal.
- Las decisiones se toman en conjunto.

🔴 **Daily disfuncional:**

- Cada persona “reporta” sin conexión entre sí.
- Monólogos técnicos que duran 10 minutos.
- No se mencionan bloqueos.
- El Scrum Master hace preguntas de control.



Sprint Review — transparencia y colaboración con stakeholders



- ✔ Inspeccionar el **Incremento** desarrollado durante el Sprint.
- ✔ Obtener **feedback real** de stakeholders.
- ✔ Colaborar para **ajustar el Product Backlog** según aprendizaje.
- ✔ Conectar trabajo realizado con la **visión del producto**.

Sprint Review — transparencia y colaboración con stakeholders



Qué NO es la Review

🚫 No es una demo o presentación estática.

🚫 No es un evento de aprobación.

🚫 No es un reporte de avance.

.



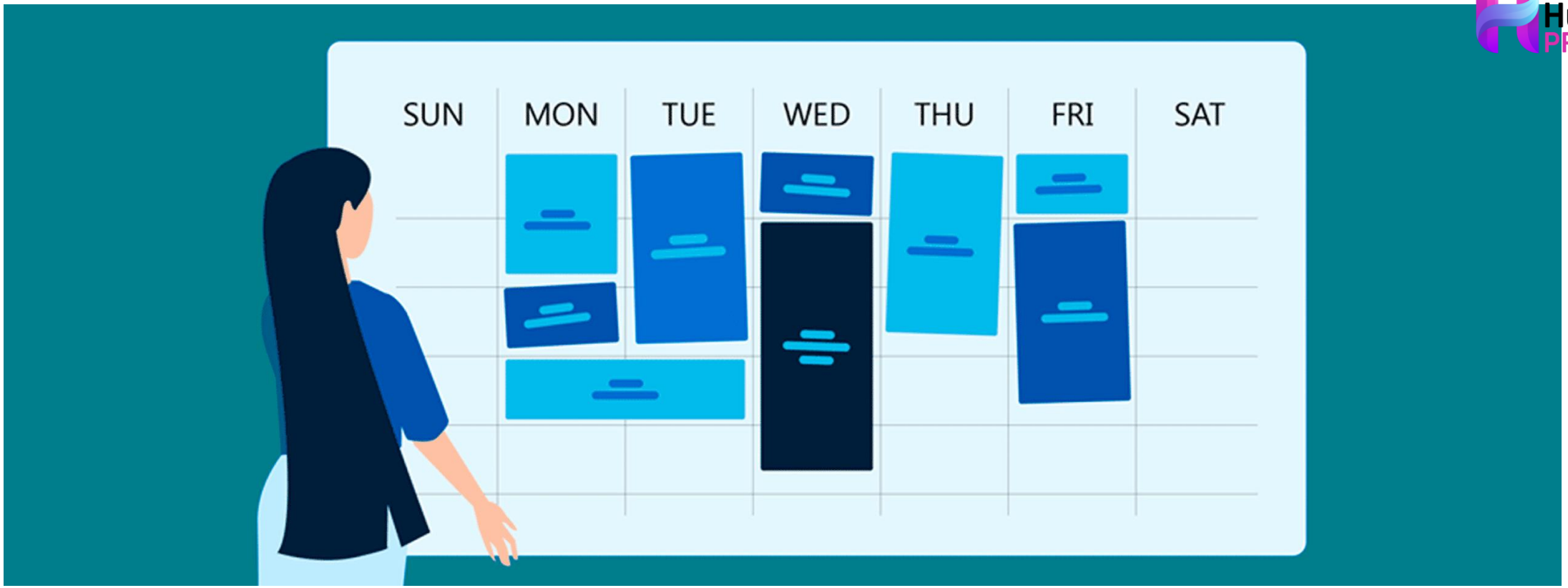
🔍 Sprint Retrospective — mejora continua y experimentación.

- Inspeccionar **cómo** trabajó el equipo.
- Identificar mejoras para aumentar calidad y colaboración.
- Comprometerse con **una o dos acciones concretas**.
- Fomentar experimentación y aprender del proceso.



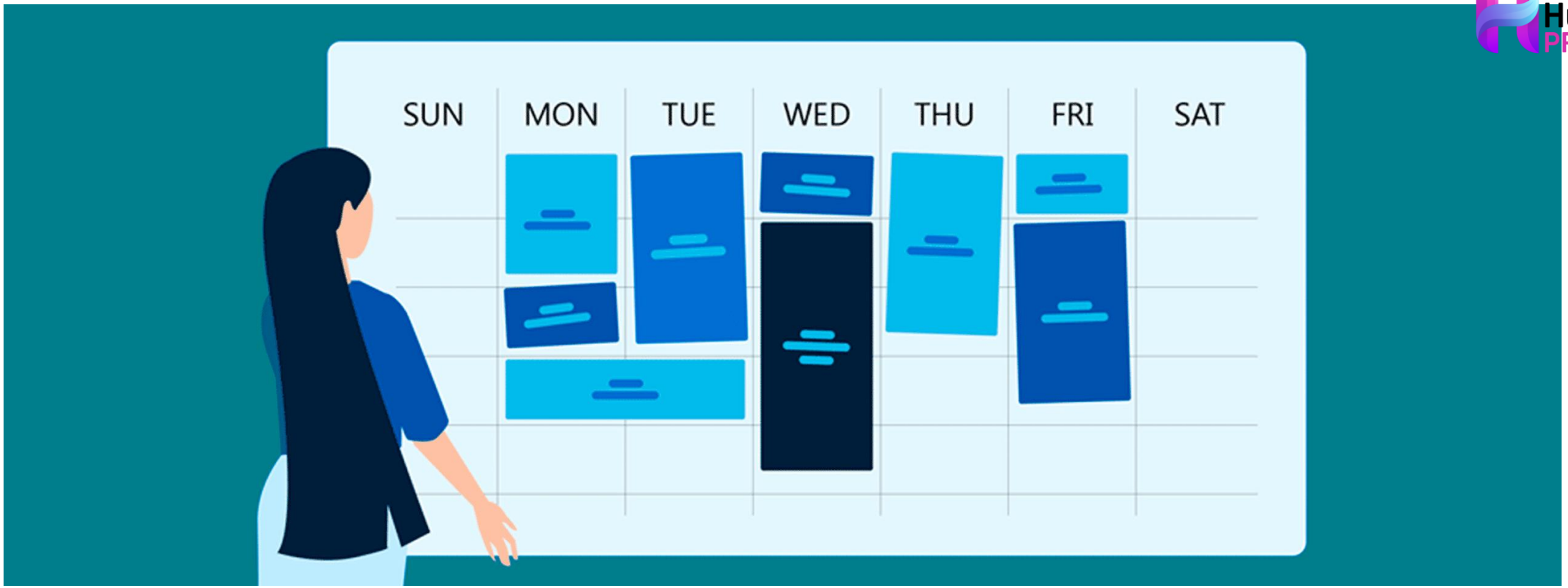
🔍 Sprint Retrospective — mejora continua y experimentación.

- Evento centrado en **personas y proceso**, no en tareas.
- Un espacio **seguro**: apertura, respeto y coraje.
- El Scrum Master facilita la conversación, no la domina.
- Las mejoras deben ser **pequeñas, alcanzables y medibles**.



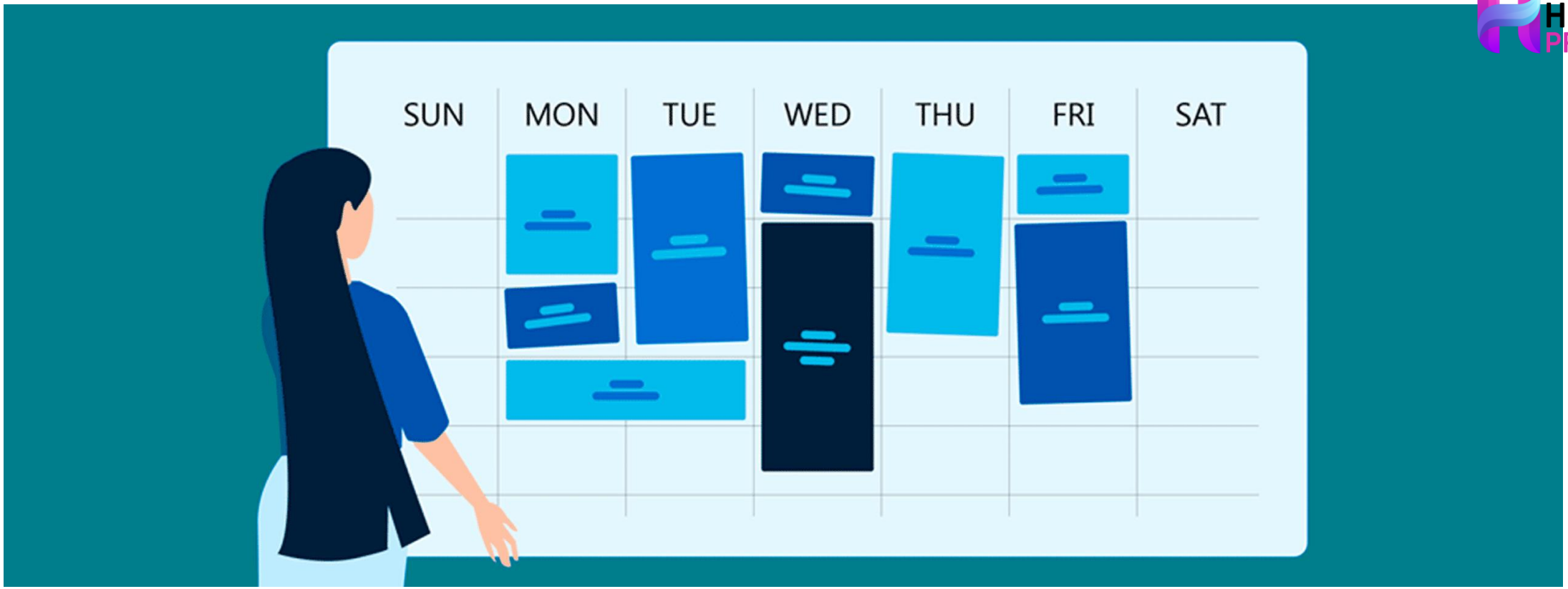
⏳ 8. Timeboxing — cómo mantener eventos productivos.

- ✔ Un límite de tiempo **firme y deliberado** para cada evento.
- ✔ Una herramienta para enfocar la conversación.
- ✔ Un mecanismo para reducir desperdicio y aumentar energía.



⏳ 8. Timeboxing — cómo mantener eventos productivos.

- 🎯 Foco en lo esencial.
- ⚡ Reuniones dinámicas y con ritmo.
- 🛑 Evita discusiones interminables.
- 🔄 Permite inspección y adaptación continua.
- ⏱️ Respeta el tiempo de todos.



⏳ 8. Timeboxing — cómo mantener eventos productivos.

🟢 **Uso efectivo del timebox:**

- Daily termina a tiempo y se crea un “parking lot” para temas técnicos.
- En la Planning, el equipo evita discusiones irrelevantes y vuelve al Sprint Goal.
- La Retro termina con 1–2 acciones claras y no se alarga sin dirección.

🔴 **Uso inefectivo del timebox:**

- La reunión se extiende “porque falta poquito”.
- Se corta una conversación importante sin acordar un espacio posterior.
- El Scrum Master se convierte en policía del tiempo, sin facilitar propósito.

Sprint Planning — Timeboxing

Duración del Sprint	Timebox máximo recomendado para la Sprint Planning
1 semana	**≈ 2 horas**
2 semanas	**≈ 4 horas**
3 semanas	**≈ 6 horas**
4 semanas	**hasta 8 horas** (valor oficial del Scrum Guide)



🤝 Sprint Review Timeboxing

Duración del Sprint	Timebox máximo recomendado para la Sprint Review
1 semana	≈ 1 hora
2 semanas	≈ 2 horas
3 semanas	≈ 3 horas
4 semanas	**hasta 4 horas** (Scrum Guide)





🔍 Sprint Retrospective — Timeboxing.

Duración del Sprint	Timebox máximo recomendado para la Sprint Retrospective
1 semana	**≈ 45 minutos – 1 hora**
2 semanas	**≈ 1.5 horas**
3 semanas	**≈ 2.25 horas**
4 semanas	**hasta 3 horas** (Scrum Guide)



🧠 Rol del Scrum Master en los eventos

✔ Facilita **propósito, foco y flujo**.
✔ Protege el **timebox** sin rigidez.
✔ Promueve **participación y seguridad psicológica**.
✔ Ayuda al equipo a conectar todo con el **Sprint Goal**.
✔ Fomenta transparencia, no control.
✔ Garantiza que los eventos **generen aprendizaje**, no burocracia.



🧠 Rol del Scrum Master en los eventos

🟢 **En la Planning:**
Hace preguntas para clarificar “por qué” y “qué”.
Ayuda a conectar el backlog con el Sprint Goal.

🟢 **En la Daily:**
Solo interviene si la conversación se convierte en reporte.
Facilita que el equipo identifique bloqueos y adapte el plan.

🟢 **En la Review:**
Promueve participación activa de stakeholders.
Refuerza transparencia sobre logros y aprendizajes.

🟢 **En la Retro:**
Crea un espacio seguro y guía dinámicas de reflexión.
Apoya en definir acciones concretas y viables.



🧩 Historia: El Sprint de la Panadería Ágil

- **¿Qué es esta historia?**

✔️ Una narrativa para vivir un Sprint completo sin usar herramientas externas.

✔️ Un cuento que muestra Planning, Daily, Review y Retro en acción.

✔️ Permite ver anti-patrones, decisiones y aprendizajes reales.

✔️ Fomenta reflexión y conversación sin obligar a la participación activa.

1️⃣ Sprint Planning — “¿Qué vamos a hornear esta semana?”

✔️ El Product Owner presenta un problema real del negocio.
✔️ El equipo co-crea un **Sprint Goal** claro:
“Reducir el tiempo de compra de los clientes.”
✔️ Se seleccionan las historias que apoyan ese objetivo.
✔️ Los Developers exploran cómo abordarlas.
✔️ El Scrum Master facilita claridad y foco.

- Sprint Goal como “la receta del Sprint”.
- Backlog como ingredientes para elegir.
- Conversación equilibrada entre PO y Developers.



2 Daily Scrum — “Revisemos la masa”

- ✔ El equipo inspecciona el progreso hacia el Sprint Goal.
- ✔ Aparece un **impedimento** real.
- ✔ Los Developers adaptan el plan del día.
- ✔ El Scrum Master apoya sin dirigir ni controlar.
- ✔ La conversación se centra en el **Sprint Goal**, no en reportes.





3️⃣ Sprint Review — “Probemos el pan con los clientes”

✔️ El equipo presenta el **Incremento real** a clientes/stakeholders.
✔️ Los stakeholders prueban el producto por sí mismos.
✔️ Surge feedback valioso e inesperado.
✔️ El Product Owner adapta el Product Backlog.
✔️ La conversación es colaborativa, no un show del equipo.



4 Sprint Retrospective — “¿Qué podemos hornear mejor?”

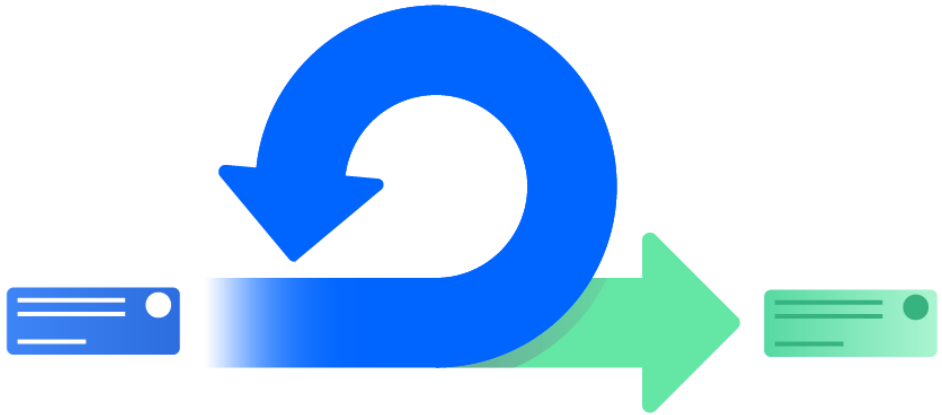
✔ El equipo reflexiona sobre **cómo trabajó**, no sobre lo que entregó.
✔ Identifican fortalezas y fricciones sin culpas.
✔ Aparece el problema recurrente (dependencia con infraestructura).
✔ Definen **una acción concreta** para el siguiente Sprint.
✔ La conversación se da en un espacio seguro.



🏁 ¿Qué aprendió la Panadería Ágil?

- ✔️ Un Sprint claro empieza con un **Sprint Goal fuerte y entendible**.
✔️ La Daily es para **coordinar y adaptarse**, no para reportar.
✔️ La Review es un espacio de **colaboración real**, no una demo.
✔️ La Retro es donde el equipo **crece, se escucha y mejora**.
✔️ El Scrum Master sostiene el espacio; el equipo crea el valor.

🏁 Cierre y conclusiones del módulo



✔ El Sprint como **contenedor de inspección y adaptación**.
✔ El propósito y uso correcto de **cada evento Scrum**.
✔ La importancia del **Sprint Goal** como brújula.
✔ Cómo lograr **Daily Scrum efectivas** y sin microgestión.
✔ Cómo convertir la Review en una **conversación de valor**.
✔ Cómo usar la Retro para **crecer y experimentar**.
✔ El **Scrum Master** como facilitador del aprendizaje y de la transparencia.

Módulo 5. Artefactos y Compromisos



Scrum Master Profesional

Contenido del Curso (1/3)

- **Módulo 1 - Fundamentos de la Agilidad y de Scrum**
  - El pensamiento ágil y el Manifiesto Ágil.
  - Contexto: complejidad, adaptabilidad y empirismo.
  - Diferencias entre la gestión tradicional y la ágil.
  - ¿Qué es Scrum y qué no es Scrum?
  - Los tres pilares de Scrum: transparencia, inspección y adaptación.
- **Módulo 2 – Valores y Cultura Scrum**
  - Compromiso, Foco, Apertura, Respeto y Coraje.
  - Cómo los valores sustentan la autogestión y la transparencia.
  - Conductas esperadas del Scrum Master para fomentar la cultura Scrum.



✨ ***El conocimiento cobra sentido cuando se convierte en acción con propósito.***

Scrum Master Profesional

Contenido del Curso (2/3)

- **Módulo 3. El Scrum Team y el Rol del Scrum Master**
  - Estructura y responsabilidades del Scrum Team.
  - Qué hace y qué no hace un Scrum Master.
  - El liderazgo servicial: coaching, facilitación y mentoring.
  - Cómo eliminar impedimentos y promover la autogestión.
  - Ejemplos de disfunciones comunes (ScrumBut) y cómo corregirlas.
- **Módulo 4. Los Eventos de Scrum**
  - El Sprint como contenedor de todo.
  - Sprint Planning: propósito, enfoque en el “por qué”, “qué” y “cómo”.
  - Daily Scrum: autogestión, foco y visibilidad del progreso.
  - Sprint Review: transparencia y colaboración con stakeholders.
  - Sprint Retrospective: mejora continua y experimentación.
  - Time-boxing y cómo mantener eventos productivos.

✨ ***El conocimiento cobra sentido cuando se convierte en acción con propósito.***



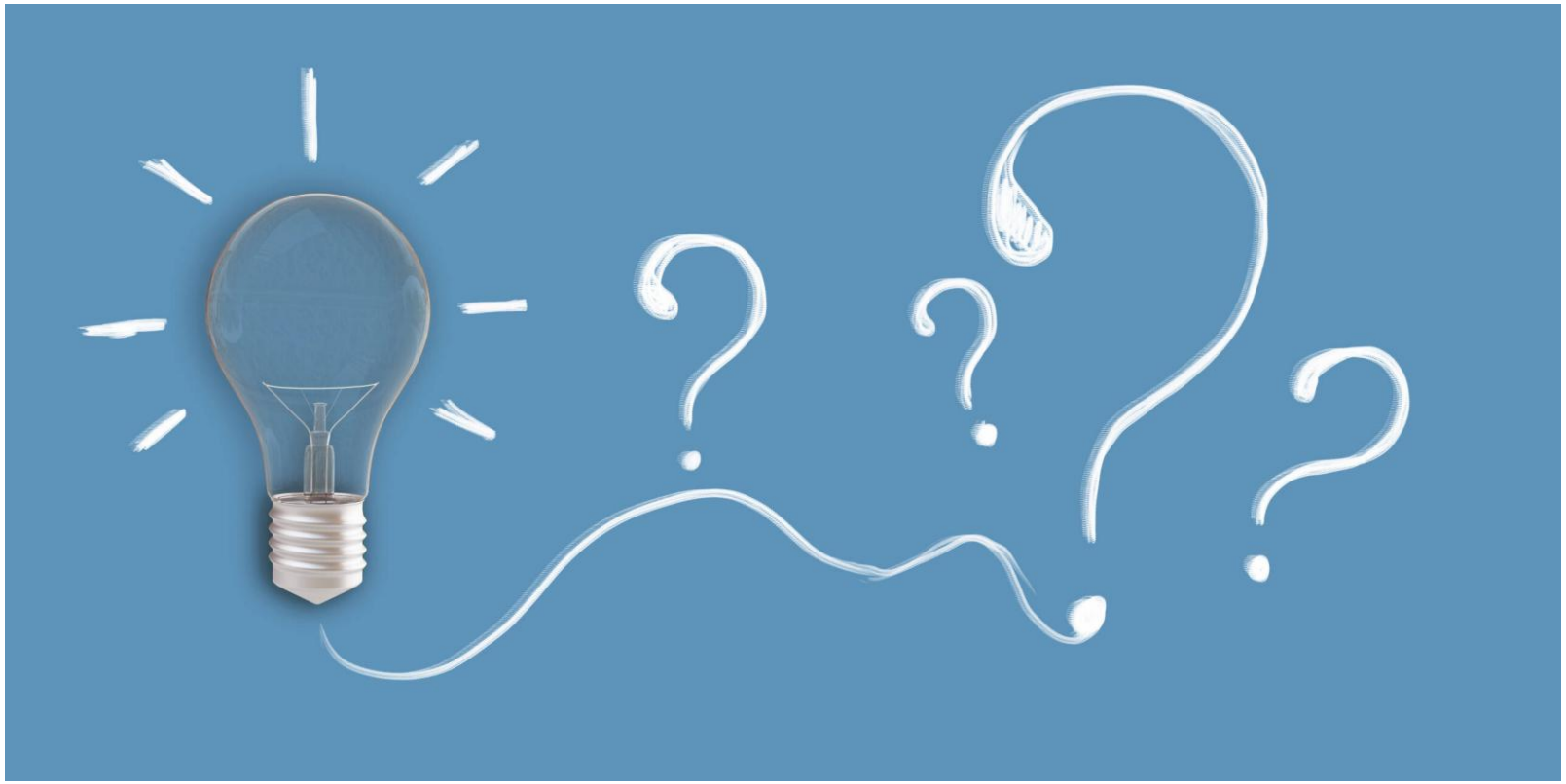
Scrum Master Profesional

Contenido del Curso (3/3)

- **Módulo 5. Artefactos y Compromisos**
  - Product Backlog, Sprint Backlog e Increment.
  - Compromisos: Objetivo del Producto, Objetivo del Sprint y Definición de Terminado.
  - Refinamiento del backlog.
  - Técnicas ágiles de estimación y priorización (INVEST, Planning Poker, SMART).
  - Control empírico del progreso (Burn-down, Burn-up).



✨ ***El conocimiento cobra sentido cuando se convierte en acción con propósito.***



Pregunta — Sprint Goal bajo amenaza

Durante el Sprint, el Product Owner llega desesperado a la Daily Scrum y dice:

“El área comercial necesita que incluyamos un nuevo requerimiento urgente. Si no lo hacemos, podríamos perder un cliente importante.”

El equipo lo analiza y concluye que incluirlo comprometería seriamente el Sprint Goal.

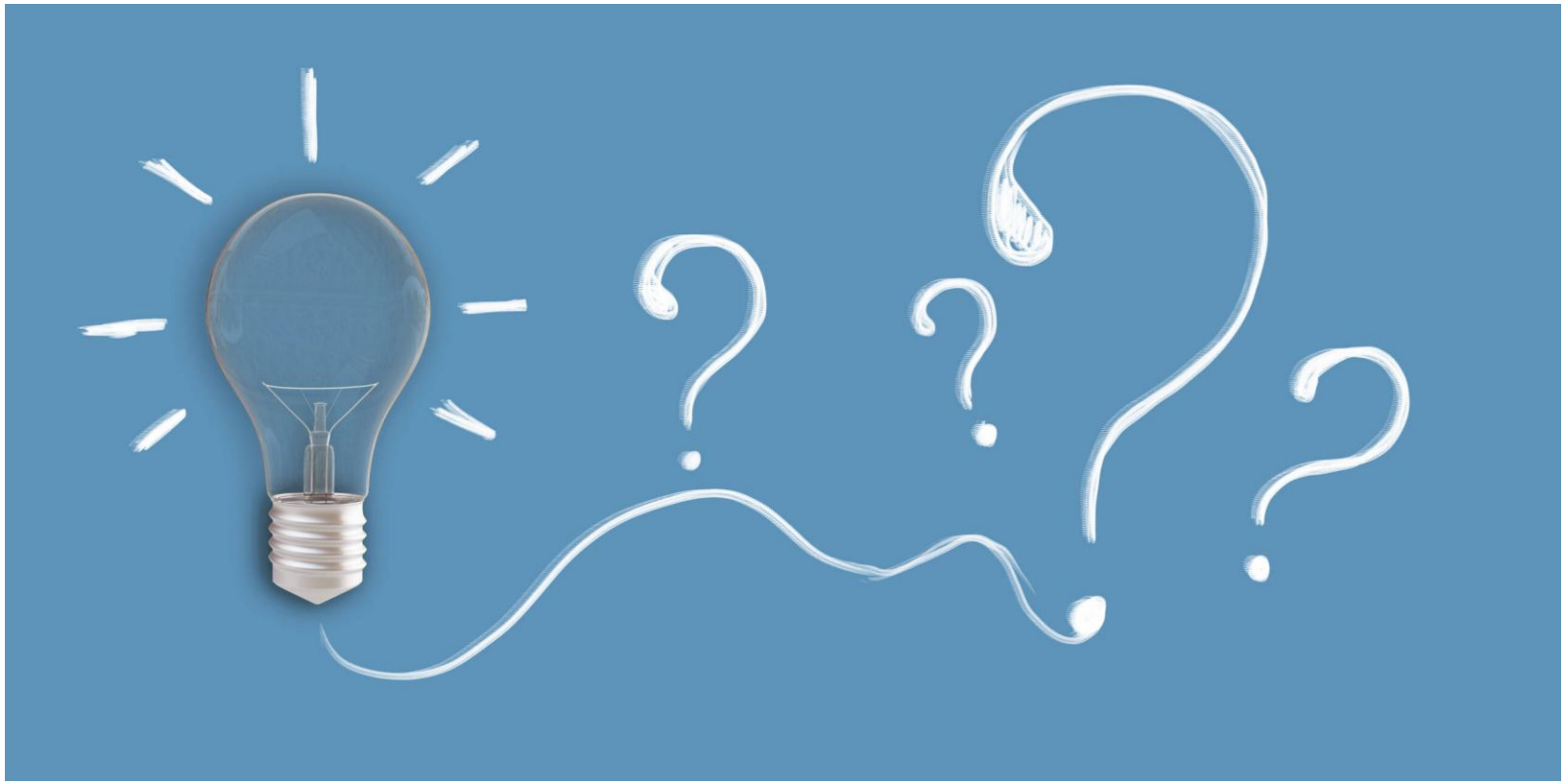
¿Qué debe hacer el Scrum Master?

A. Permitir que el PO añada el ítem porque el negocio lo necesita urgentemente.

B. Indicar que debe crearse un nuevo Sprint, cancelando el actual.

C. Guiar la conversación para que el PO y Developers evalúen si el cambio aporta al Sprint Goal.

D. Decidir él mismo si el ítem entra o no, protegiendo al equipo.



Pregunta — Daily Scrum como reporte

En un equipo maduro, observas que la Daily Scrum se ha convertido en un turno de "actualización de status" donde cada Developer le reporta al Scrum Master, usando la estructura clásica:

"Ayer hice..."

"Hoy haré..."

"Tengo este bloqueo..."

No hay conversación entre Developers ni adaptación del plan.

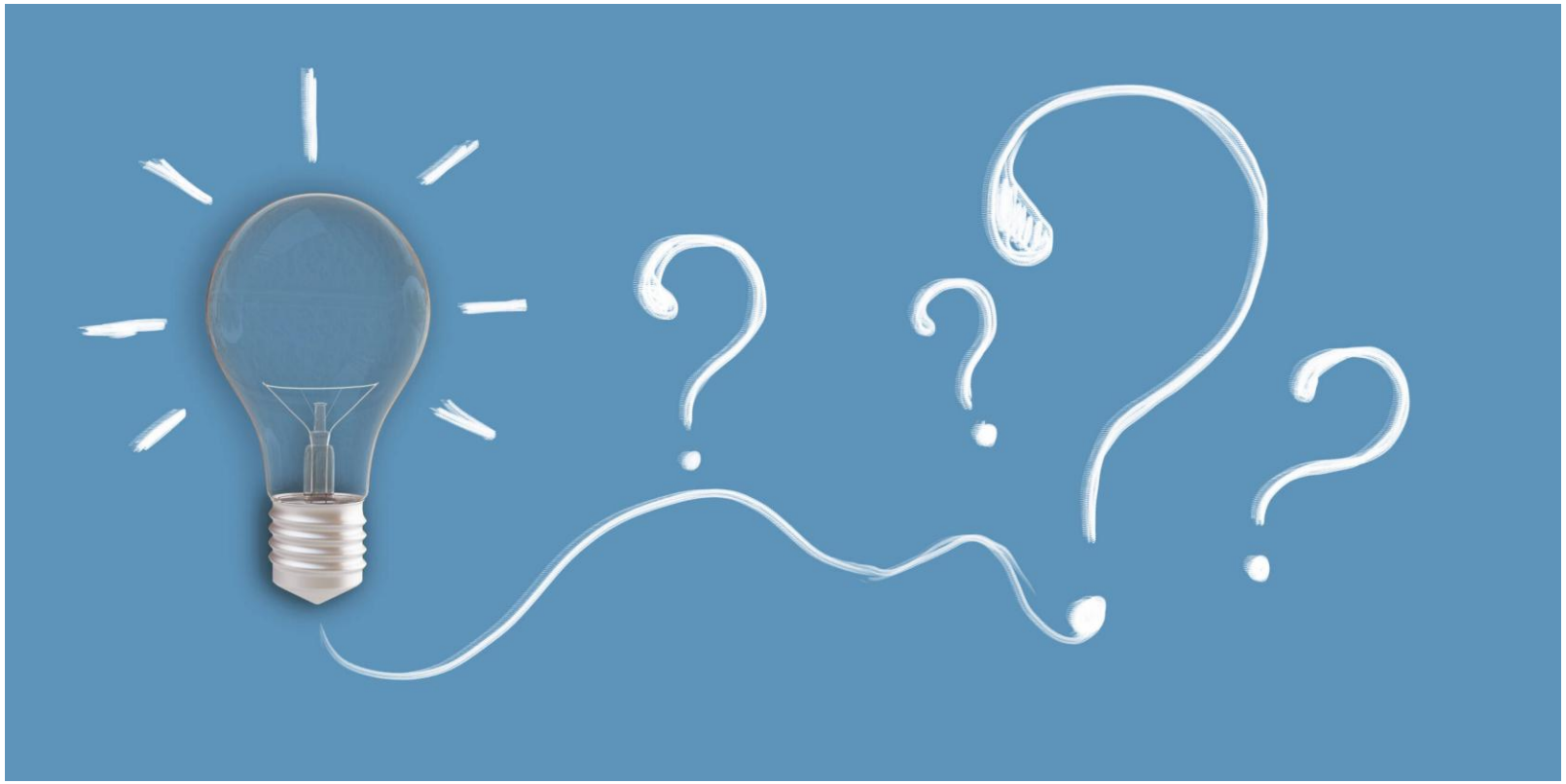
¿Qué indica esto?

A. El equipo está aplicando correctamente la técnica tradicional de Daily.

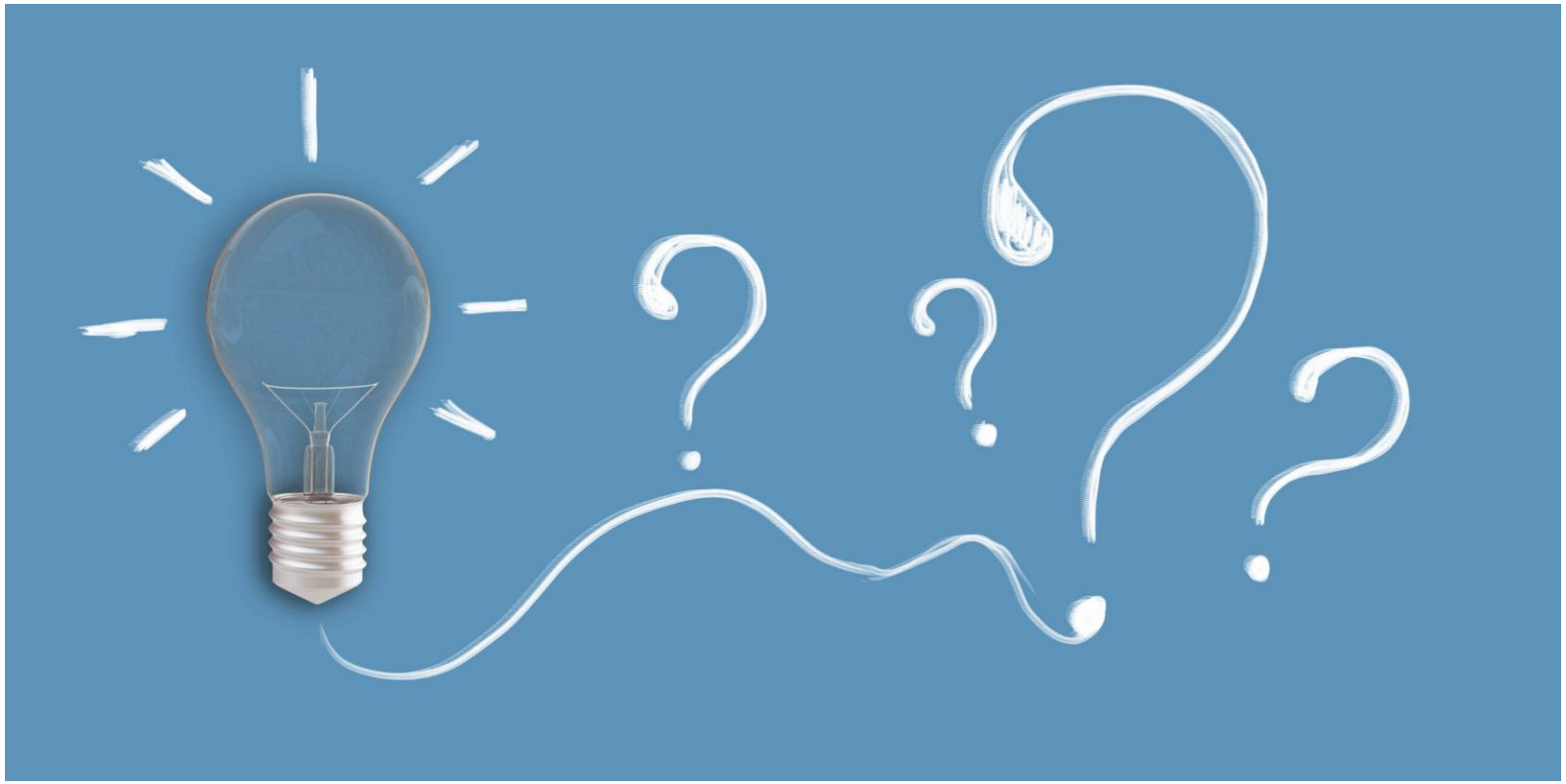
B. La Daily Scrum se está usando incorrectamente como un reporte en cascada.

C. El Scrum Master está facilitando demasiado y debería callarse más.

D. Es normal mientras el equipo complete sus historias.



Pregunta — Sprint Review sin Incremento
El Scrum Team ha trabajado intensamente durante el Sprint. Sin embargo, el día de la Sprint Review descubren que el Increment no cumple la Definition of Done por un error crítico pendiente.
¿Qué es lo más adecuado?
A. Presentar el trabajo “casi listo”, explicando que solo queda un detalle por corregir.
B. Cancelar la Sprint Review.
C. No presentar el ítem como Incremento y discutir con stakeholders lo que se aprendió.
D. Firmar una excepción a la DoD para esta vez solamente.



Pregunta — Sprint Retrospective políticamente difícil

El equipo evita hablar de un conflicto con un stakeholder clave, argumentando: *“Mejor no toquemos ese tema para que no haya problemas; solo revisemos lo técnico.”*

El Scrum Master sabe que ese problema afecta la confianza y bloquea decisiones clave.

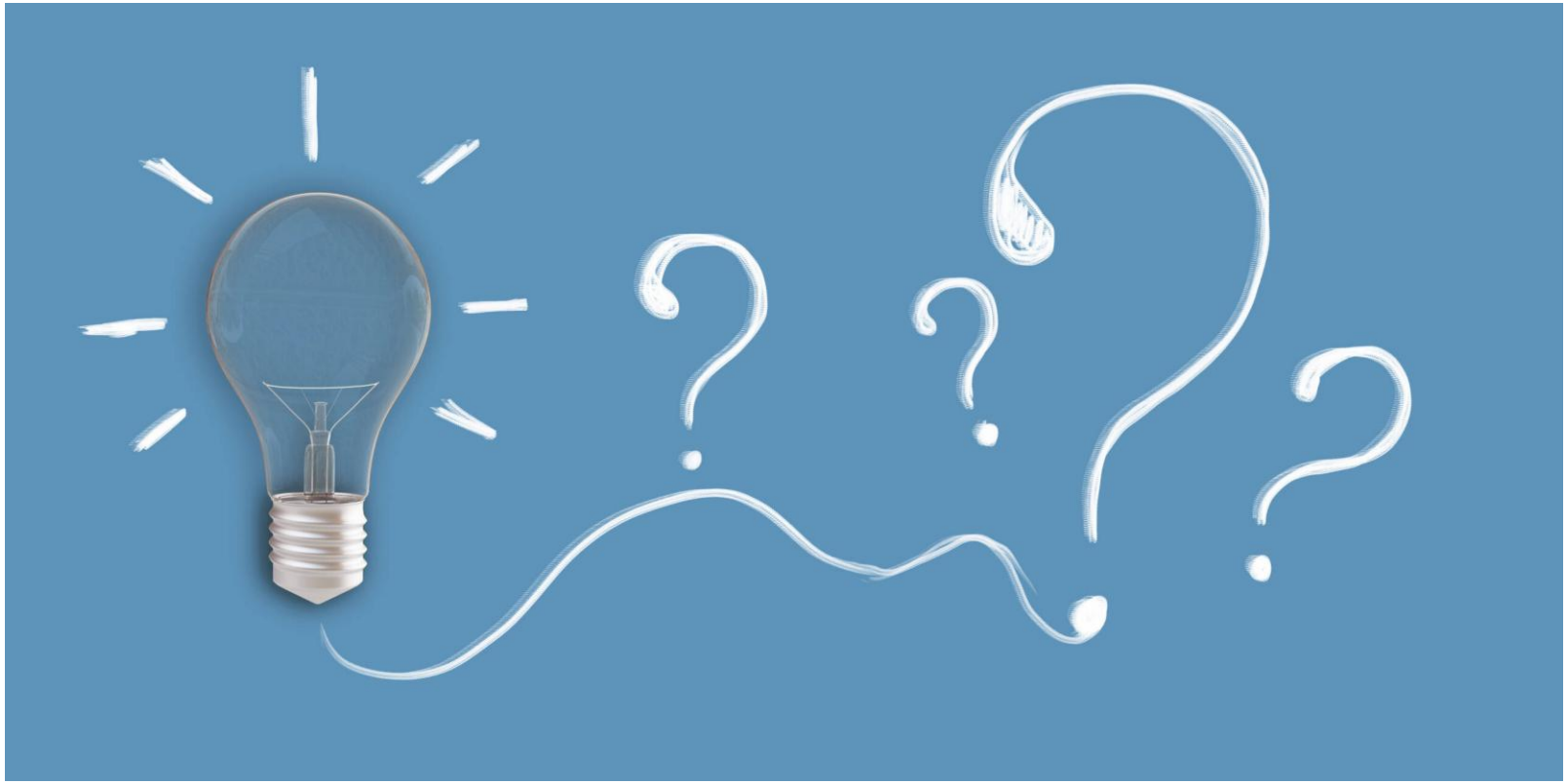
¿Qué debería hacer?

A. Respetar la decisión del equipo: si no quieren hablarlo, no se fuerza.

B. Escalar el conflicto a gerencia para que alguien intervenga.

C. Crear un espacio seguro para que la conversación ocurra sin miedo a repercusiones.

D. Agendar una reunión privada con el PO solamente.



Pregunta — Sprint Planning y falta de claridad

Durante la Sprint Planning, el Product Owner llega con 15 historias nuevas que no han sido refinadas. El equipo no entiende criterios de aceptación ni dependencias y siente que el PO quiere “meter cosas por presión del calendario”.

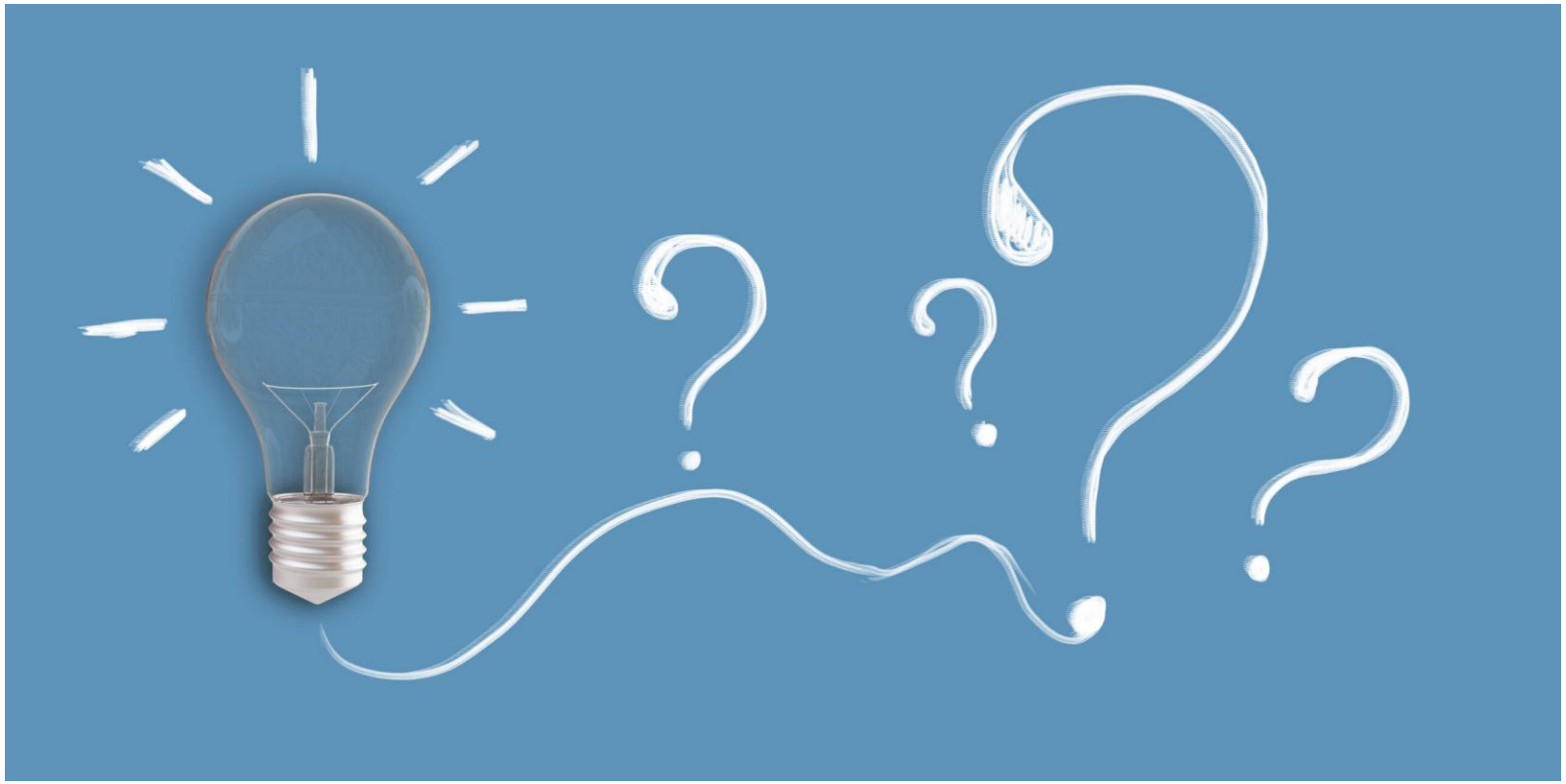
¿Qué debe hacer el Scrum Master?

A. Permitir que las historias entren al Sprint y aclararlas durante la Daily.

B. Indicar que, sin claridad suficiente, los Developers no pueden comprometerse.

C. Negociar por el equipo para seleccionar solo las historias más importantes.

D. Asignar tareas entre Developers para asegurar que todo avance.



Pregunta — Timeboxing en la Sprint Retrospective

El equipo tiene una Retrospectiva timebox de 1.5 horas.

Sin embargo, después de 40 minutos, un Developer dice:

“No alcanzaremos a cubrir todos los temas; propongo extender la Retro una hora más para terminar todo.”

El Product Owner está de acuerdo y quiere quedarse. La mitad del equipo también está de acuerdo; la otra mitad quiere respetar el timebox.

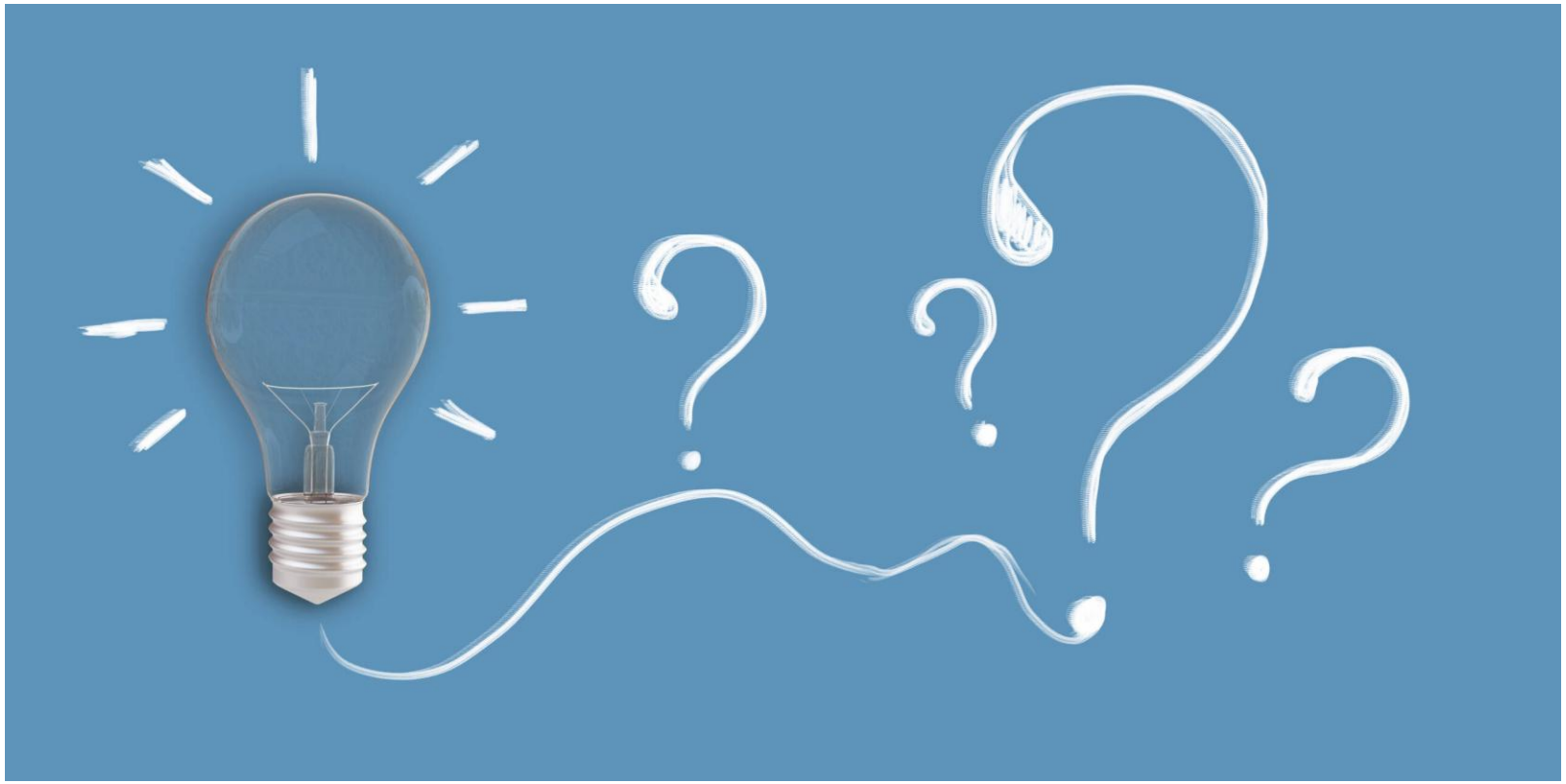
¿Qué debería hacer el Scrum Master?

A. Extender la Retro; si el equipo quiere hablar, hay que aprovechar el impulso.

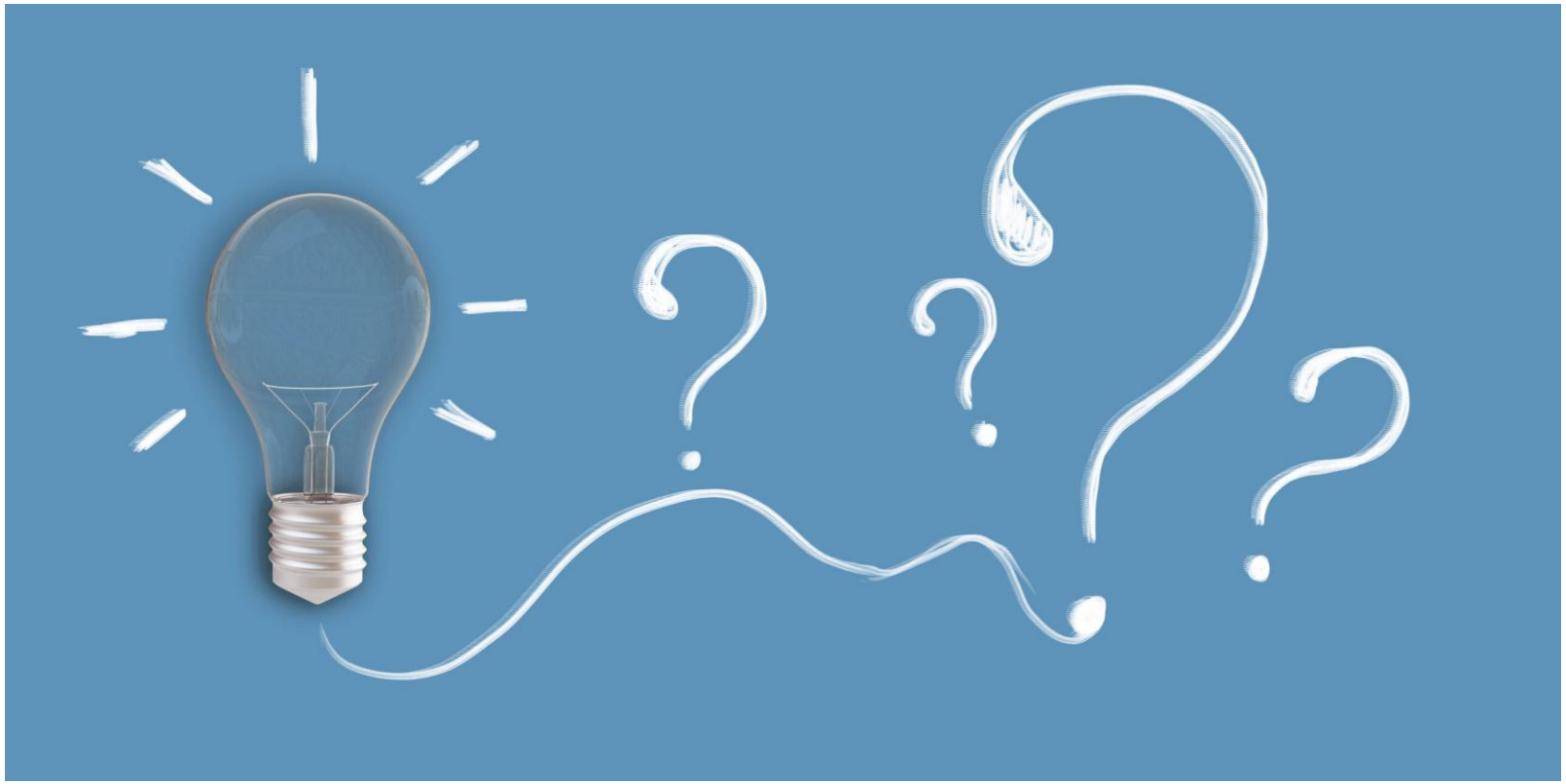
B. Mantener el timebox, y priorizar los temas esenciales dentro del tiempo disponible.

C. Cancelar la Retro y reagendarla completa para otro día.

D. Decidir según votación democrática; si la mayoría quiere extender, se extiende.



Pregunta — Daily Scrum y autoridad del timebox
Un equipo realiza su Daily Scrum siempre en 30 minutos.
Argumentan que “es porque todos quieren hablar, y es difícil cortar la conversación”.
El Scrum Master intenta facilitar para acortar, pero el equipo lo ve como “control excesivo”.
¿Qué debería hacer el Scrum Master?
A. Permitir que dure 30 minutos: si el equipo lo prefiere, es autogestión.
B. Cortar la reunión en 15 minutos, sin importar quién esté hablando.
C. Reentrenar al equipo en el propósito de la Daily y facilitar que la conversación se enfoque en inspección del Sprint Backlog.
D. Eliminar la Daily y reemplazarla por actualizaciones asincrónicas.



Pregunta — Retro con problema sistémico

En una Retrospectiva, el equipo expresa que la mayor fricción cada Sprint es que los deployments requieren aprobación manual del área de infraestructura, lo cual genera retrasos de 2–3 días.

Sin embargo, el área de infraestructura nunca asiste y argumenta que “no forma parte del Scrum Team”.

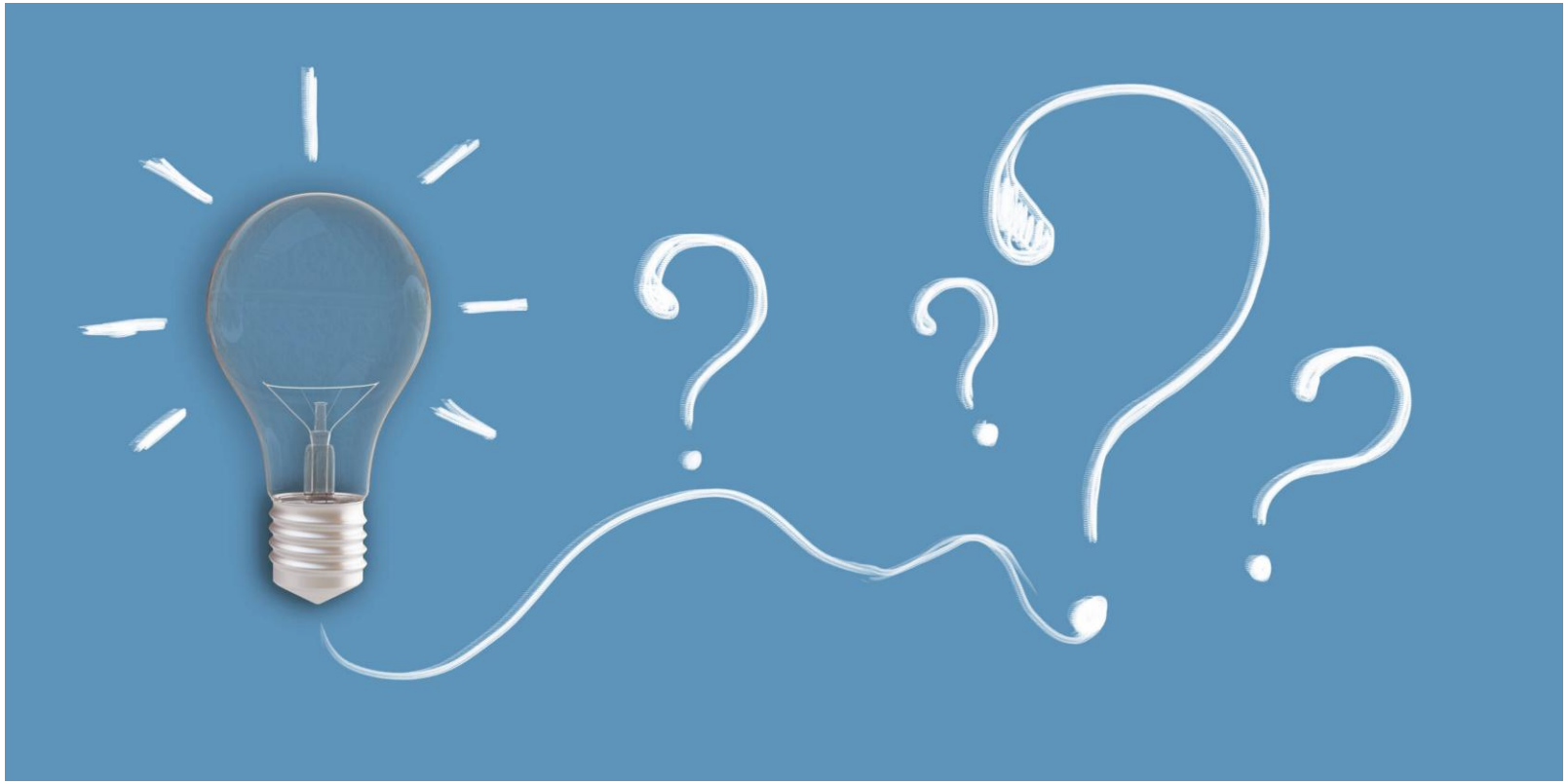
¿Qué debería hacer el Scrum Master?

A. Resolver él mismo la automatización de despliegues.

B. Presionar a infraestructura para que venga a todas las Retros.

C. Facilitar conversaciones entre equipos para tratar impedimentos sistémicos.

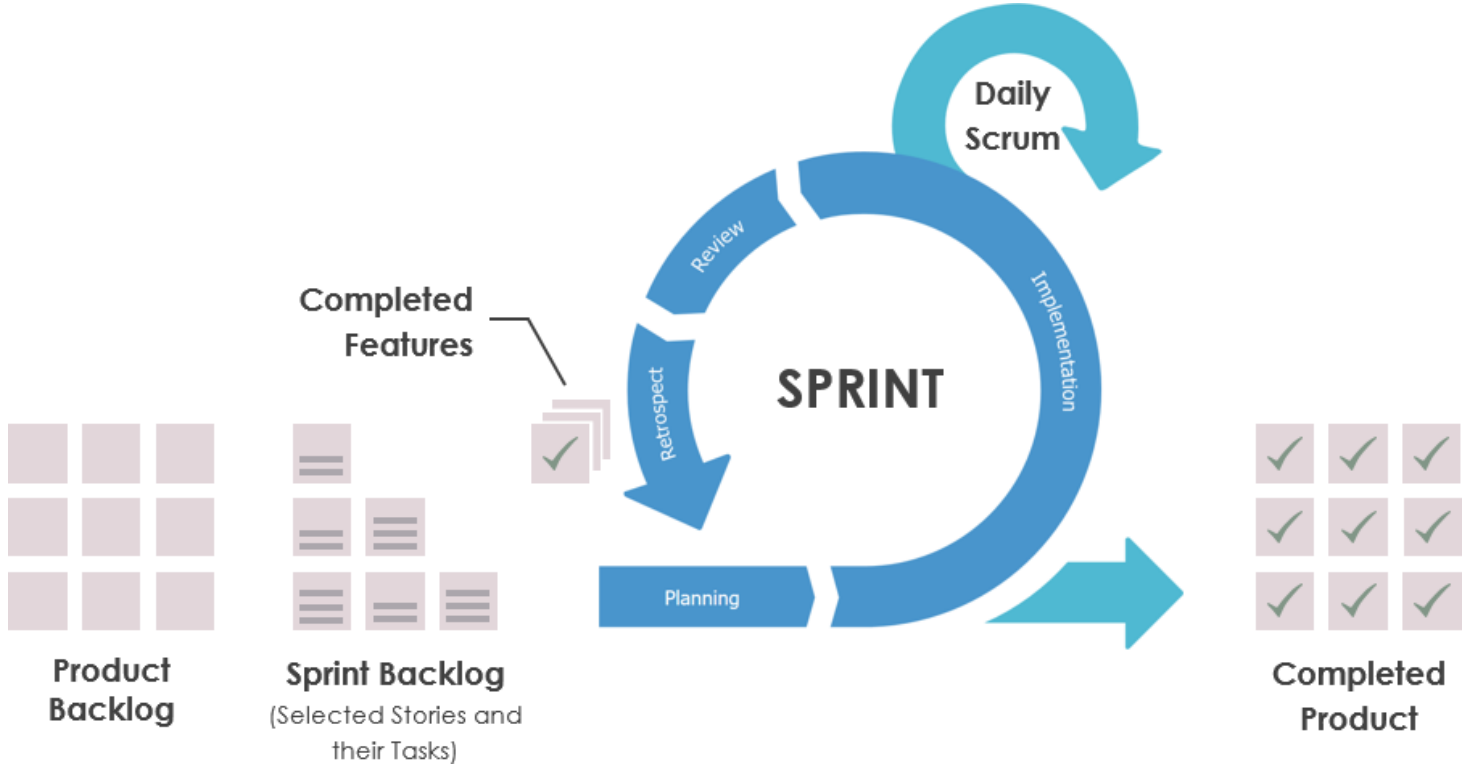
D. Indicar que la Retro solo trata de procesos internos del Scrum Team.



Pregunta — Sprint Review como “demo”
En tu organización, la Sprint Review siempre es una presentación en PowerPoint dirigida por el Scrum Master, donde se muestran diapositivas, no el producto funcionando.
Los stakeholders están callados.
No hacen preguntas ni dan feedback.
¿Cuál es el problema principal?
A. La Review debería durar más tiempo.
B. El Scrum Master está asumiendo el rol del equipo y del PO.
C. El equipo no necesita feedback si ya completó el Sprint.
D. Las diapositivas reemplazan bien al Incremento.

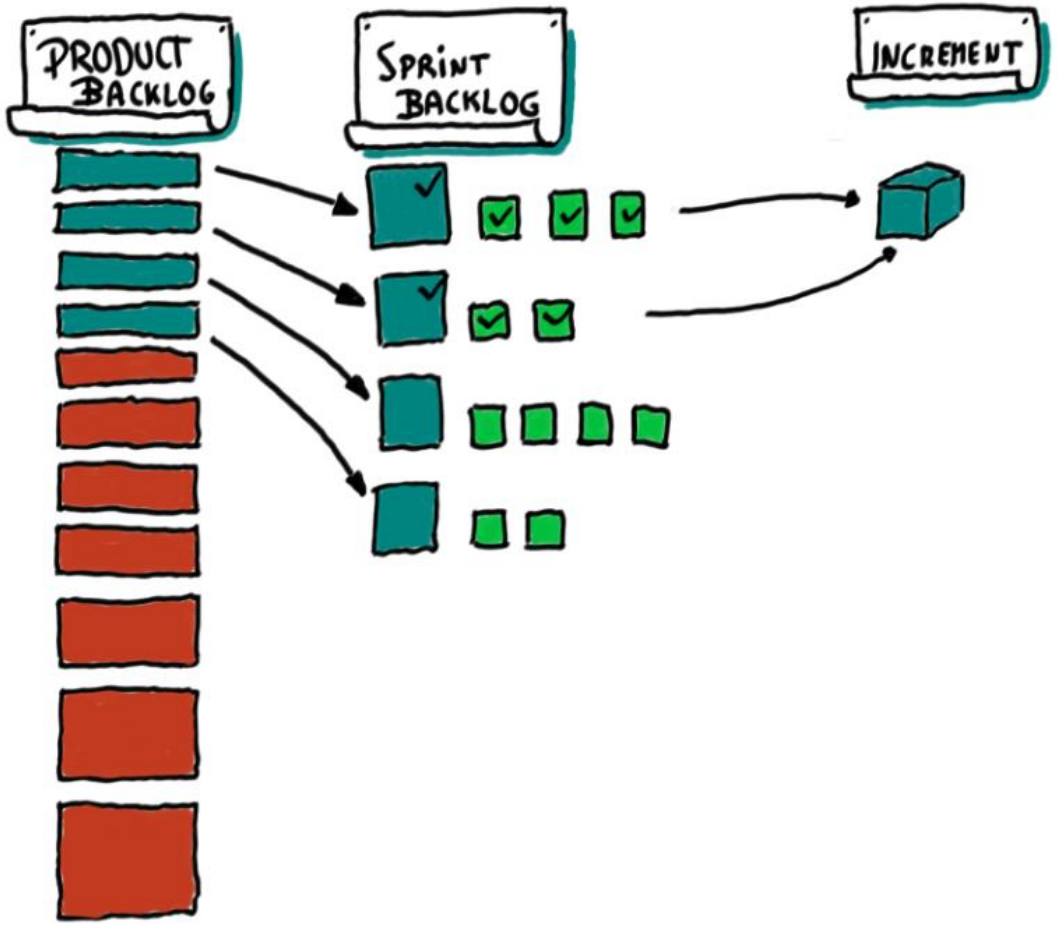
Resumen del Módulo 4 — Eventos de Scrum

✔ El Sprint como **contenedor de inspección y adaptación**.
✔ Sprint Planning: propósito → *por qué, qué y cómo*.
✔ Daily Scrum: coordinación y autogestión.
✔ Sprint Review: colaboración real con stakeholders.
✔ Sprint Retrospective: mejora continua y experimentación.
✔ El rol del Scrum Master como **facilitador del aprendizaje**.



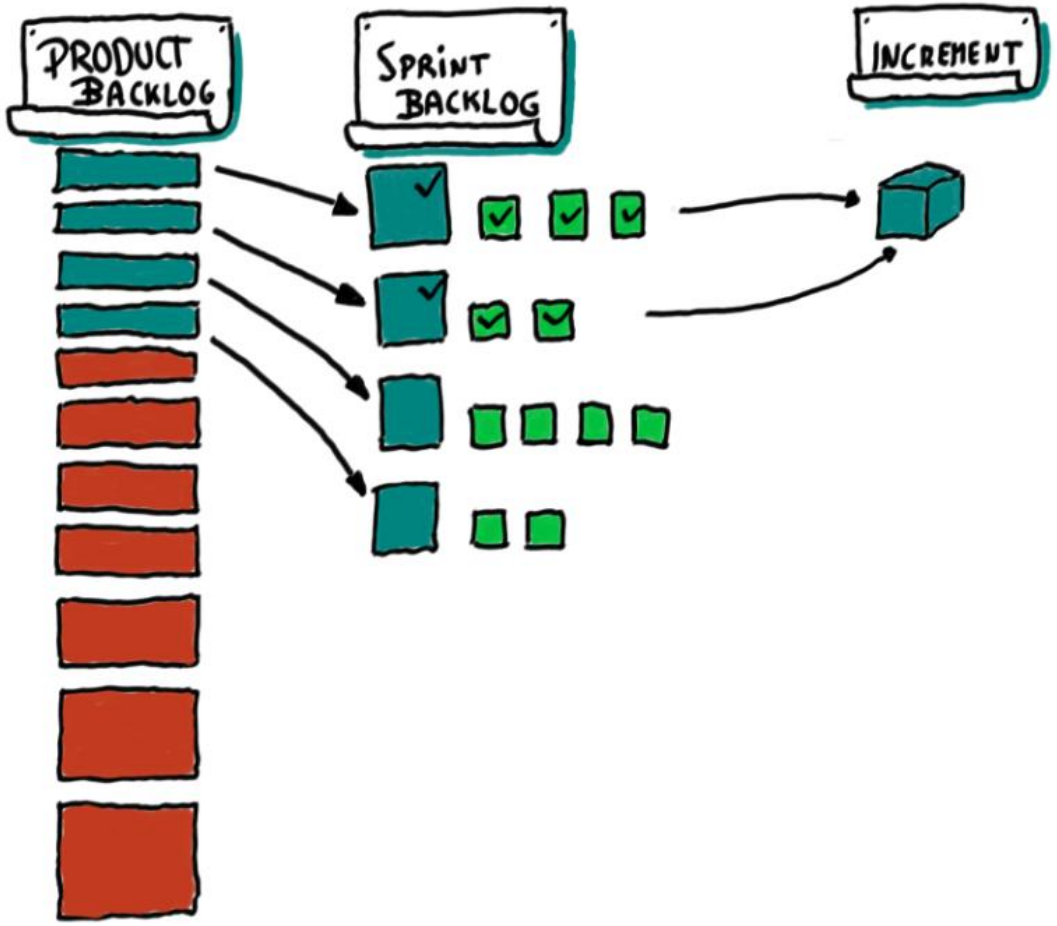
3 ¿Qué son los artefactos de Scrum?

✔ **Transparencia** → que todos vean lo mismo.
✔ **Inspección** → entender el progreso real.
✔ **Adaptación** → decidir cómo avanzar y mejorar.

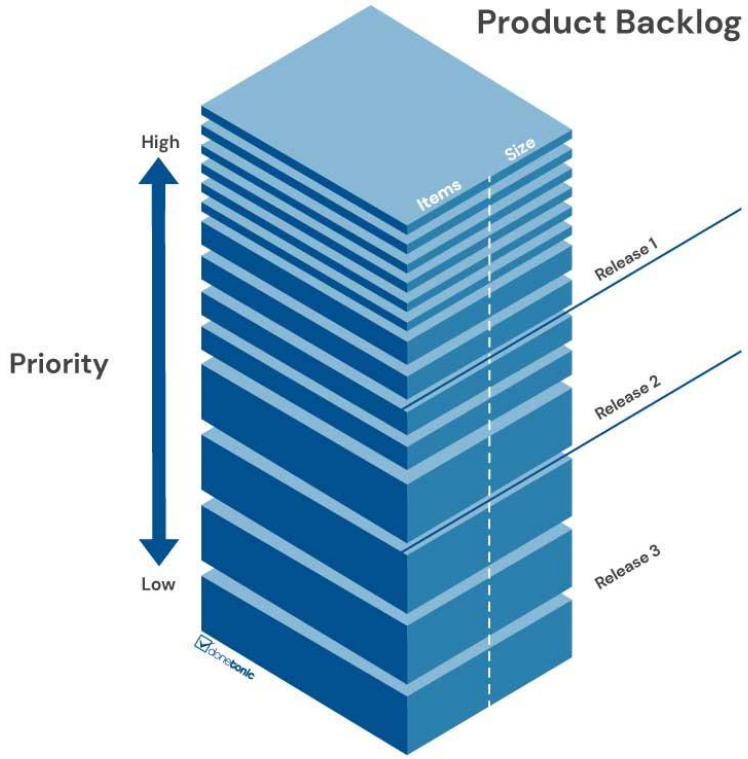


3 ¿Qué son los artefactos de Scrum?

- **Product Backlog** → *todo el trabajo posible para el producto.*
- **Sprint Backlog** → *el plan para alcanzar el Sprint Goal.*
- **Increment** → *resultado usable y "Done" del Sprint.*

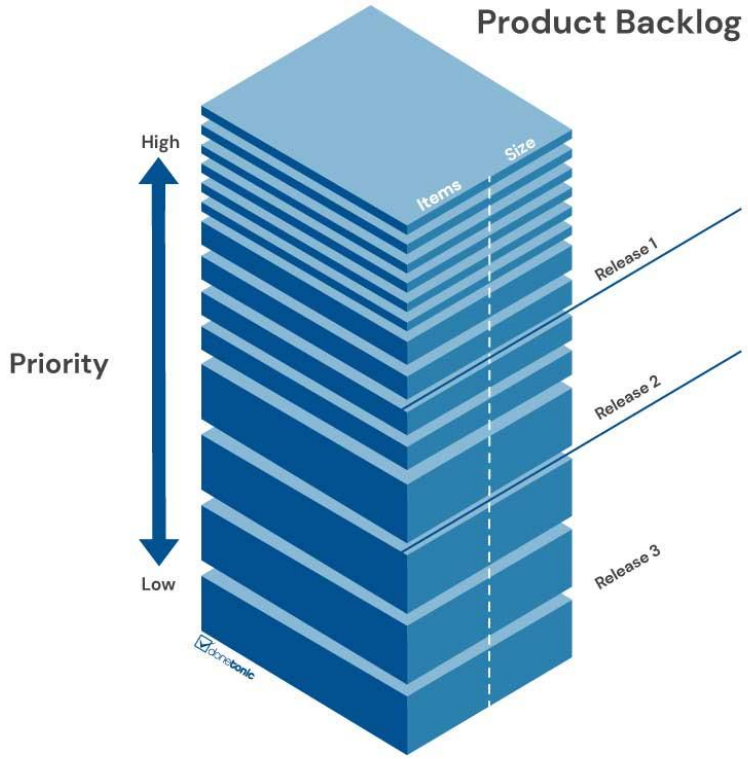


4 Product Backlog: el contenedor de todo el trabajo del producto



✔ La **lista ordenada** de todo el trabajo necesario para mejorar el producto.
✔ Evoluciona constantemente (no es un documento estático).
✔ Es **propiedad del Product Owner**.
✔ Es la fuente única de trabajo para los Developers.

4️⃣ Product Backlog: el contenedor de todo el trabajo del producto



🔁 Vivo y dinámico (se actualiza todo el tiempo).
🎯 Ordenado según **valor**, riesgo y estrategia.
✍️ Puede incluir: historias, bugs, mejoras, spikes, tareas técnicas, etc.
🌱 Refinado de forma continua.

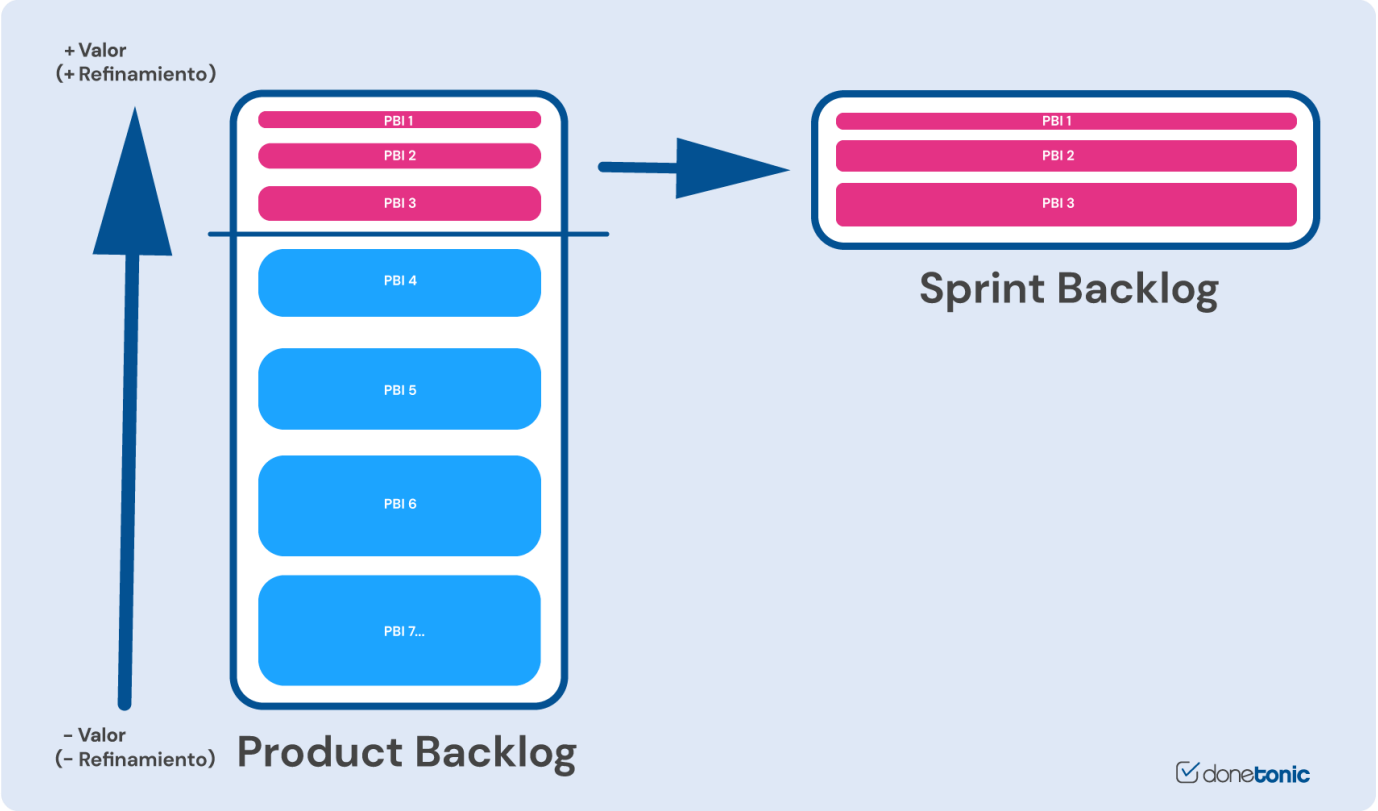
5 Sprint Backlog: el plan para alcanzar el Sprint Goal

✔ El **plan táctico** del equipo para cumplir el Sprint Goal.
✔ Incluye:

Ítems seleccionados del Product Backlog

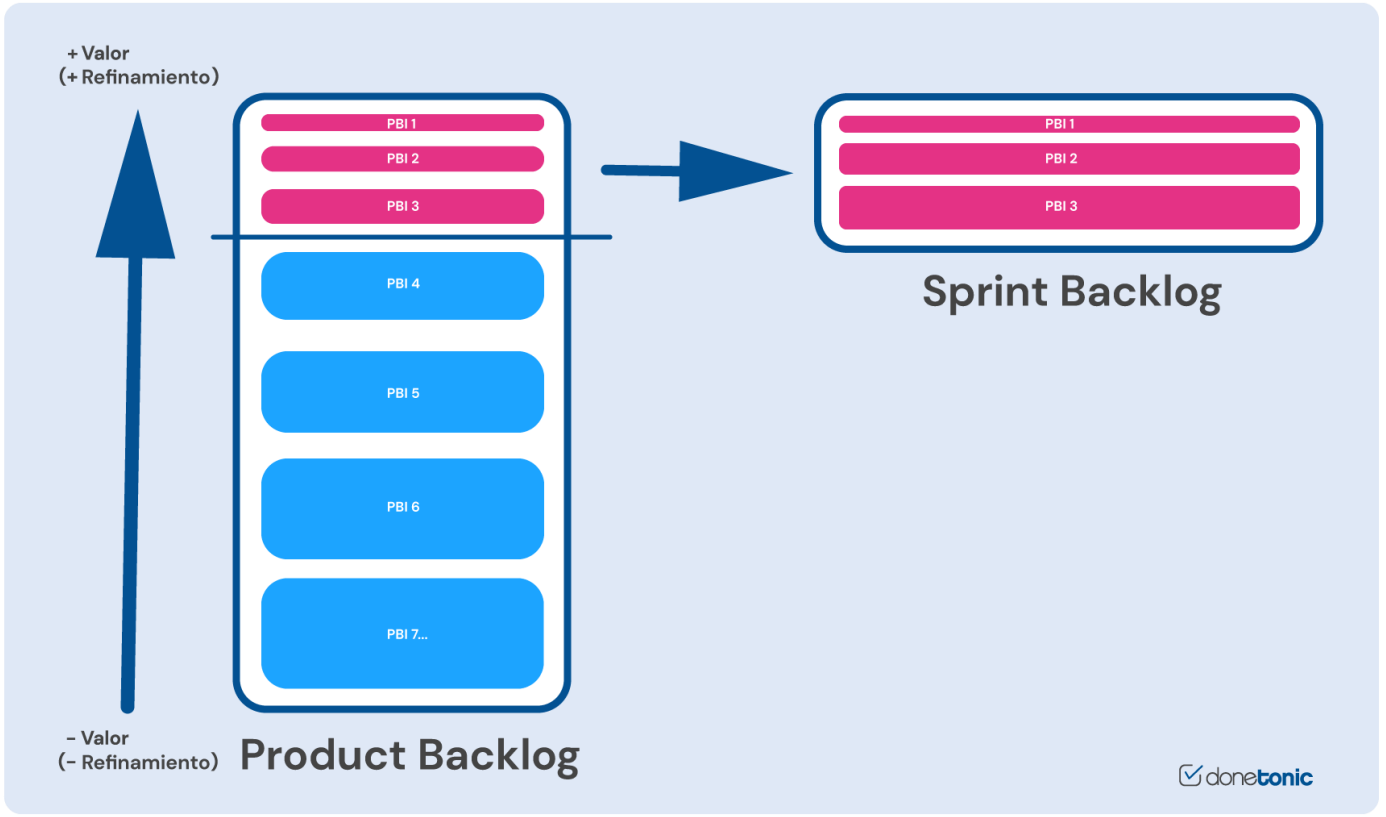
El Sprint Goal

El plan de ejecución de los Developers
✔ Es **propiedad exclusiva de los Developers**.



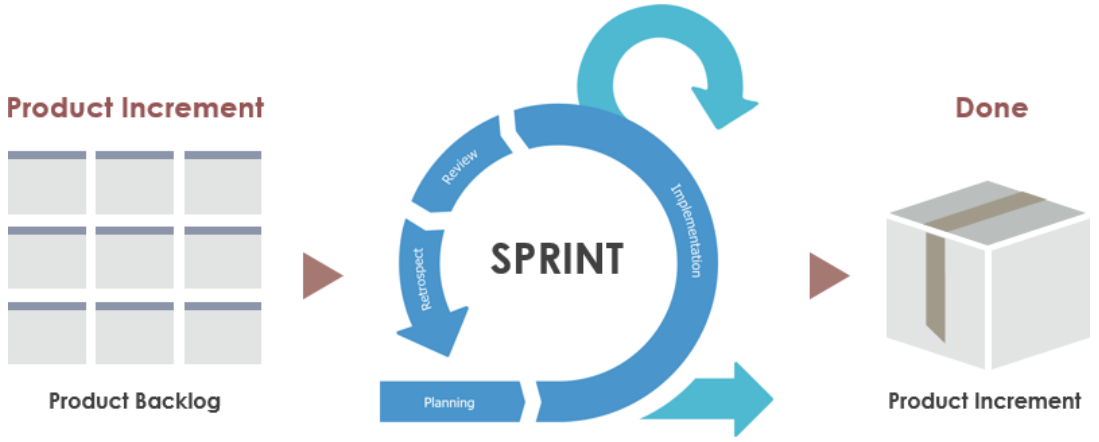
5️⃣ Sprint Backlog: el plan para alcanzar el Sprint Goal

🔁 **Vivo y emergente**: se actualiza cada día.
🗓️ Refleja el **trabajo real**, no una predicción rígida.
🧩 Conecta cada actividad con el Sprint Goal.
👀 100% transparente (visiblemente disponible).



6 Increment: el corazón de la transparencia

✔ El **resultado usable** del Sprint.
✔ Debe cumplir la **Definition of Done**.
✔ Puede haber **uno o varios Incrementos** en un mismo Sprint.
✔ Es el indicador más transparente del progreso real del producto.



7 Compromisos de Scrum: por qué existen

✔ Para aportar **transparencia** a cada artefacto.
✔ Para que el equipo tenga **claridad y foco**.
✔ Para conectar trabajo diario con **valor real**.
✔ Para evitar que los artefactos se conviertan en documentos vacíos.



7 Compromisos de Scrum: por qué existen

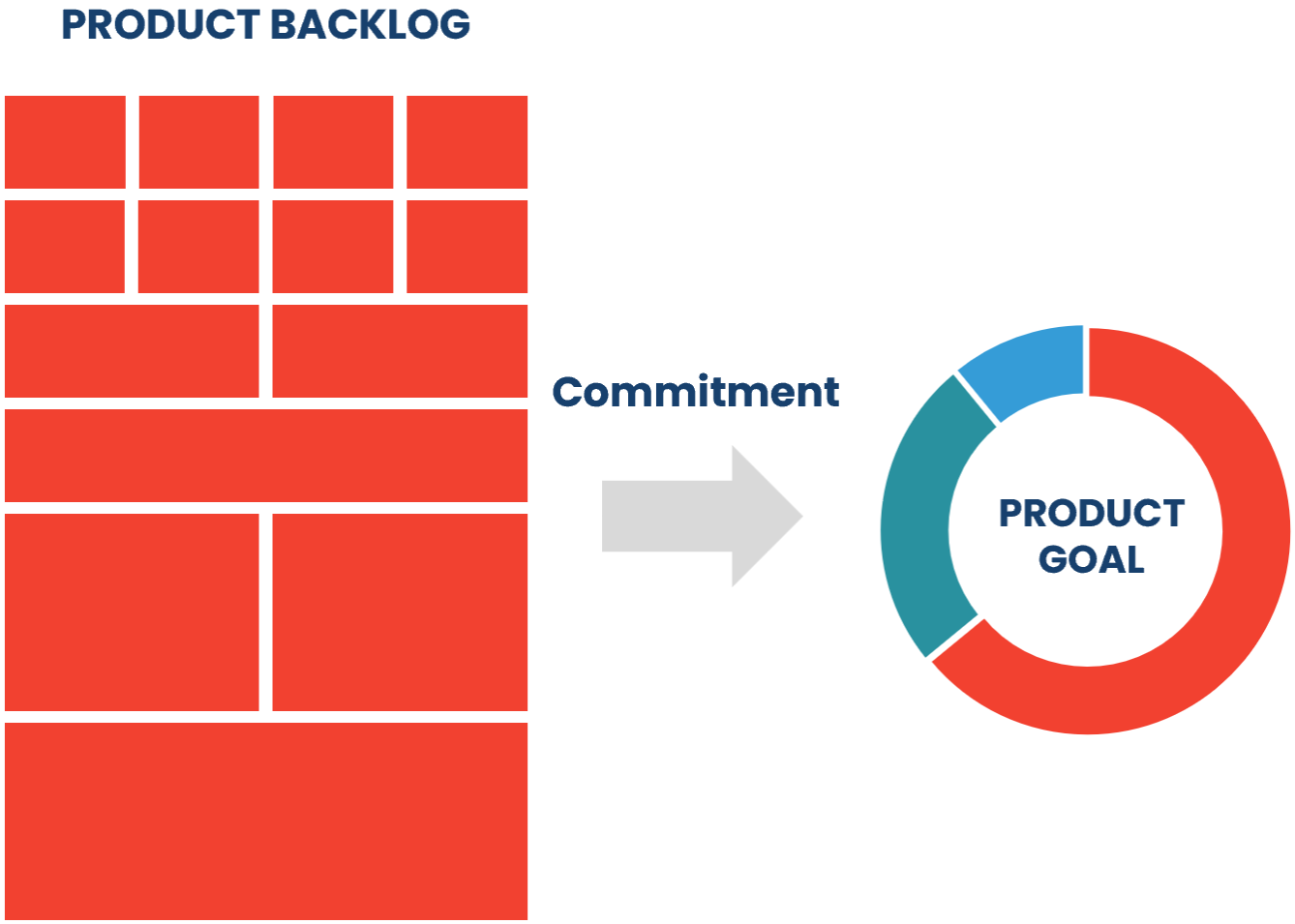
Artefacto	Compromiso	Propósito
Product Backlog	Product Goal	Dirección del producto
Sprint Backlog	Sprint Goal	Foco del Sprint
Increment	Definition of Done	Calidad y transparencia



8 Objetivo del Producto (Product Goal)

¿Qué es el Product Goal?

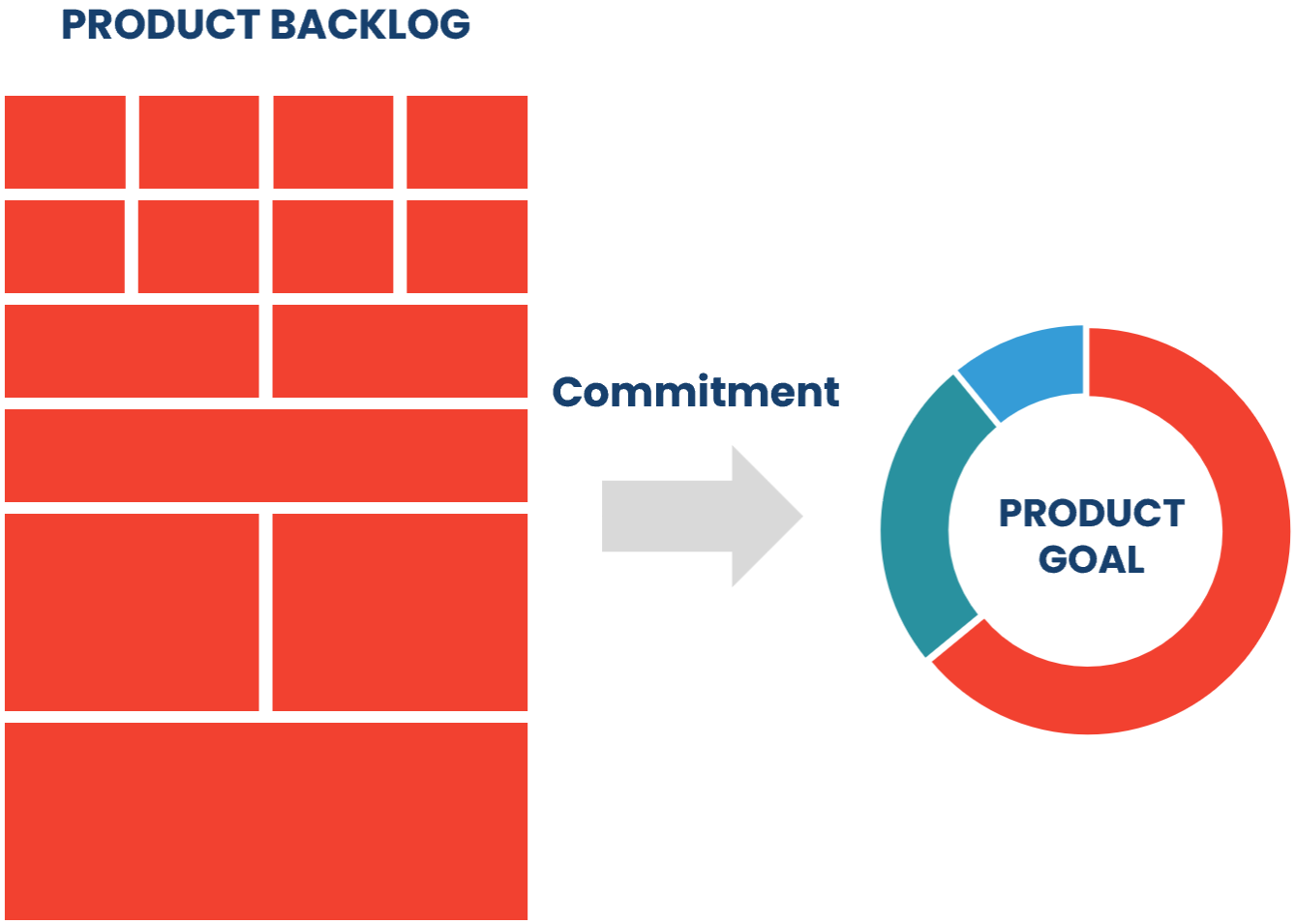
- ✔ La **meta estratégica** del producto a mediano/largo plazo.
- ✔ Unifica el trabajo de múltiples Sprints hacia un mismo resultado.
- ✔ Permite tomar decisiones de **priorización y enfoque**.
- ✔ Es responsabilidad del **Product Owner** definirlo y comunicarlo.



8 Objetivo del Producto (Product Goal)

¿Qué es el Product Goal?

🎯 Representa “lo que queremos lograr como producto”.
🧭 Guía la estrategia y el orden del Product Backlog.
🔁 Puede evolucionar a medida que el mercado o el aprendizaje cambian.
🧩 Divide la visión en pasos alcanzables a través de Sprints.



9️⃣ Objetivo del Sprint (Sprint Goal)

- **¿Qué es el Sprint Goal?**

✔️ El **propósito** del Sprint.
✔️ Explica **por qué** se hace el trabajo del Sprint.
✔️ Alinea al equipo hacia un resultado común.
✔️ Permite tomar decisiones bajo incertidumbre.



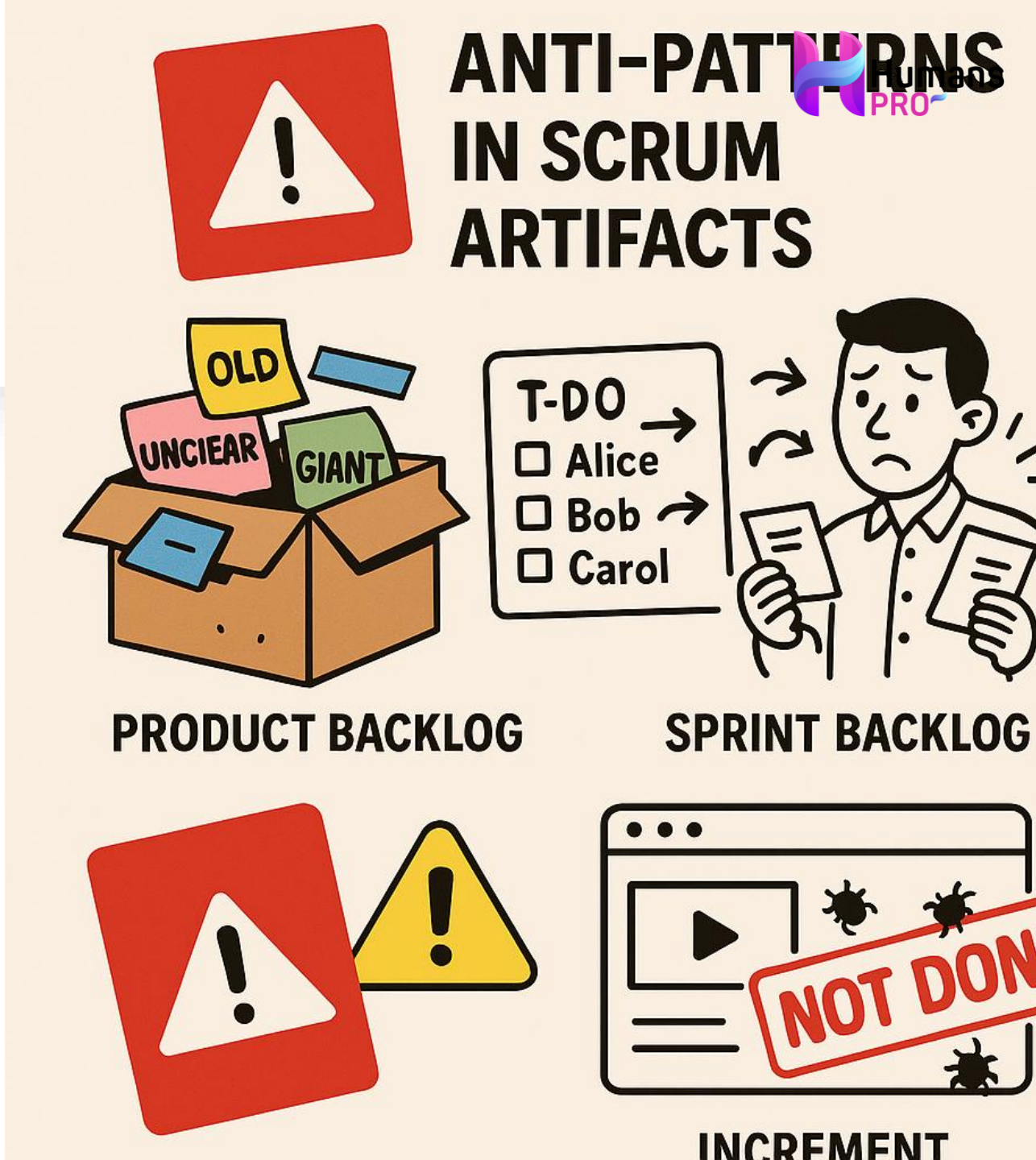
9️⃣ Objetivo del Sprint (Sprint Goal)

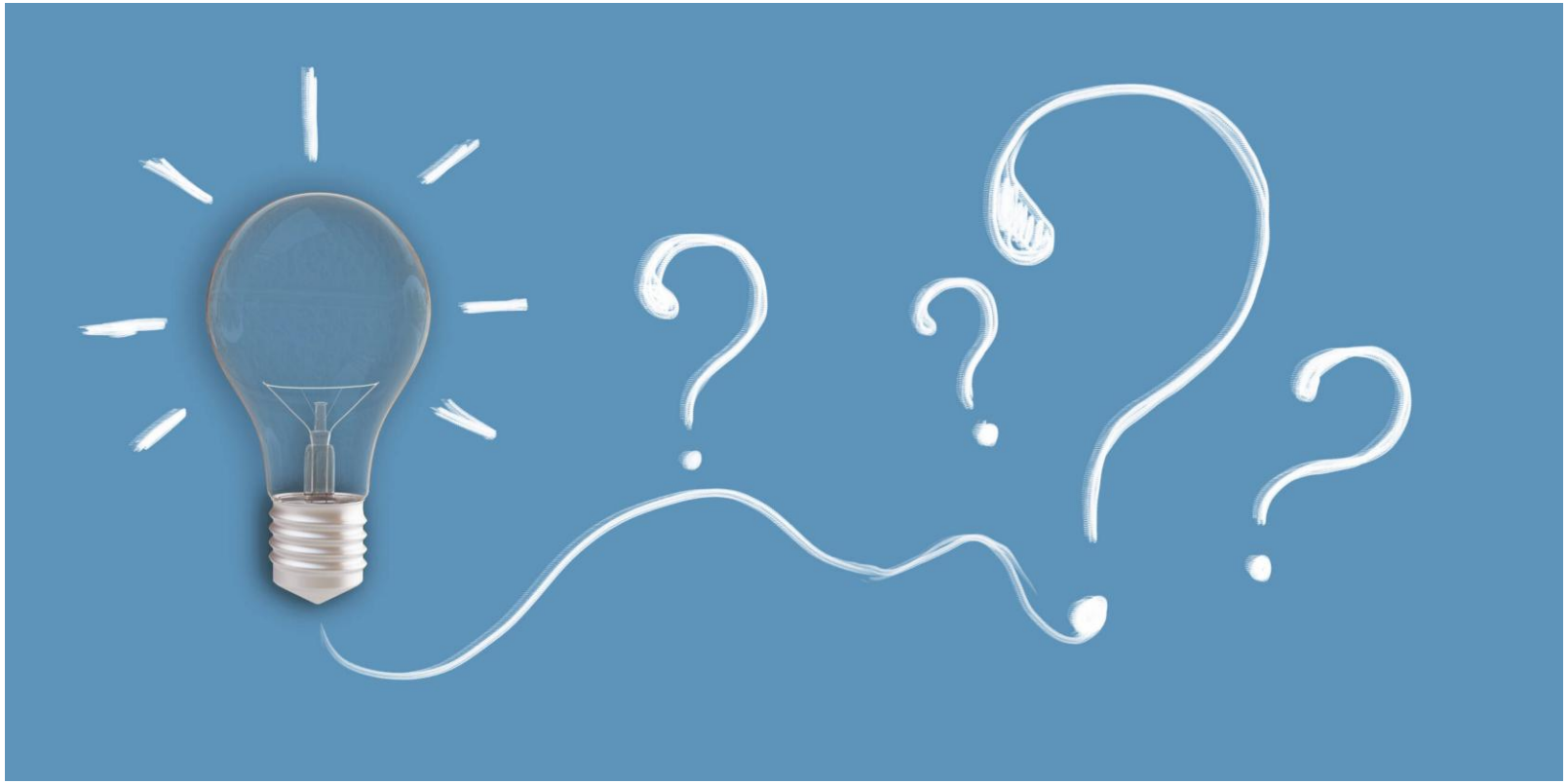
🎯 Se define durante la Sprint Planning.
👥 Es un **compromiso** del Scrum Team, no solo del PO.
🔁 Permite flexibilidad: se puede cumplir incluso si cambia el plan.
🧭 Guía la adaptación diaria en la Daily.



Anti-patrones comunes en los artefactos

- ❌ **Anti-patrones del Product Backlog**
- Backlog inflado con cientos de ítems sin dueño.
- Historias gigantes sin refinamiento.
- Ítems antiguos que nadie recuerda quién pidió.
- Prioridad influenciada por “quién grita más fuerte”.
- ❌ **Anti-patrones del Sprint Backlog**
- Convertido en una lista de tareas asignadas por persona.
- Cambios de alcance constantes sin revisar el Sprint Goal.
- El SM “distribuye” trabajo.
- Tracking por horas en lugar de valor.
- ❌ **Anti-patrones del Increment**
- Presentar trabajo incompleto en la Review (“casi listo”).
- Ignorar la DoD “porque no hay tiempo”.
- Múltiples versiones del producto circulando.
- Bugs críticos considerados como “terminado”.





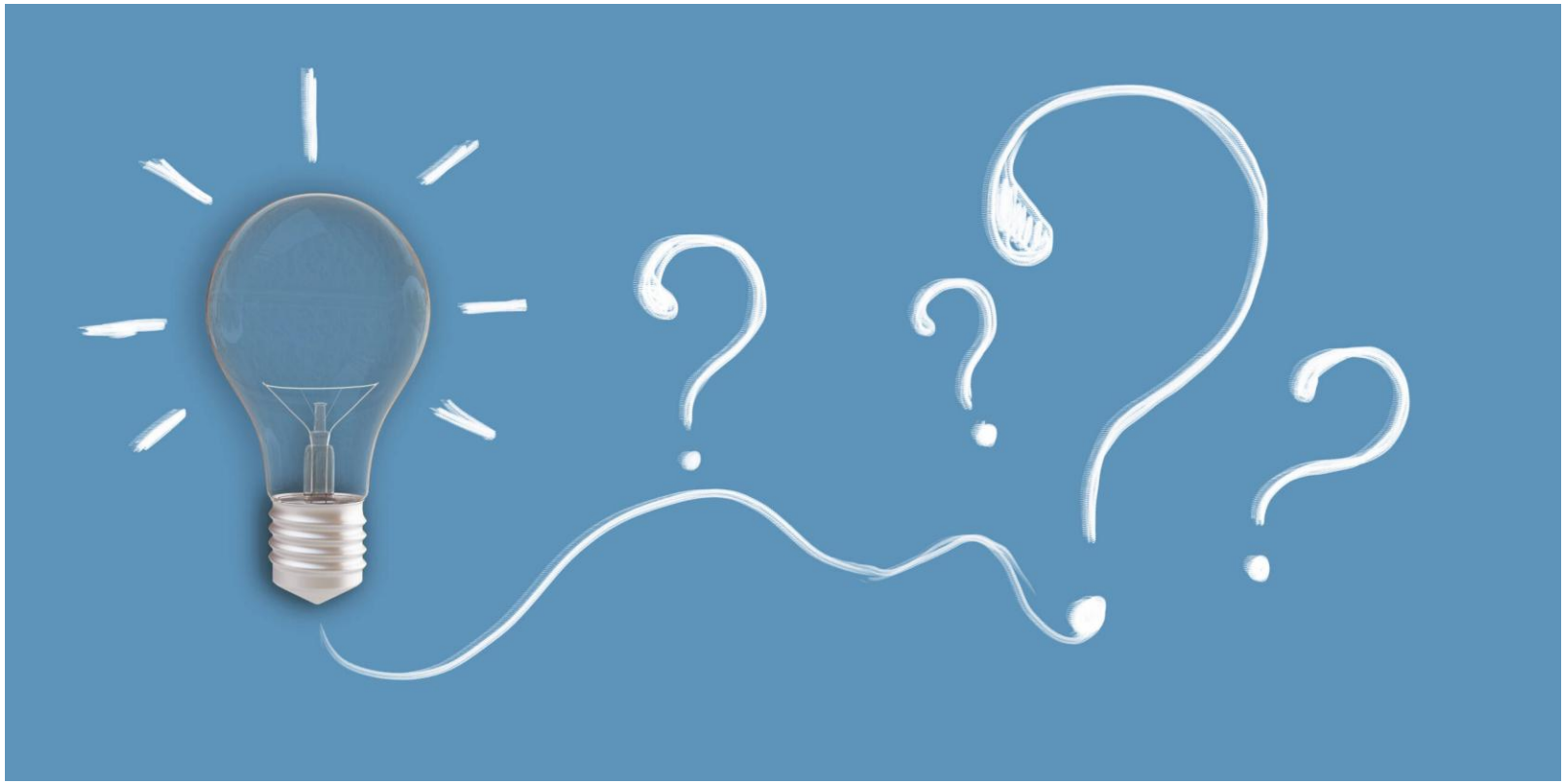
¿Cuál de los siguientes enunciados describe mejor el propósito de los artefactos de Scrum?

A. Ayudar al Scrum Master a medir la productividad del equipo.

B. Aumentar la documentación del proyecto.

C. Generar transparencia para habilitar inspección y adaptación.

D. Documentar el proceso de entrega del equipo.



¿Cuál de los siguientes elementos NO debería estar en un Product Backlog?

A. Historias de usuario
B. Bugs
C. Tareas internas del equipo
D. Objetivos personales del del equipo



🔟 Definition of Done (DoD)

✔️ Un **acuerdo explícito** sobre qué significa “terminado”.
✔️ Garantiza **calidad**, consistencia y transparencia.
✔️ Aplica a **cada Incremento**, no a “cuando se pueda”.
✔️ Reduce retrabajo y evita deuda técnica.



🔟 Definition of Done (DoD)

🧩 Es un compromiso del **Scrum Team**, no del SM ni del PO.

⭐ Debe cumplirse en **cada Sprint**, sin excepciones.

🔍 Da claridad sobre qué es entregable.

🚫 Evita entregar “casi”, “parcial” o “demo”.



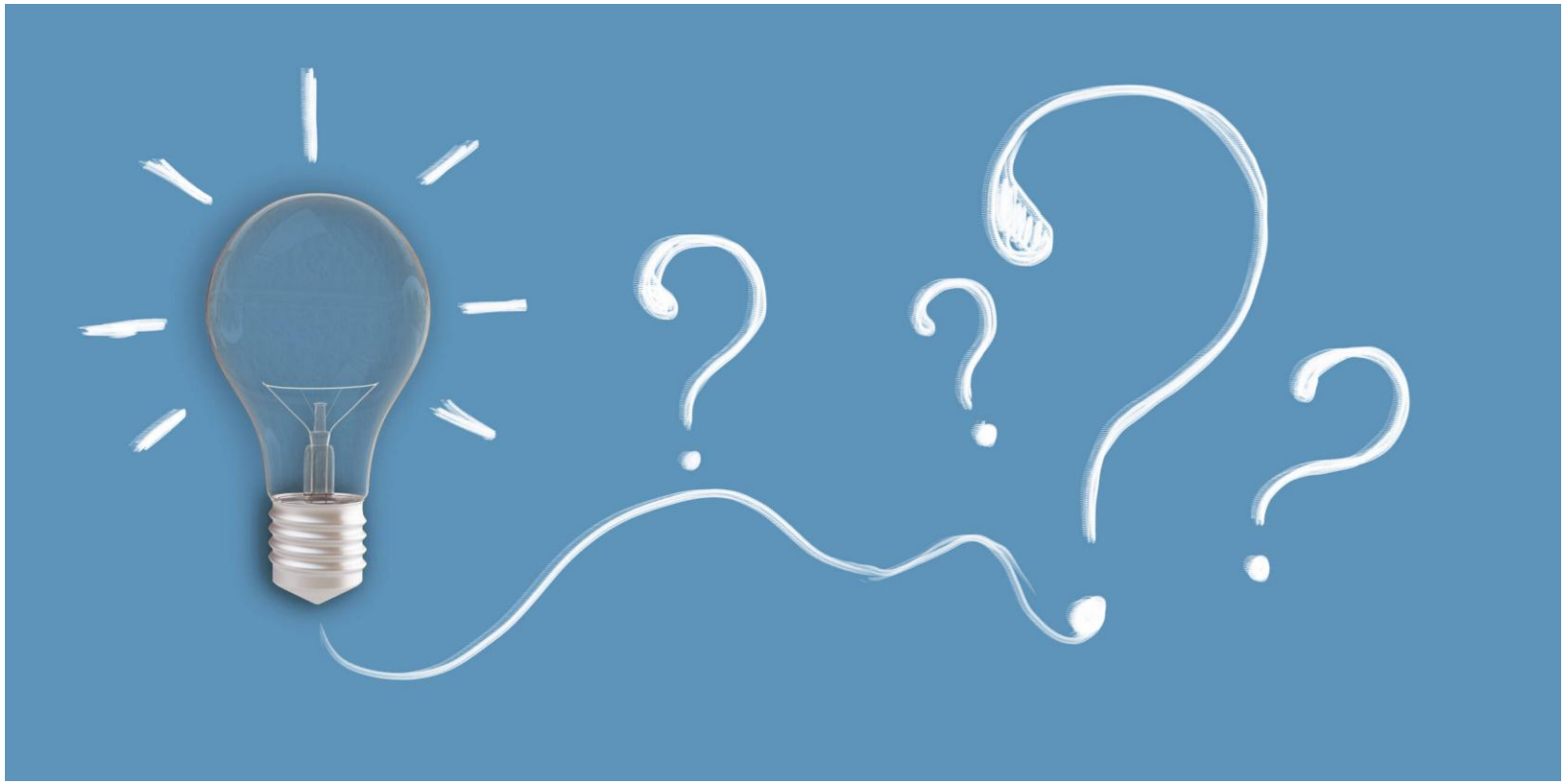
🔟 Definition of Done (DoD)

🟢 **Buenas prácticas:**

- Código revisado por pares.
- Pruebas automatizadas pasando.
- Documentación mínima actualizada.
- Integración continua completada.
- Desplegado en ambiente de staging.
- Sin deudas técnicas pendientes.

🔴 **Anti-ejemplos:**

- “Probado por Juan.” → No es sostenible.
- “Funciona en mi máquina.” → No es transparente.
- “Falta solo conectar con la API.” → No está Done.
- “Lo mostramos pero no funciona del todo.” → Ruptura del empirismo.



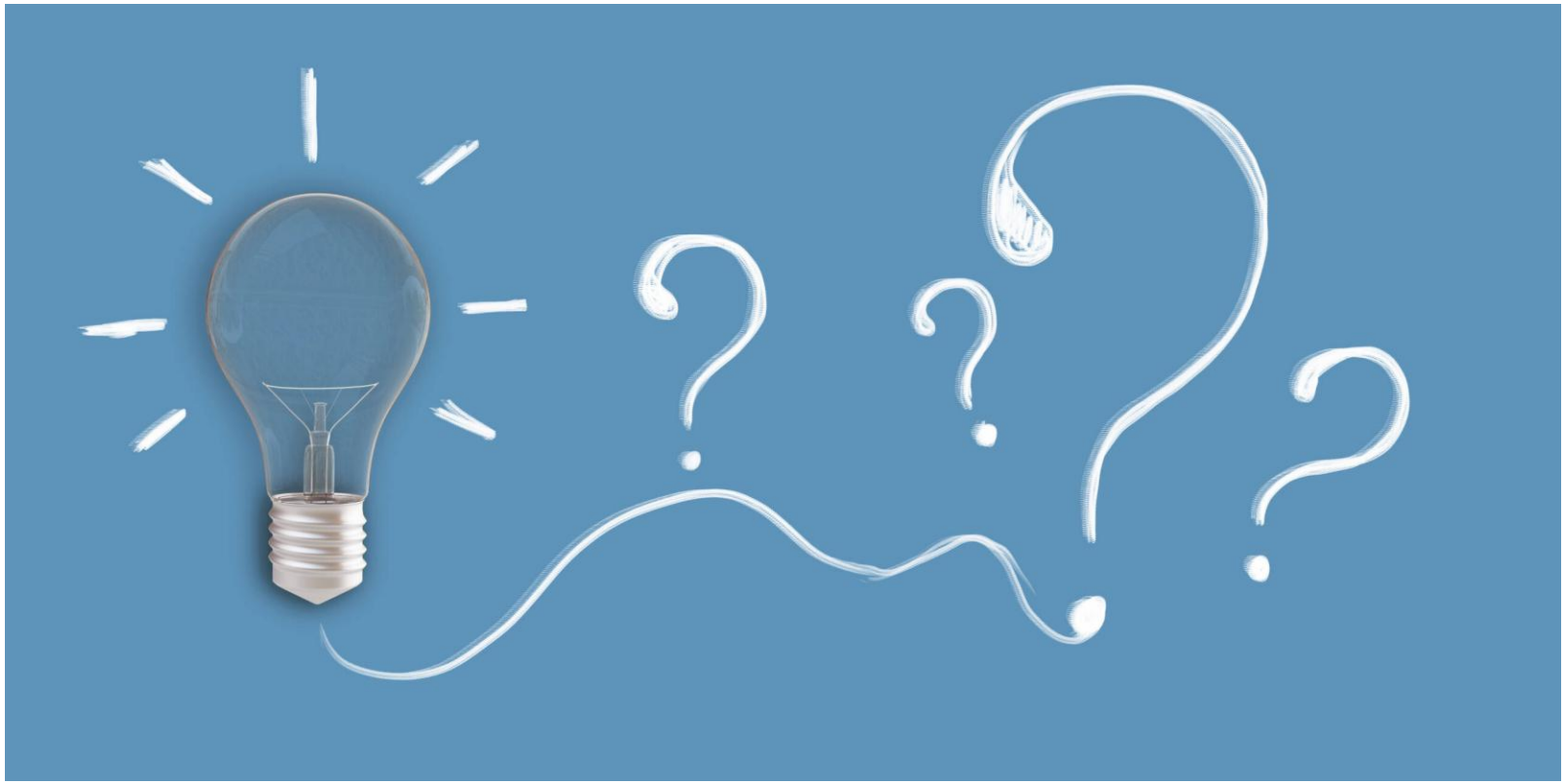
¿Cuál de los siguientes es un Incremento válido?

A. Funcionalidad parcialmente desarrollada pero demostrable.

B. Trabajo que cumple la Definition of Done y es usable.

C. Prototipo visual aprobado por el PO.

D. Código que compila pero aún no ha sido probado.

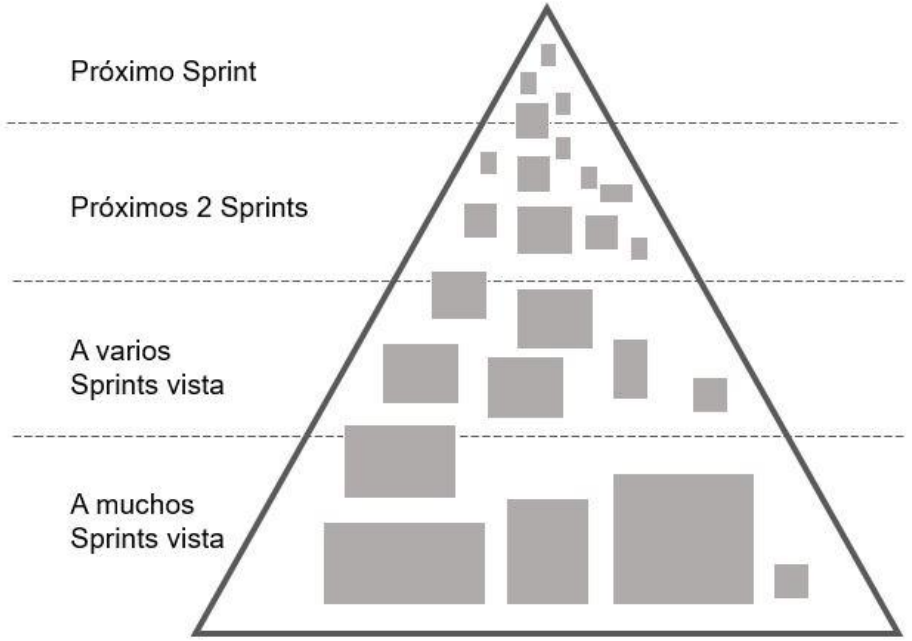


¿Cuál compromiso asegura que el Increment sea verdaderamente transparente?

A. Product Goal
B. Sprint Goal
C. Definition of Done
D. Definition of Ready

1️⃣1️⃣ Refinamiento del Backlog

✔️ El proceso continuo de **clarificar**, **dividir**, **estimar** y **ordenar** ítems del Product Backlog.
✔️ Se hace **durante el Sprint**, no en un evento fijo.
✔️ Involucra al Product Owner y a los Developers.
✔️ Asegura que el backlog esté listo para futuros Sprints.



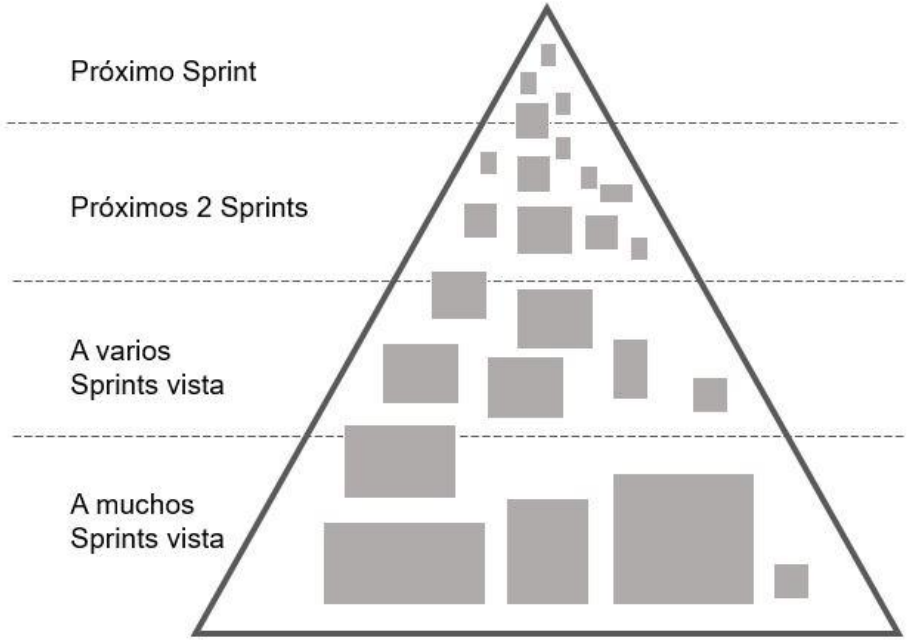
1️⃣1️⃣ Refinamiento del Backlog

🟢 **Buenas prácticas:**

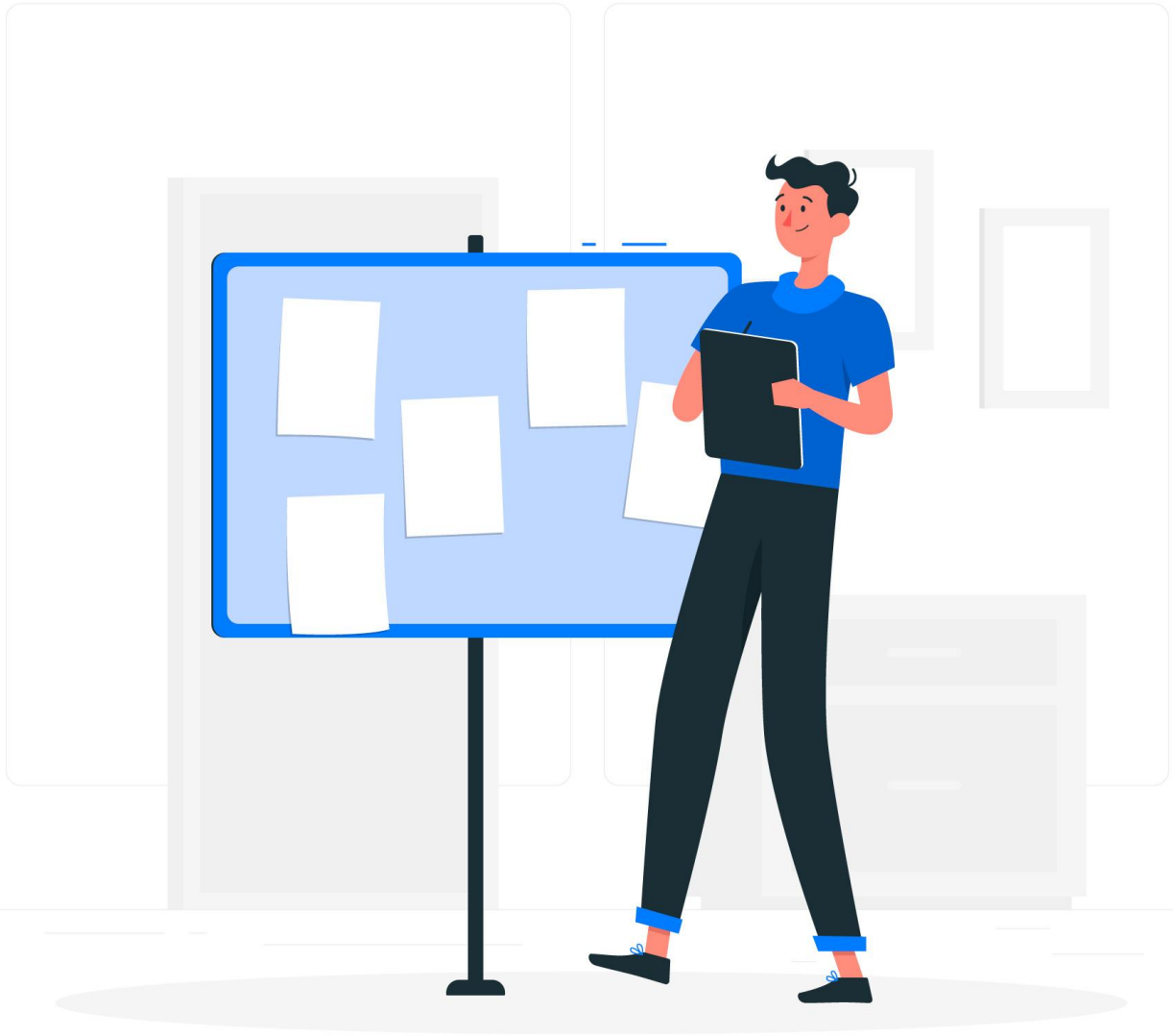
- Convertir requerimientos grandes (“Epicas”) confuso en 5 historias claras.
- Reescribir criterios de aceptación usando
- Descubrir dependencias técnicas antes del Sprint.
- Preparar stories que cumplan con INVEST.

🔴 **Malas prácticas:**

- Refinar solo cuando “falta poco para la Planning”.
- Dejar stories gigantes (“bloques de cemento”).
- PO refinando solo, sin Developers.
- Convertir el refinamiento en una reunión eterna.



EPICAS



1. **Definición**: Gran conjunto de trabajos con un objetivo común.
2. **Características**:
    1. Más grande que una historia de usuario.
    2. Puede abarcar múltiples sprints.
    3. Se divide en historias de usuario más pequeñas para su implementación.
3. **Uso**:
    1. Ayuda a agrupar y organizar requisitos.
    2. Proporciona una visión de alto nivel de una funcionalidad o característica.
4. **Ejemplo**:
    1. Épica: "Mejorar la experiencia de usuario en la plataforma de compras."
        1. Historia 1: "Agregar sugerencias de productos basadas en historial."
        2. Historia 2: "Optimizar el tiempo de carga de la página de productos."
        3. Historia 3: "Implementar una opción de feedback después de la compra."

Como construir un Product Backlog

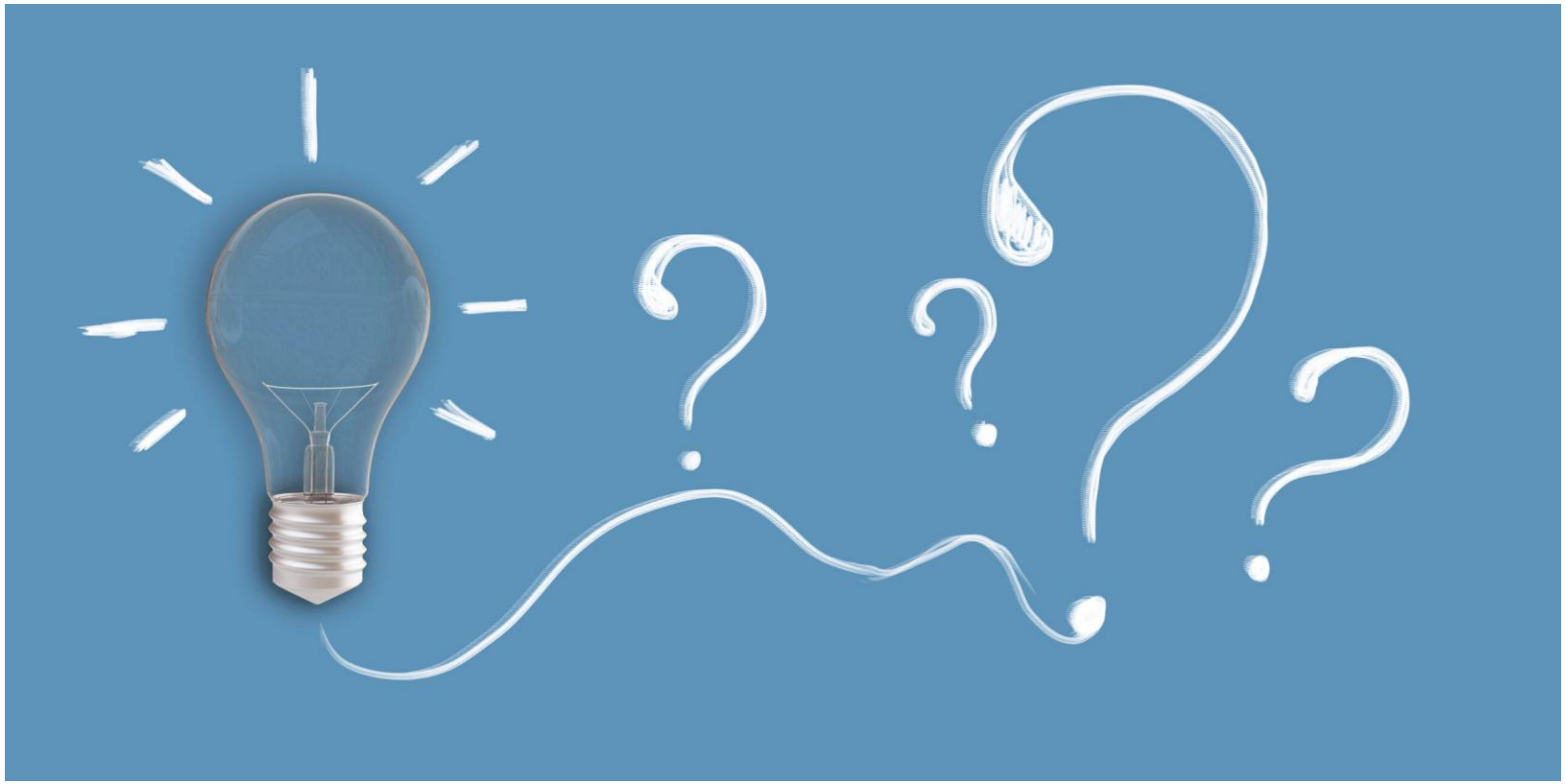


EPICAS

PRIORIZACIÓN

HISTORIAS DE USUARIO

ID	EPICAS	HISTORIAS DE USUARIO
1		
2		



¿Cuál de las siguientes NO es una actividad típica del refinamiento?

A. Dividir historias grandes.
B. Clarificar criterios de aceptación.
C. Estimar historias.
D. Asignar historias a Developers antes del Sprint.

Antes y Después del Refinamiento

ANTES del refinamiento

- HU gigante:
“Implementar checkout completo.”
- Poco entendimiento.
- Sin criterios de aceptación.
- Estimación imposible.
- Alta incertidumbre.
-

DESPUÉS del refinamiento

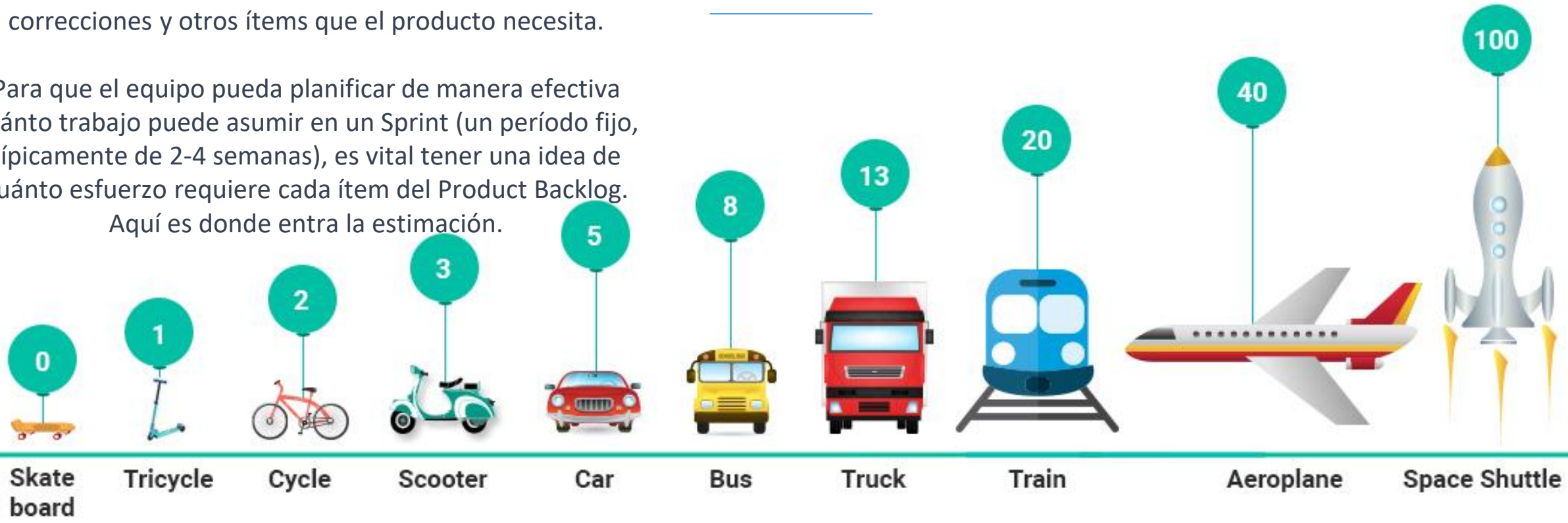
- 4 historias pequeñas y claras:
- Resumen del pedido
- Costo total con impuestos y envío
- Editar cantidades del carrito
- Tiempo estimado de entrega



1 2 Técnicas de estimación ágiles

Scrum es un marco de trabajo ágil que busca facilitar el desarrollo y entrega de productos de alta calidad. Uno de sus componentes es el Product Backlog, que es esencialmente una lista de características, funciones, correcciones y otros ítems que el producto necesita.

Para que el equipo pueda planificar de manera efectiva cuánto trabajo puede asumir en un Sprint (un período fijo, típicamente de 2-4 semanas), es vital tener una idea de cuánto esfuerzo requiere cada ítem del Product Backlog. Aquí es donde entra la estimación.

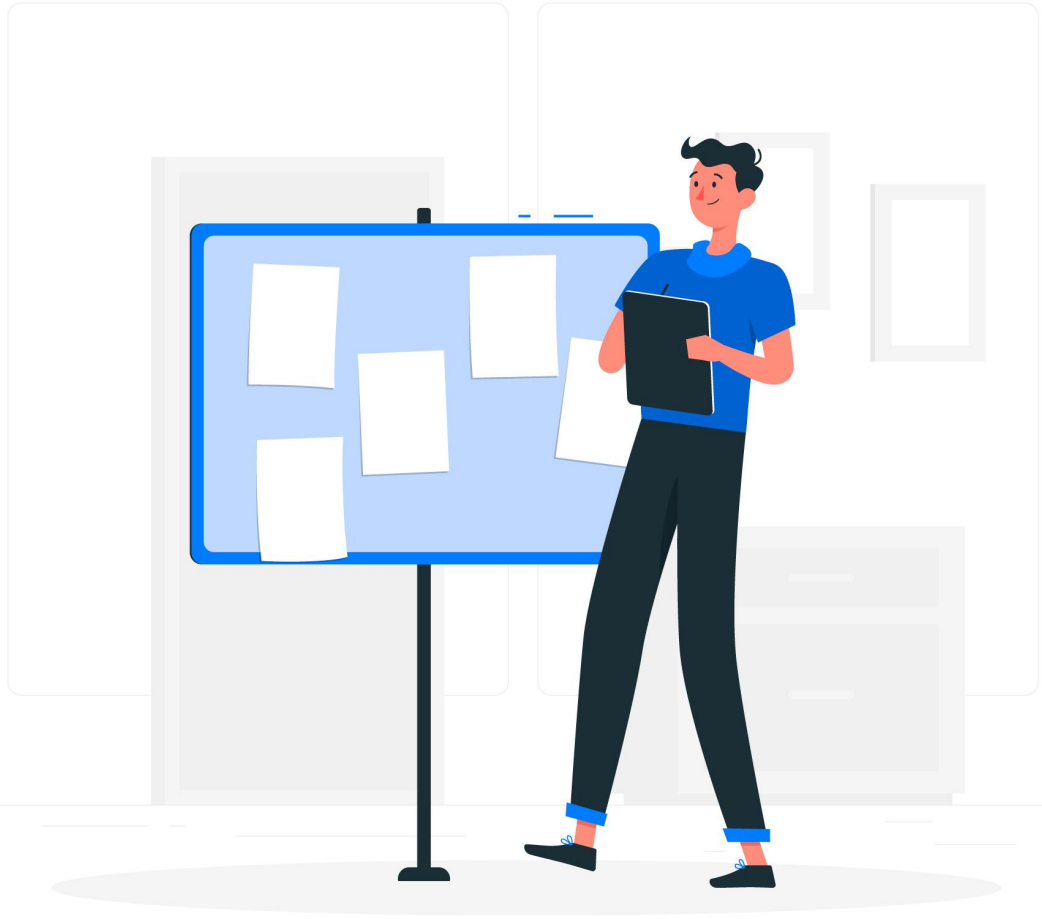


1 2 Técnicas de estimación ágiles



1. **Priorización**: Ayuda a los dueños de producto (Product Owners) a priorizar ítems basándose no solo en el valor sino también en el esfuerzo requerido.
2. **Planificación**: Ayuda al equipo a decidir cuántos ítems pueden completarse en un Sprint.
3. **Transparencia**: Ofrece una visión clara a todas las partes interesadas sobre la magnitud del trabajo.
4. **Colaboración**: Estimular el diálogo entre los miembros del equipo sobre la naturaleza del trabajo.

1 2 Técnicas de estimación ágiles



- **Relatividad**: En Scrum, las estimaciones suelen ser relativas y no absolutas. Esto significa que en lugar de decir "este ítem tomará 10 horas", se podría decir "este ítem es dos veces más complejo que este otro ítem".

- **No es un Compromiso**: La estimación no es una promesa de cuándo se completará un ítem, sino una mejor suposición basada en la información actual.

- **Consistencia**: Aunque las estimaciones no necesitan ser precisas, sí deben ser consistentes. Si un ítem se estima como de "5 puntos" en un Sprint, un ítem de esfuerzo similar debe tener una estimación cercana en futuros Sprints.

Estimación por Esfuerzo

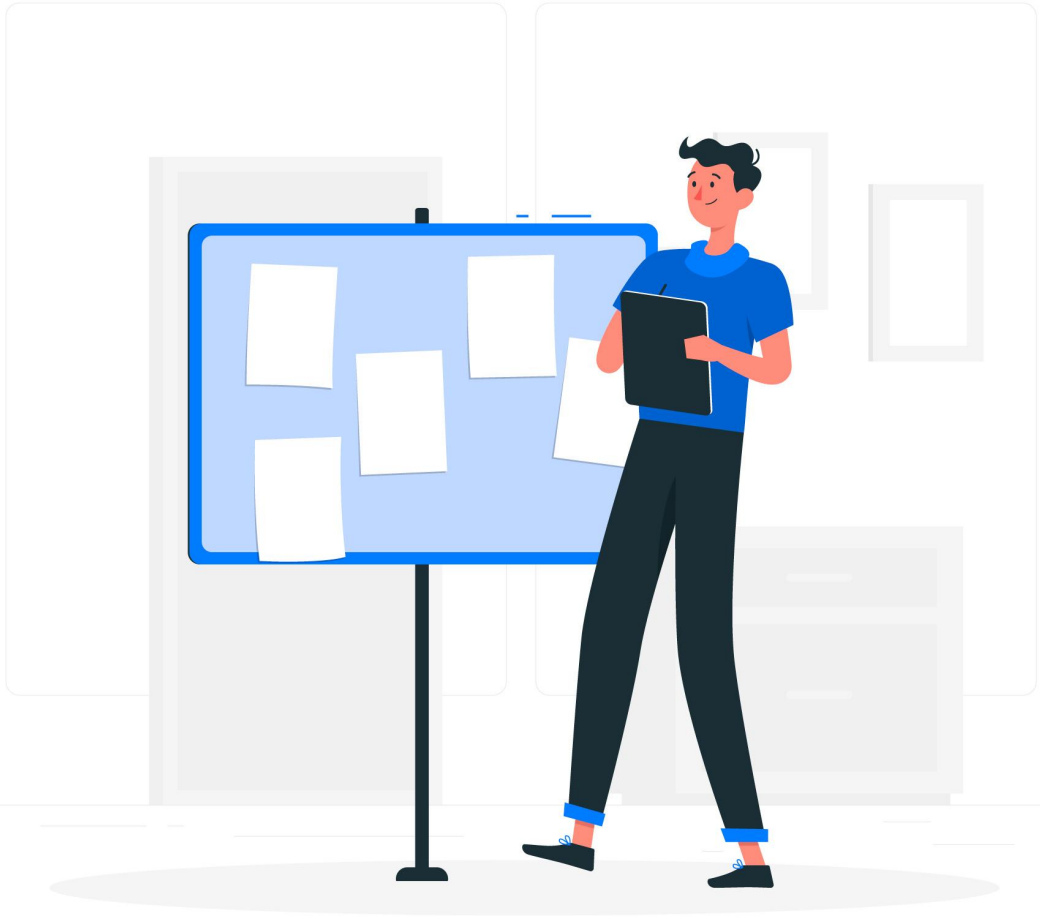


En Scrum, generalmente se prefiere la estimación por esfuerzo en lugar de por tiempo. Aunque a primera vista podría parecer que es más sencillo y directo estimar en horas, hay razones sólidas para enfocarse en el esfuerzo.

¿Qué es la Estimación por Esfuerzo?

La estimación por esfuerzo se refiere a evaluar cuánto trabajo, complejidad o desafío representa un ítem en relación con otros ítems, sin vincularlo directamente a una cantidad fija de tiempo. Por ejemplo, un equipo podría decir que una tarea es "3 puntos de esfuerzo" en lugar de "6 horas".

¿Por qué Estimar por Esfuerzo y no por Tiempo?



- **Variabilidad Individual**: Las personas trabajan a diferentes ritmos. Lo que podría tomar 3 horas a un miembro del equipo, podría tomar 5 horas a otro. Estimando por esfuerzo, el equipo se centra en la carga de trabajo en general, y no en cuánto tiempo le tomará a una persona específica.

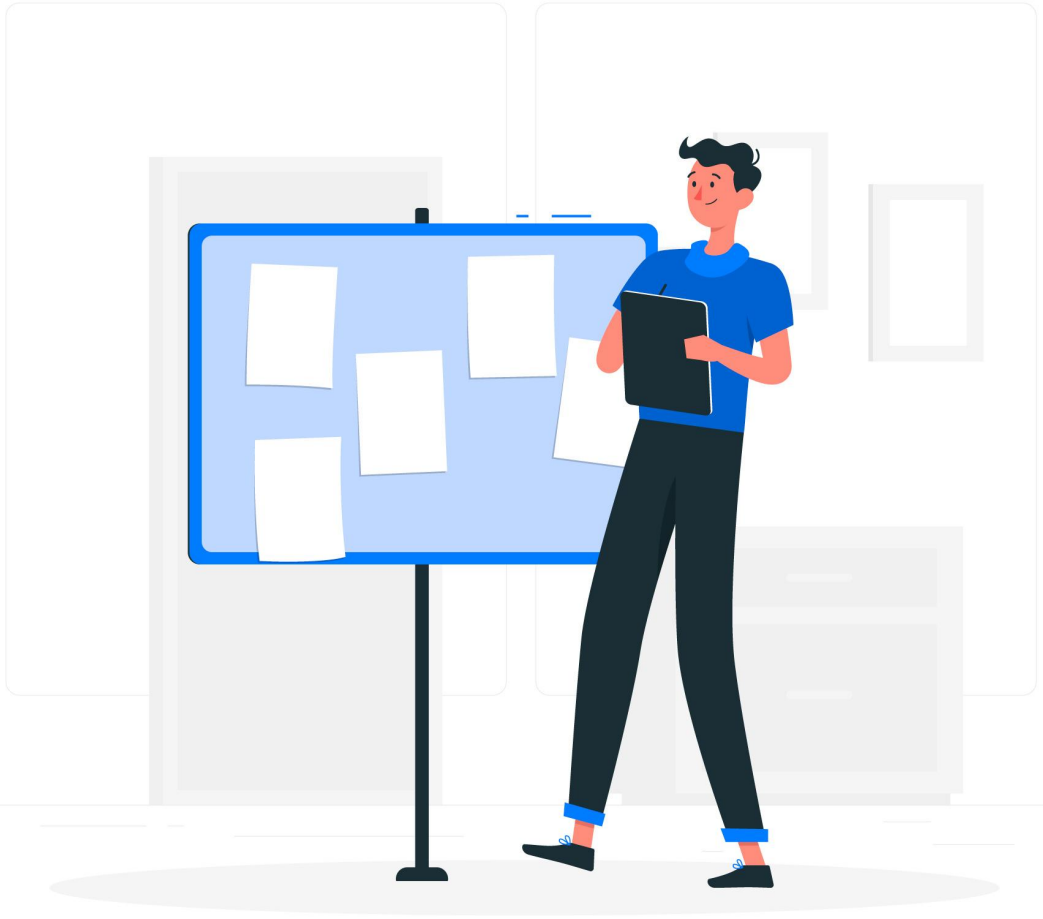
- **Evita la Falsa Precisión**: Estimar en horas puede dar una falsa sensación de precisión. Por ejemplo, decir que una tarea tomará "4 horas y 15 minutos" suena muy exacto, pero en la realidad, muchos factores pueden influir en el tiempo real de ejecución.

- **Facilita la Planificación**: Cuando se estima por esfuerzo, el equipo puede acumular rápidamente puntos de esfuerzo para ver cuánto trabajo pueden asumir en un Sprint, basándose en su capacidad histórica (la velocidad o "velocity").

- **Promueve el Enfoque en el Valor**: Al desvincular la estimación del tiempo, los equipos y las partes interesadas se concentran más en el valor y la prioridad de los ítems que en la duración exacta de las tareas.

- **Adaptabilidad**: Las estimaciones por esfuerzo son más adaptables a cambios. Si un ítem se revela como más complicado de lo esperado, no es necesario reestimar todo en horas; simplemente se ajusta la percepción del esfuerzo relativo.

Comprendiendo el Esfuerzo



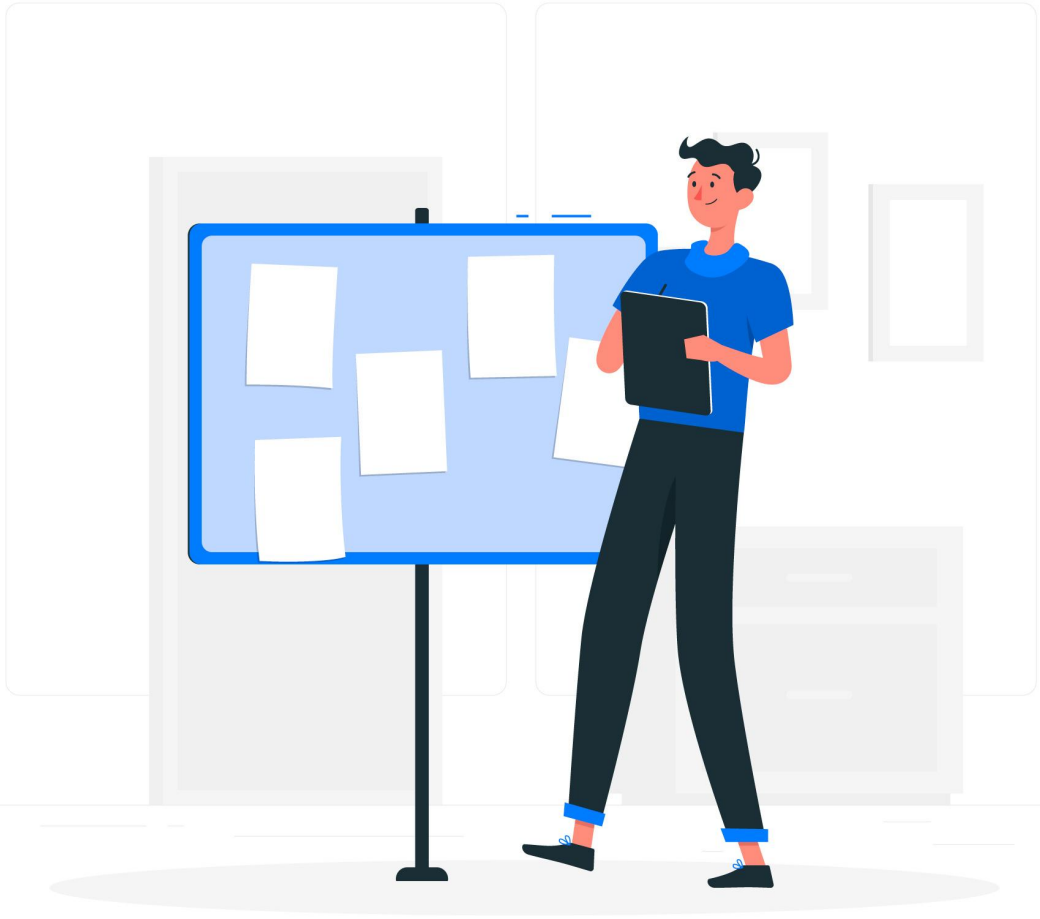
Complejidad: ¿Qué tan complicado es el ítem?

Incertidumbre: ¿Hay muchos desconocidos en este ítem?

Riesgo: ¿Qué posibilidad hay de que algo salga mal?

Trabajo Manual: Claro, el tiempo real que se espera que se invierta también es un factor.

Comprendiendo el Esfuerzo



⭐ **¿Qué son los Story Points? (explicación completa y profesional)**

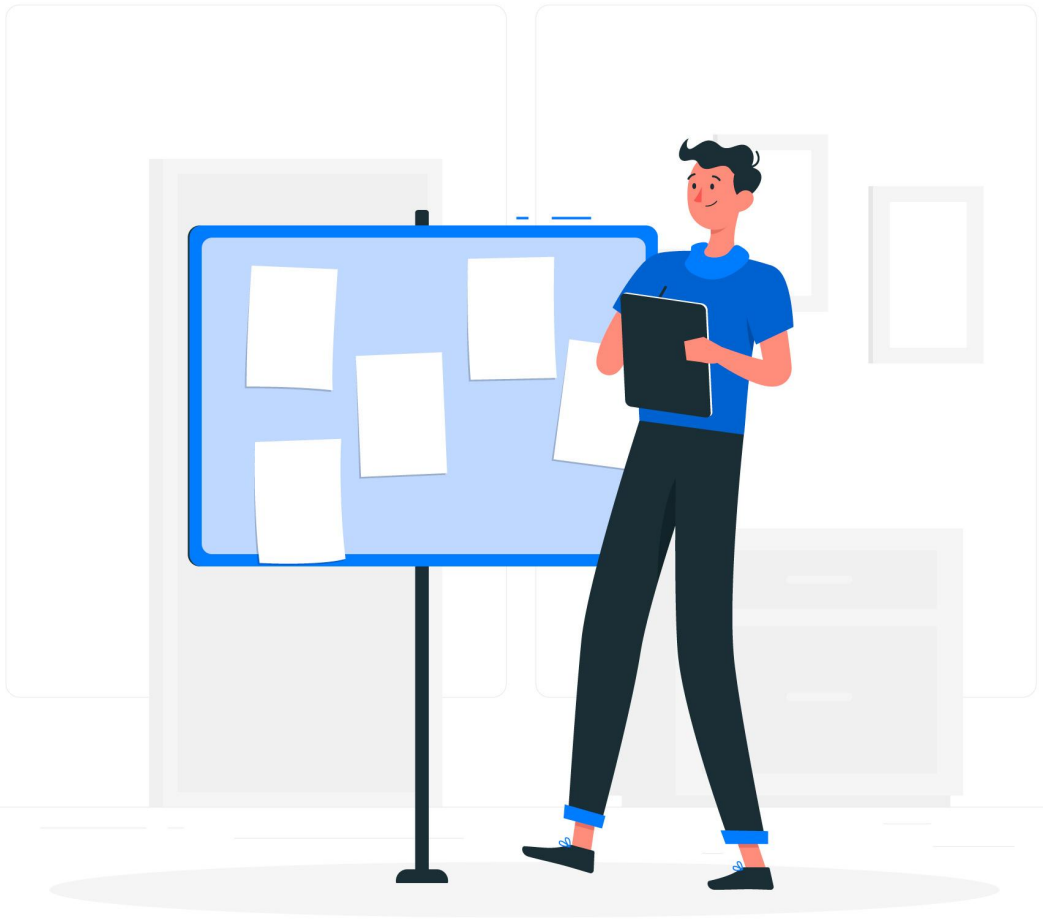
1️⃣ **Los Story Points NO son horas**

Este es el error más común.

Los Story Points no representan tiempo, ni días, ni “esfuerzo humano”.

Representan algo mucho más útil

Comprendiendo el Esfuerzo



🧠 **2 Los Story Points miden *complejidad relativa***

Cuando el equipo estima una historia en *5 puntos*, no está diciendo:

“5 horas”.

Está diciendo:

“Esto es más complejo que las de 3 puntos, pero menos que las de 8.”

Los Story Points capturan 3 cosas:

✔️ **Complejidad técnica**

(lógica, arquitectura, condiciones)

✔️ **Incertidumbre / riesgo**

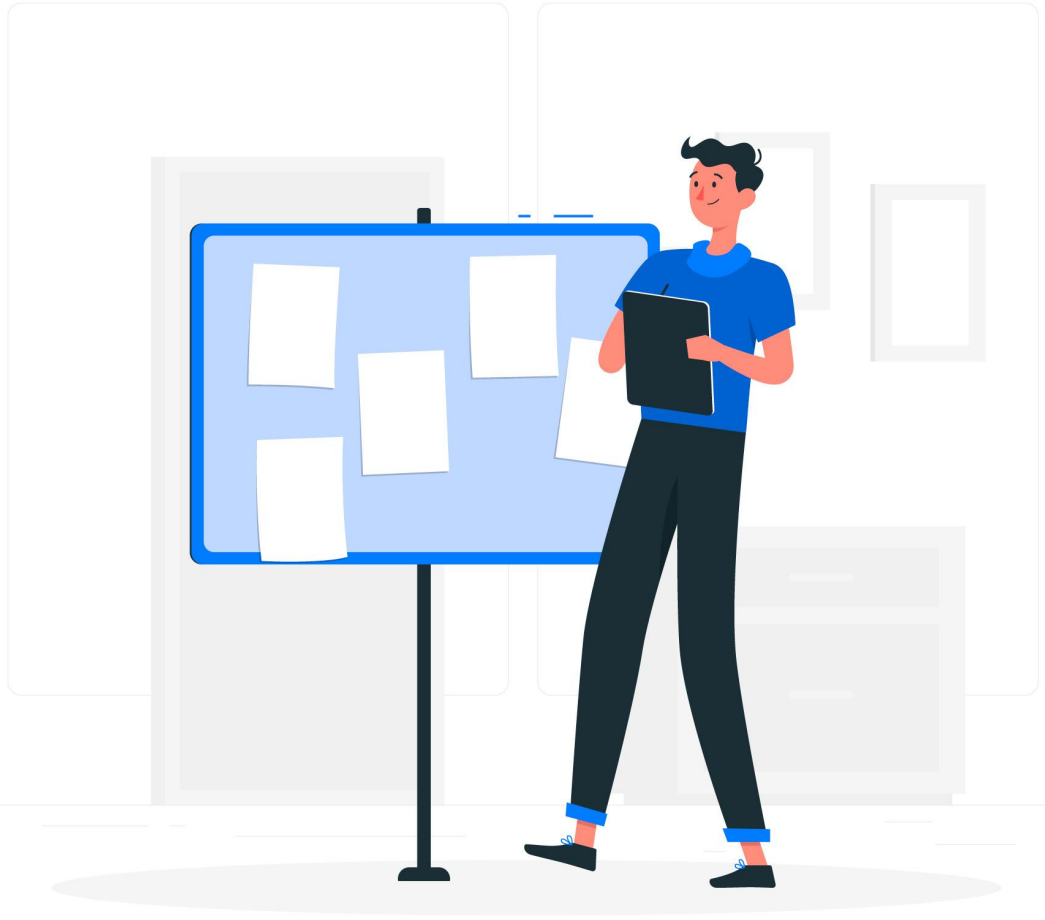
(¿hemos hecho esto antes?, ¿hay dependencias?, ¿tecnología nueva?)

✔️ **Esfuerzo conceptual**

(cuánto debemos pensar, clarificar o probar)

❌ NO capturan horas → porque horas dependen de cada persona, cada día, cada contexto.

Comprendiendo el Esfuerzo



🔥 3️⃣ **¿Por qué se usan Story Points en lugar de horas?**

Porque las horas son engañosas.

Ejemplos reales:

Un desarrollador senior hace algo en 2h.

Uno nuevo lo hace en 8h.

¿Cuántas horas son? → ❌ Inconsistente

¿Es la misma complejidad? → ✔️ Sí

Los Story Points permiten **estimación relativa estable** sin depender del individuo.

🎯 4️⃣ **¿Cómo se asignan Story Points? (Principio clave)**

El equipo elige una historia como referencia (ej. “esto es un 3”).

Luego compara todas las demás historias con esa referencia:

Más pequeña → 1 o 2

Similar → 3

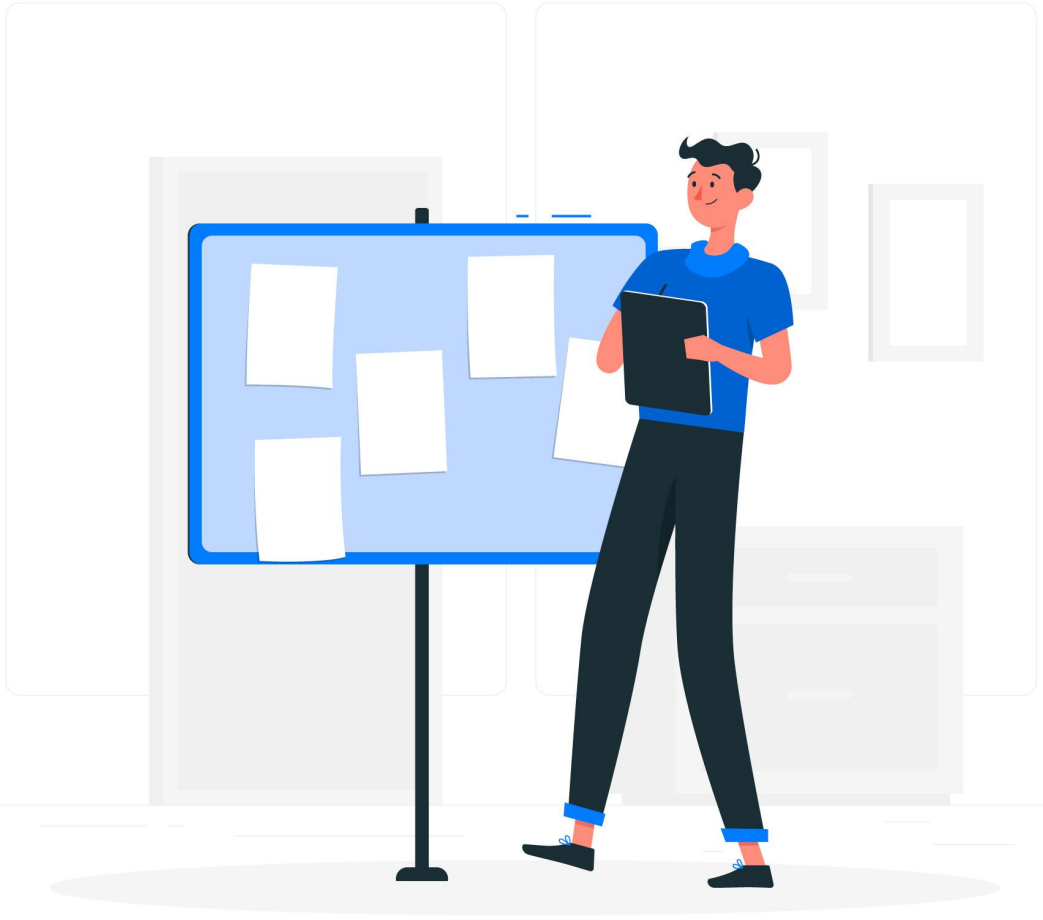
Más compleja → 5

Mucho más compleja → 8, 13

Es exactamente como decir:

“Esto es el doble de complejo que X.”

Comprendiendo el Esfuerzo



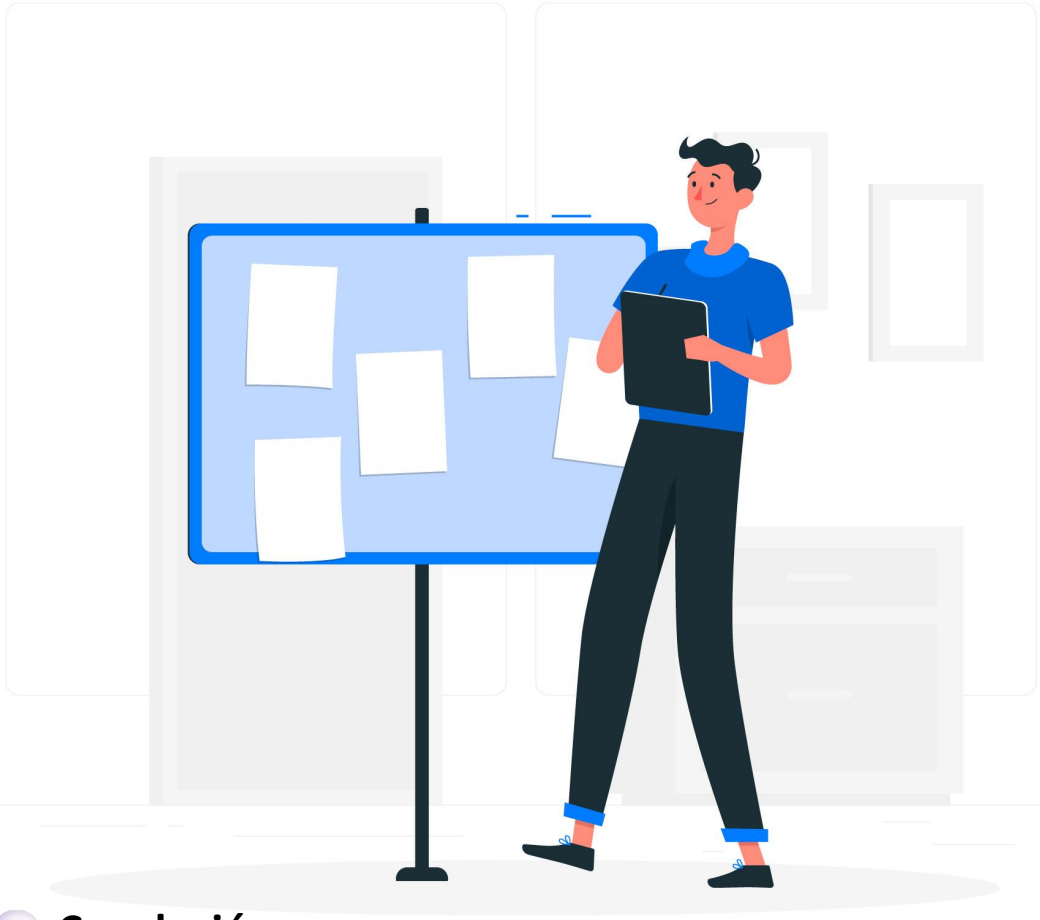
🌀 5️⃣ **¿Por qué la secuencia de Fibonacci?**

Porque ayuda a evitar debates infinitos:
1, 2, 3, 5, 8, 13, 21…
Cada número “salta” lo suficiente para obligar al equipo a elegir:
“¿Es un 5 o es un 8?”
No existe el 6 o el 7 → fuerza conversación.

📈 6️⃣ **¿Qué hacemos con los Story Points después?**

Se usan para:
☑️ **Calcular *velocidad* del equipo**
(promedio de puntos entregados por Sprint)
☑️ **Organizar Sprint Planning**
(sin sobrecargar al equipo)
☑️ **Mejorar predictibilidad**
(sin caer en microgestión ni horas)
Pero OJO:
La velocidad **no es** una métrica de desempeño individual
ni una meta a maximizar.
La velocidad es un *dato*, no un objetivo.

Comprendiendo el Esfuerzo



🧪 7️⃣ **Ejemplo completo**

Historia A: “Como usuario, quiero cambiar mi contraseña.”
Nivel técnico bajo
Pocas dependencias
→ **2 puntos**

Historia B: “Como usuario, quiero ver recomendaciones basadas en historial.”
Alto análisis
Riesgo
Depende de datos
→ **8 puntos**

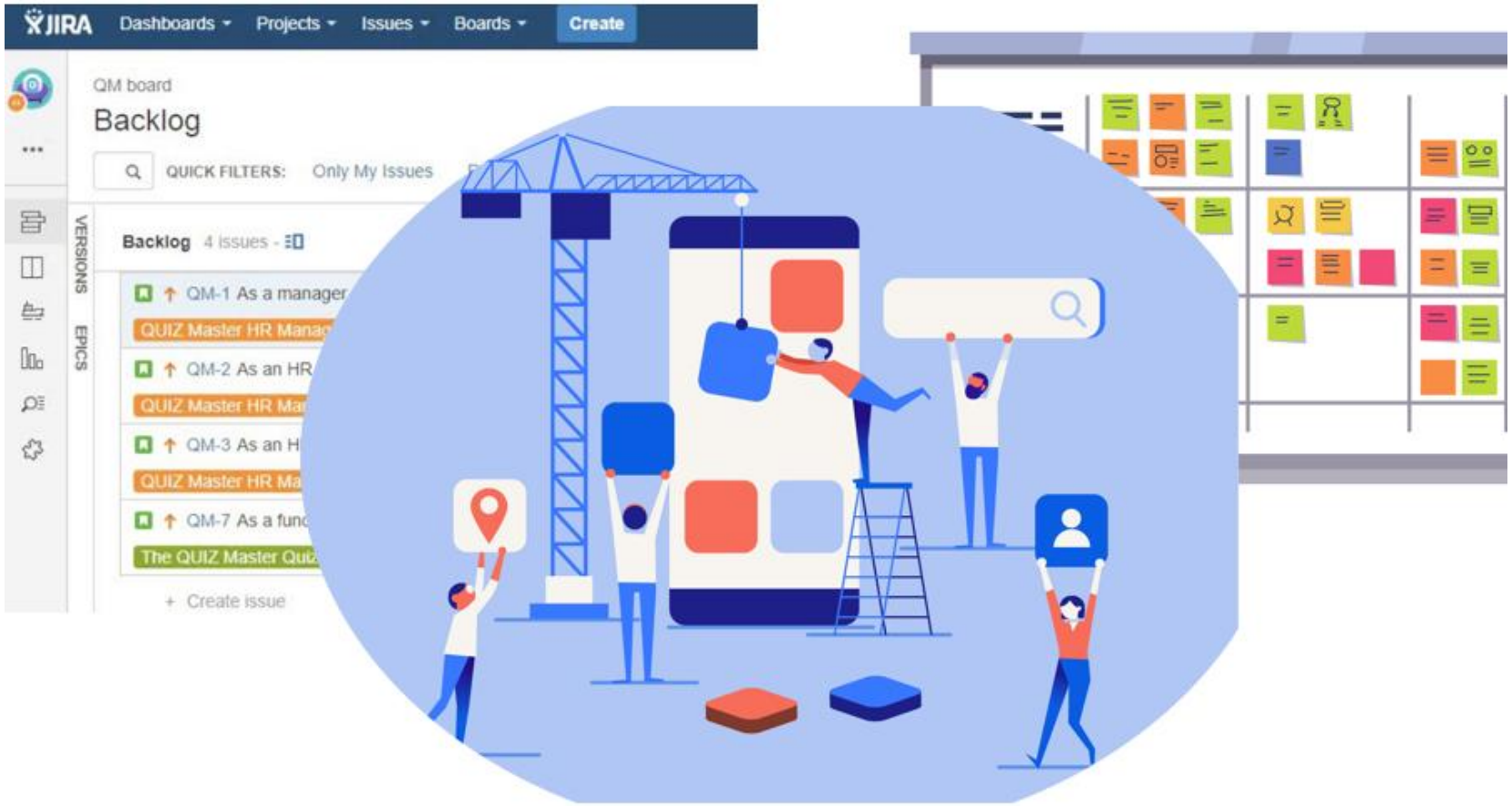
Historia C: “Actualizar botón del checkout.”
Simple
→ **1 punto**

El equipo no pregunta “¿cuánto nos demoramos?”, sino:
“¿Cuánto más o menos complejo es esto comparado con lo que conocemos?”

💬 **Conclusión**

“Los Story Points no intentan predecir horas.
Intentan capturar dificultad, incertidumbre y riesgo.
Si los conviertes en horas, destruyes el propósito.”

Técnicas de Estimación

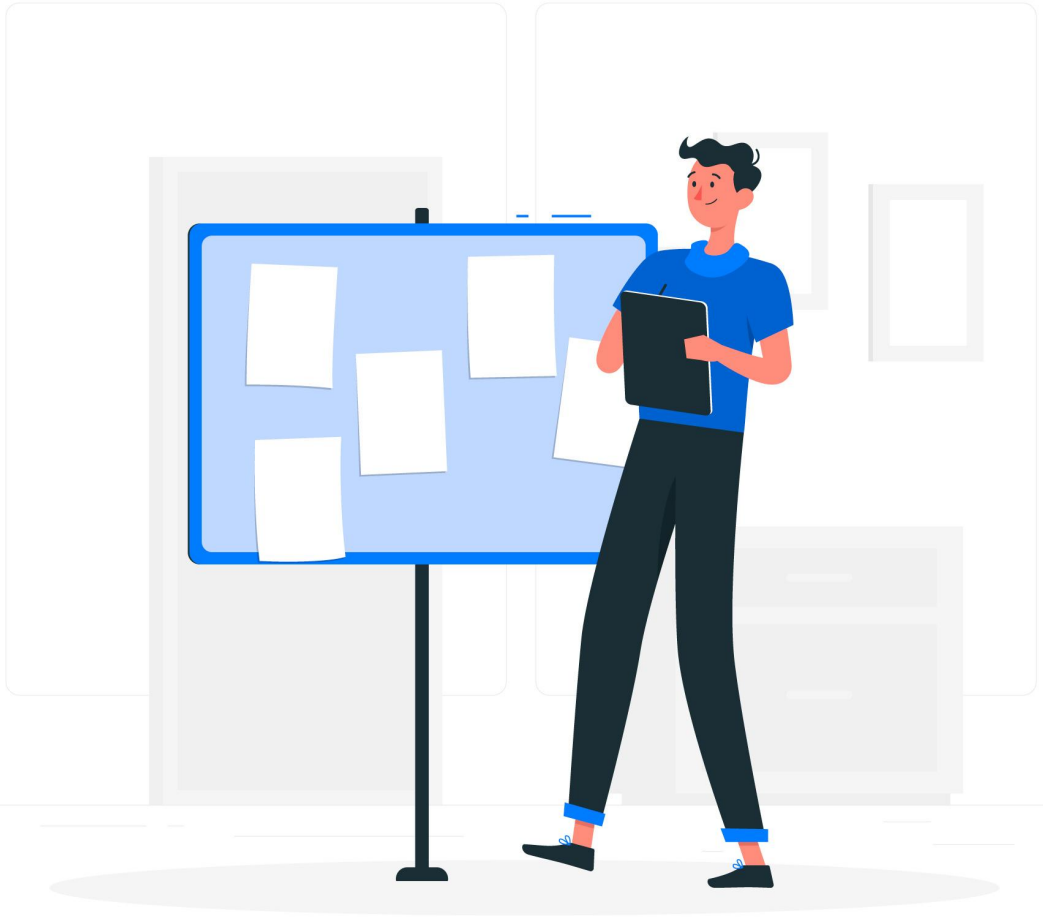


Una vez que se comprende la importancia y el propósito de la estimación en Scrum, es el momento de introducir técnicas específicas.

Las técnicas como Planning Poker, T-Shirt Sizing, entre otras herramientas que los equipos Scrum pueden utilizar para llegar a consensos y proporcionar estimaciones relativas a los ítems del Product Backlog.

El uso y la elección de una técnica específica dependerán del equipo, su naturaleza, el producto y las preferencias. Es común que los equipos prueben diferentes técnicas antes de asentarse en una que funcione mejor para ellos.

Técnicas de Estimación



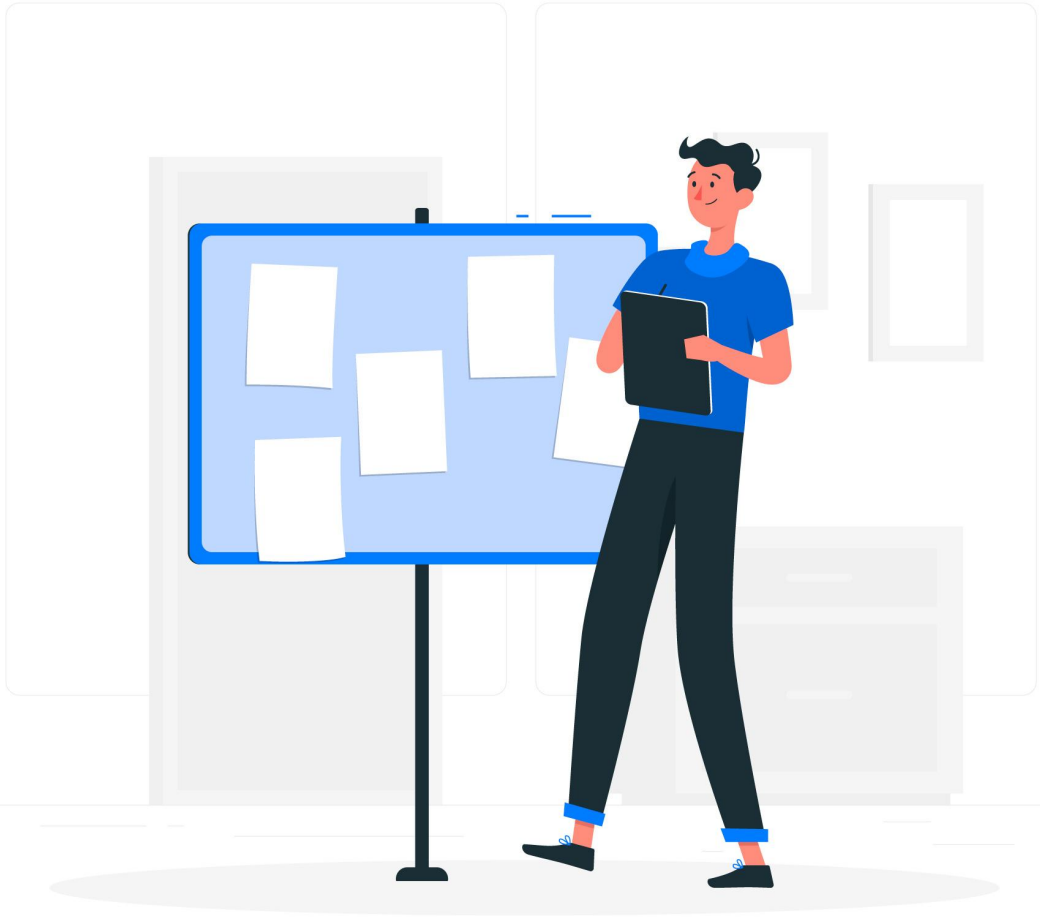
Planning Poker (o Scrum Poker):

1. Cada miembro del equipo recibe un conjunto de cartas con números que suelen seguir una secuencia como la serie de Fibonacci (por ejemplo, 0, 1, 2, 3, 5, 8, 13, …) o cualquier otra secuencia que el equipo encuentre útil.
2. Para cada ítem del Product Backlog, los miembros del equipo seleccionan una carta que represente su estimación de esfuerzo.
3. Todos muestran sus cartas al mismo tiempo.
4. Si las estimaciones varían significativamente, se discuten las razones y se vuelve a estimar hasta llegar a un consenso.

T-Shirt Sizing:

1. Los ítems del Product Backlog se clasifican en tamaños como XS, S, M, L, XL basándose en la complejidad o esfuerzo requerido.
2. Esta es una técnica menos precisa pero rápida y puede ser útil en las etapas iniciales de la planificación.

Técnicas de Estimación



Dot Voting:

1. Se asignan puntos o "dots" a los ítems del Product Backlog.
2. Los miembros del equipo colocan puntos en los ítems que consideran que requieren más esfuerzo. El ítem con más puntos es considerado el más complejo.

Bucket System:

1. Se asigna una cantidad fija de puntos a un "bucket" o cubo.
2. Los ítems del Product Backlog se colocan en diferentes cubos según su estimación relativa de esfuerzo.

Analogía:

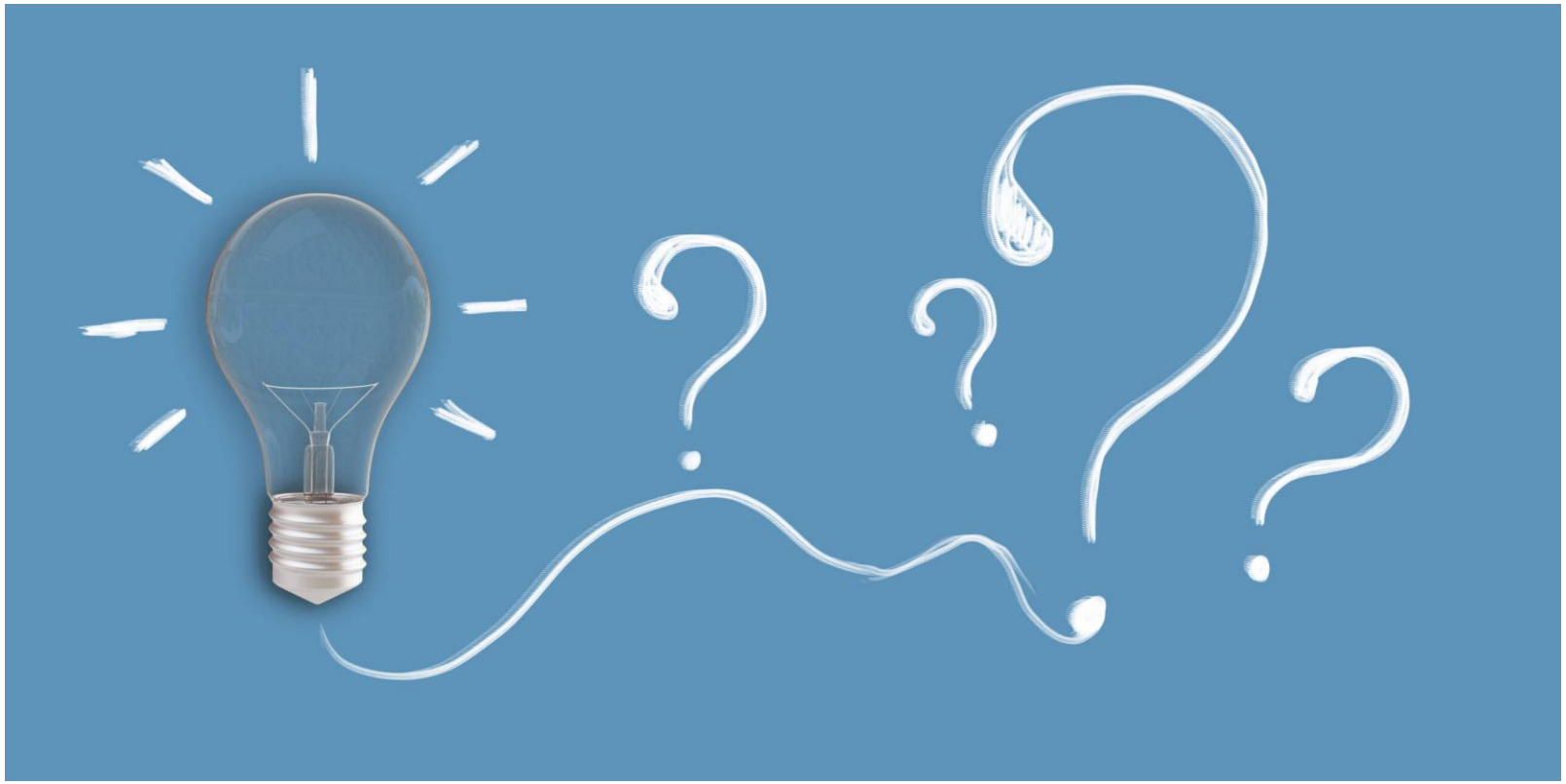
1. Se comparan nuevos ítems del Product Backlog con ítems anteriores que ya han sido estimados y se les asigna un valor similar basándose en esa analogía.

Descomposición:

1. Los ítems grandes o épicos se descomponen en ítems más pequeños y manejables que son más fáciles de estimar.

Estimación de Tiempo:

1. Aunque en Scrum generalmente se prefiere estimar el esfuerzo en lugar del tiempo, en algunos casos, los equipos pueden optar por estimar directamente cuánto tiempo tomará completar un ítem.



¿Cuál enunciado describe mejor los Story Points?

A. Son horas aproximadas de trabajo técnico.

B. Son una medida relativa de complejidad, riesgo e incertidumbre.

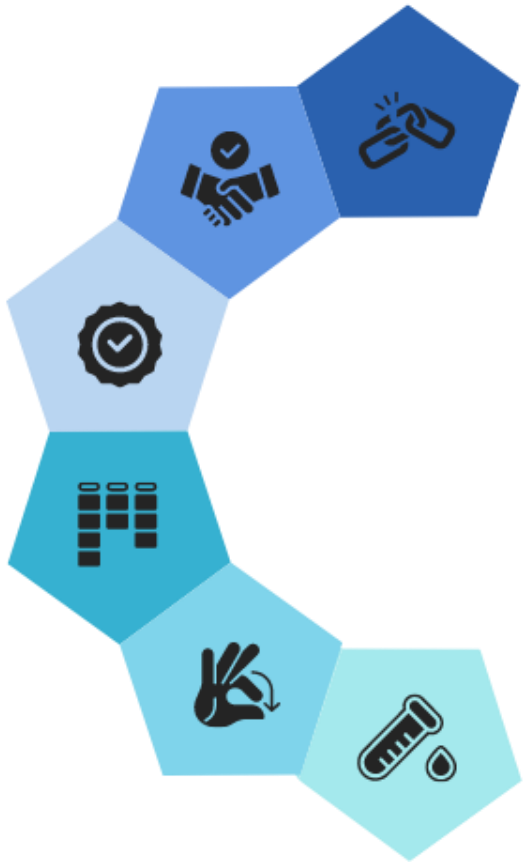
C. Representan el tiempo que debe dedicar cada Developer.

D. Se usan para comparar rendimiento entre equipos.

.

1️⃣3️⃣ INVEST para escribir buenas historias

- ✔ Asegura claridad.
- ✔ Facilita el refinamiento.
- ✔ Evita stories gigantes o ambiguas.
- ✔ Aumenta la calidad del Sprint Planning.



I — **Independiente**
Cada historia es independiente de las otras.

N — **Negociable**
Las historias son flexibles: surgen del diálogo.

V — **Valiosa**
Ofrecen un valor tangible al negocio y los usuarios.

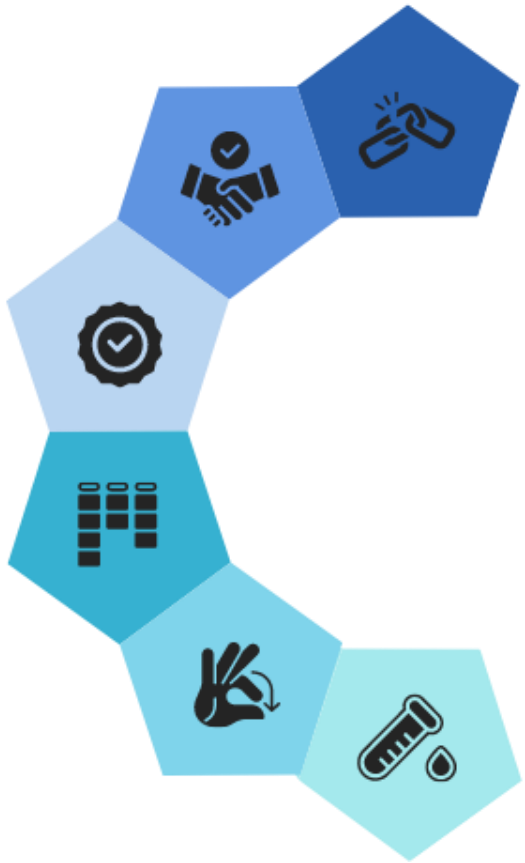
E — **Estimable**
Pueden estimarse.

S — **Pequeña (small)**
Abarcan pocas semanas o días.

T — **Testable (comprobable)**
Pueden comprobarse a través de pruebas.

1️⃣3️⃣ INVEST para escribir buenas historias

I — Independent (Independiente)
“La historia debería poder desarrollarse sin depender fuertemente de otras.
Si todo depende de todo, el equipo se bloquea innecesariamente.”
Ejemplo:
🟢 “Como usuario, quiero guardar mis favoritos para verlos después.”
🔴 “Como usuario, quiero pagar... pero depende de 4 historias más.”



I — **Independiente**
Cada historia es independiente de las otras.

N — **Negociable**
Las historias son flexibles: surgen del diálogo.

V — **Valiosa**
Ofrecen un valor tangible al negocio y los usuarios.

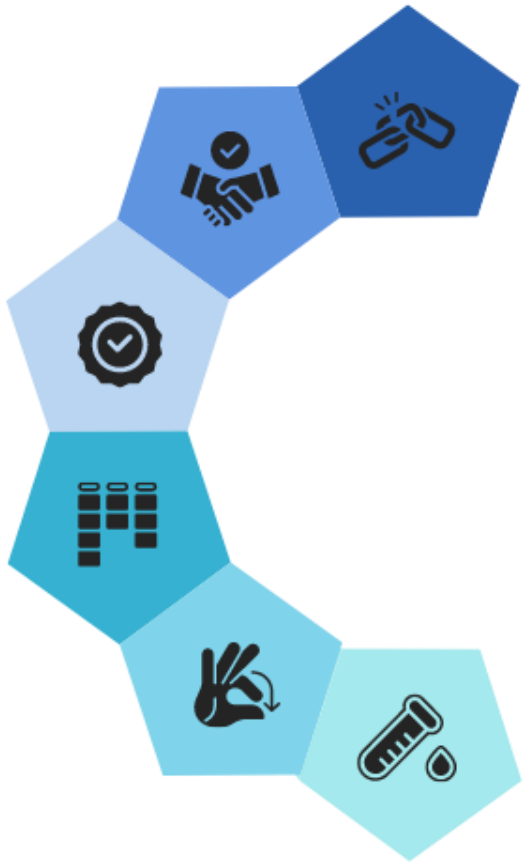
E — **Estimable**
Pueden estimarse.

S — **Pequeña (small)**
Abarcan pocas semanas o días.

T — **Testable (comprobable)**
Pueden comprobarse a través de pruebas.

1️⃣3️⃣ INVEST para escribir buenas historias

N — Negotiable (Negociable)
“No debe ser un contrato rígido.
La historia inicia conversaciones, pero no las reemplaza.”
Ejemplo:
🟢 PO y Developers discuten ajustes sobre el diseño durante el refinamiento.
🔴 Historia que se entrega como un PDF “cerrado” por un stakeholder.



I — **Independiente**
Cada historia es independiente de las otras.

N — **Negociable**
Las historias son flexibles: surgen del diálogo.

V — **Valiosa**
Ofrecen un valor tangible al negocio y los usuarios.

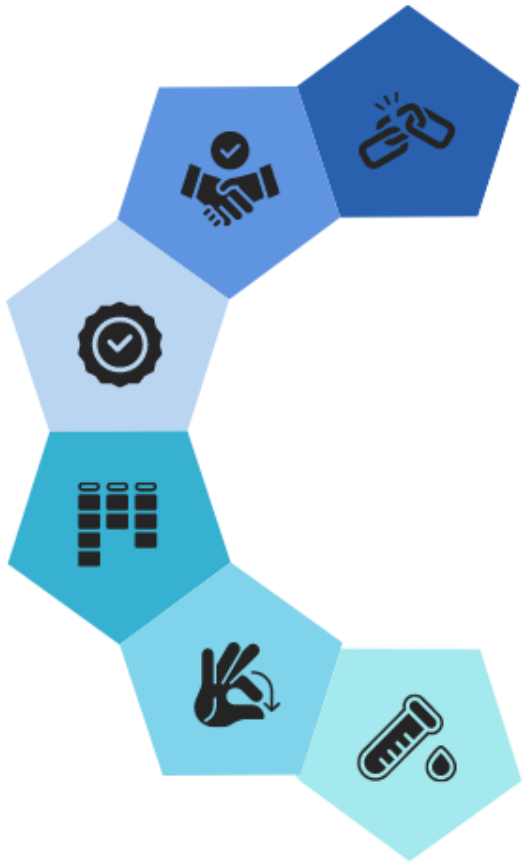
E — **Estimable**
Pueden estimarse.

S — **Pequeña (small)**
Abarcan pocas semanas o días.

T — **Testable (comprobable)**
Pueden comprobarse a través de pruebas.

1️⃣3️⃣ INVEST para escribir buenas historias

V — Valuable (Valiosa)
"Debe entregar valor al usuario, negocio o producto.
Si no aporta valor, es un requisito técnico → se encapsula en otra cosa."
Ejemplo:
🟢 "Como comprador, quiero ver recomendaciones basadas en mi historial..."
🔴 "Crear archivo JSON interno..." (tarea técnica → parte de otra story)



I — **Independiente**
Cada historia es independiente de las otras.

N — **Negociable**
Las historias son flexibles: surgen del diálogo.

V — **Valiosa**
Ofrecen un valor tangible al negocio y los usuarios.

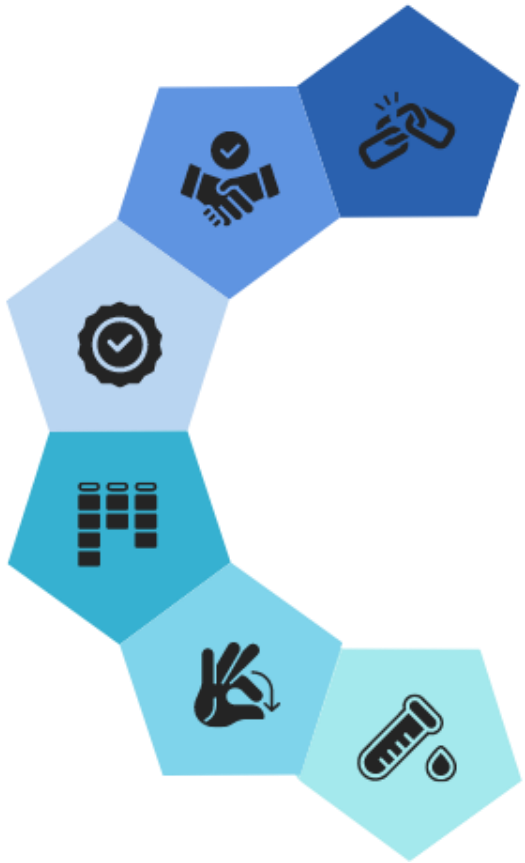
E — **Estimable**
Pueden estimarse.

S — **Pequeña (small)**
Abarcan pocas semanas o días.

T — **Testable (comprobable)**
Pueden comprobarse a través de pruebas.

1️⃣3️⃣ INVEST para escribir buenas historias

E — Estimable (Estimable)
"El equipo debe poder estimarla. Si no se entiende, no se puede estimar → requiere refinamiento."
Ejemplo:
🟢 Historia con criterios claros → se estima en 3 puntos.
🔴 Historia que genera silencio → demasiado ambigua.



I — **Independiente**
Cada historia es independiente de las otras.

N — **Negociable**
Las historias son flexibles: surgen del diálogo.

V — **Valiosa**
Ofrecen un valor tangible al negocio y los usuarios.

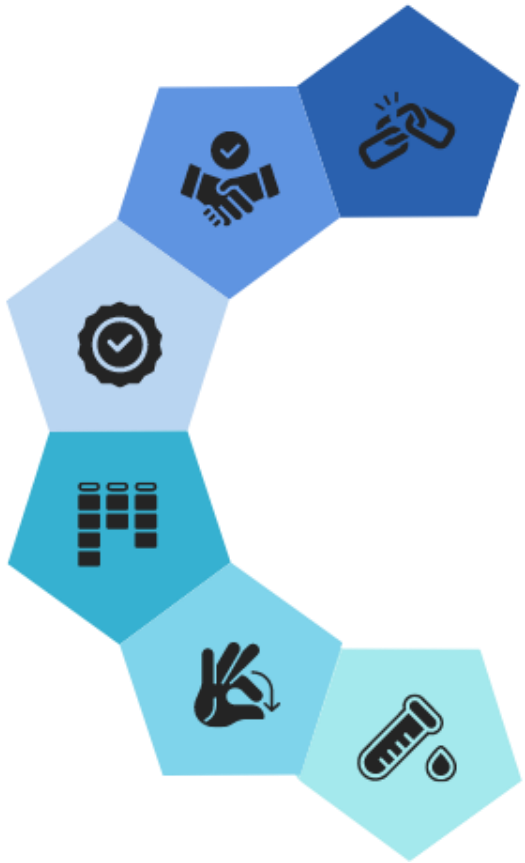
E — **Estimable**
Pueden estimarse.

S — **Pequeña (small)**
Abarcan pocas semanas o días.

T — **Testable (comprobable)**
Pueden comprobarse a través de pruebas.

1️⃣3️⃣ INVEST para escribir buenas historias

S — Small (Pequeña)
"Debe ser lo suficientemente pequeña para completarse dentro de un Sprint."
Ejemplo:
🟢 Story que se completa en 1–3 días.
🔴 Épica disfrazada de user story → requiere división.



I — **Independiente**
Cada historia es independiente de las otras.

N — **Negociable**
Las historias son flexibles: surgen del diálogo.

V — **Valiosa**
Ofrecen un valor tangible al negocio y los usuarios.

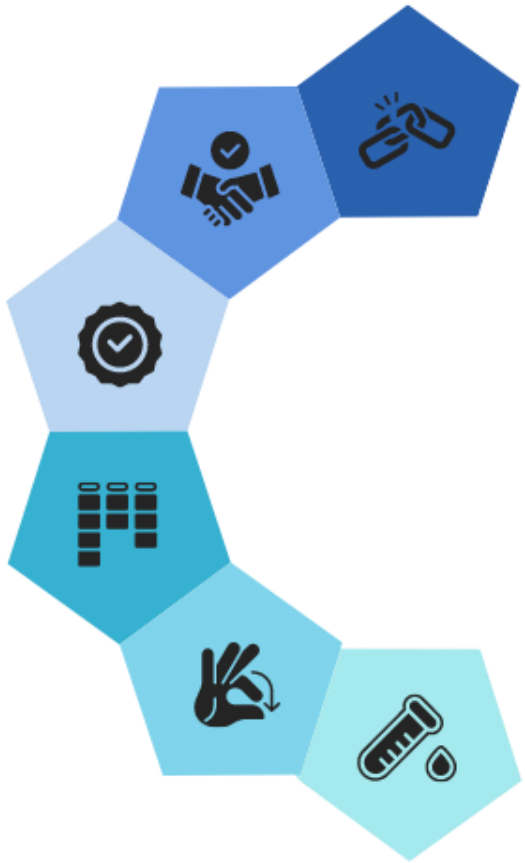
E — **Estimable**
Pueden estimarse.

S — **Pequeña (small)**
Abarcan pocas semanas o días.

T — **Testable (comprobable)**
Pueden comprobarse a través de pruebas.

1️⃣3️⃣ INVEST para escribir buenas historias

T — Testable (Comprobable)
“Debe tener criterios de aceptación claros.
Si no puedo probar si está ‘Done’, no es historia.”
Ejemplo:
🟢 Criterios escritos en formato BDD: Dado / Cuando / Entonces.
🔴 Story sin criterios → imposible validar.



I — **Independiente**
Cada historia es independiente de las otras.

N — **Negociable**
Las historias son flexibles: surgen del diálogo.

V — **Valiosa**
Ofrecen un valor tangible al negocio y los usuarios.

E — **Estimable**
Pueden estimarse.

S — **Pequeña (small)**
Abarcan pocas semanas o días.

T — **Testable (comprobable)**
Pueden comprobarse a través de pruebas.

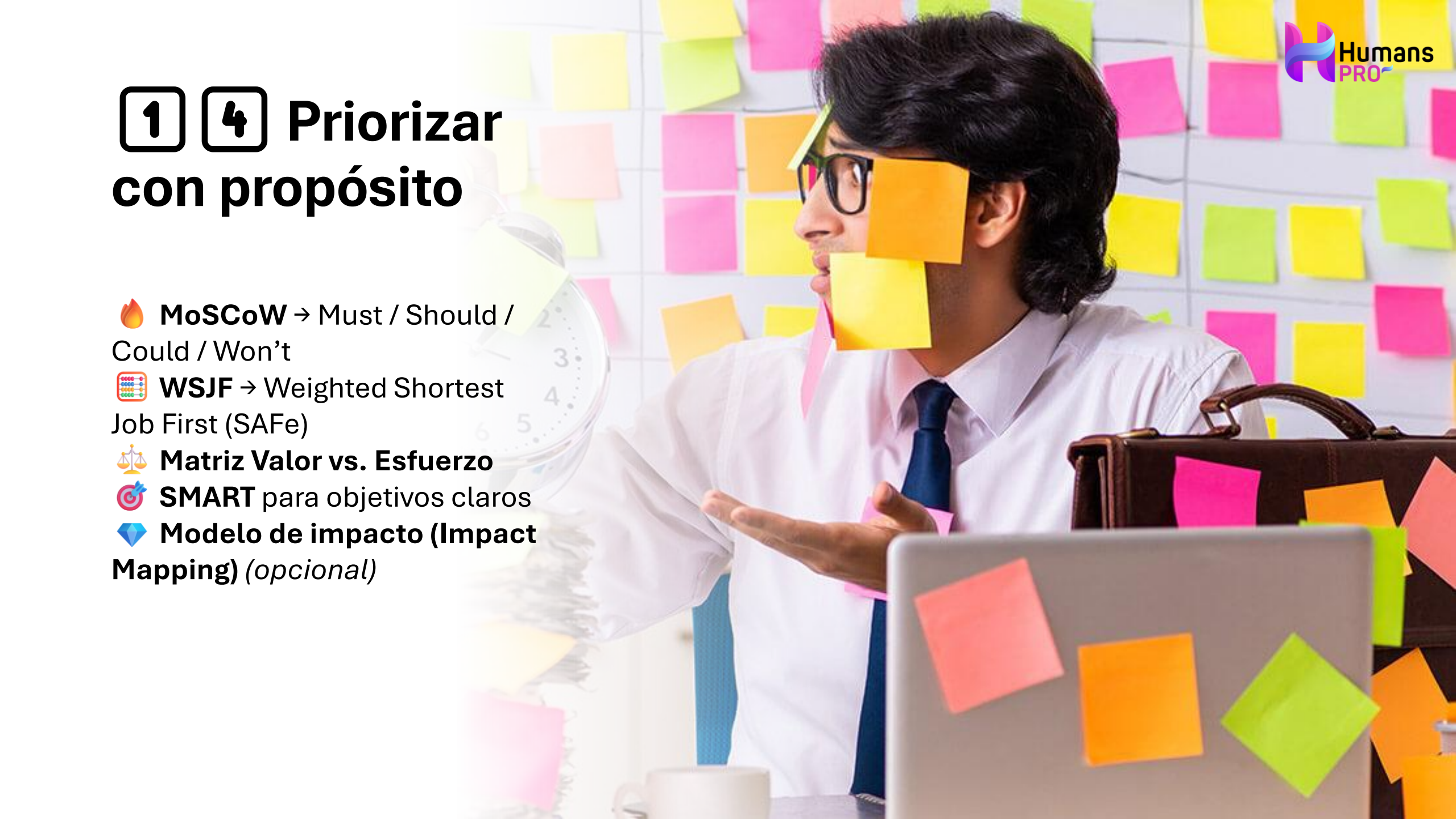
1️⃣4️⃣ Priorizar con propósito

- ✔️ Para maximizar **valor**.
- ✔️ Para reducir riesgo y desperdicio.
- ✔️ Para responder rápido al mercado.
- ✔️ Para alinear expectativas con stakeholders.



1️⃣4️⃣ Priorizar con propósito

🔥 **MoSCoW** → Must / Should / Could / Won’t
🧮 **WSJF** → Weighted Shortest Job First (SAFe)
⚖️ **Matriz Valor vs. Esfuerzo**
🎯 **SMART** para objetivos claros
💎 **Modelo de impacto (Impact Mapping)** *(opcional)*



1️⃣4️⃣ Priorizar con propósito

🔥 **MoSCoW**
“Súper simple y eficaz.
Divide el trabajo en cuatro categorías:
— **Must**: sin esto no hay producto.
— **Should**: muy importante, pero no crítico.
— **Could**: si hay tiempo.
— **Won’t**: no se hará ahora.”

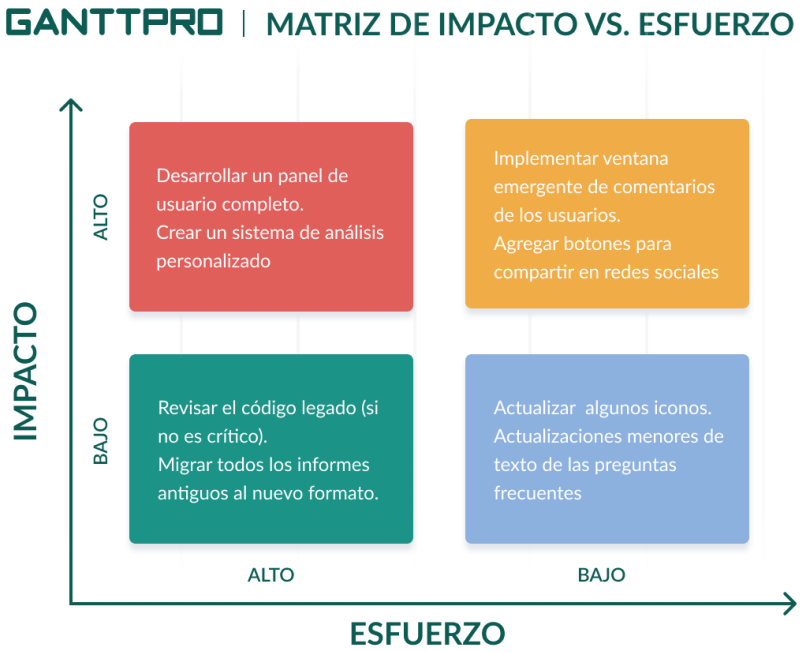


1️⃣4️⃣ Priorizar con propósito



🧮 **WSJF (Weighted Shortest Job First)**
“Sistema más analítico.
Prioriza lo que genera mayor valor relativo respecto a su tamaño.”
Fórmula simplificada:
WSJF = (Valor del negocio + Urgencia + Reducción de riesgo) / Tamaño





1️⃣4️⃣ Priorizar con propósito

- ⚖️ **Matriz Valor vs. Esfuerzo**
- “Visual y simple:
coloca cada historia en un cuadrante según cuánto valor aporta y cuánto esfuerzo requiere.”
- Cuadrantes:
- 🟩 Alto valor / Bajo esfuerzo → hacer ya
- 🟨 Alto valor / Alto esfuerzo → planificar
- 🟧 Bajo valor / Bajo esfuerzo → considerar
- 🟥 Bajo valor / Alto esfuerzo → descartar

1️⃣4️⃣ Priorizar con propósito



🎯 **SMART**

“No es una técnica de priorización, pero sí de claridad.

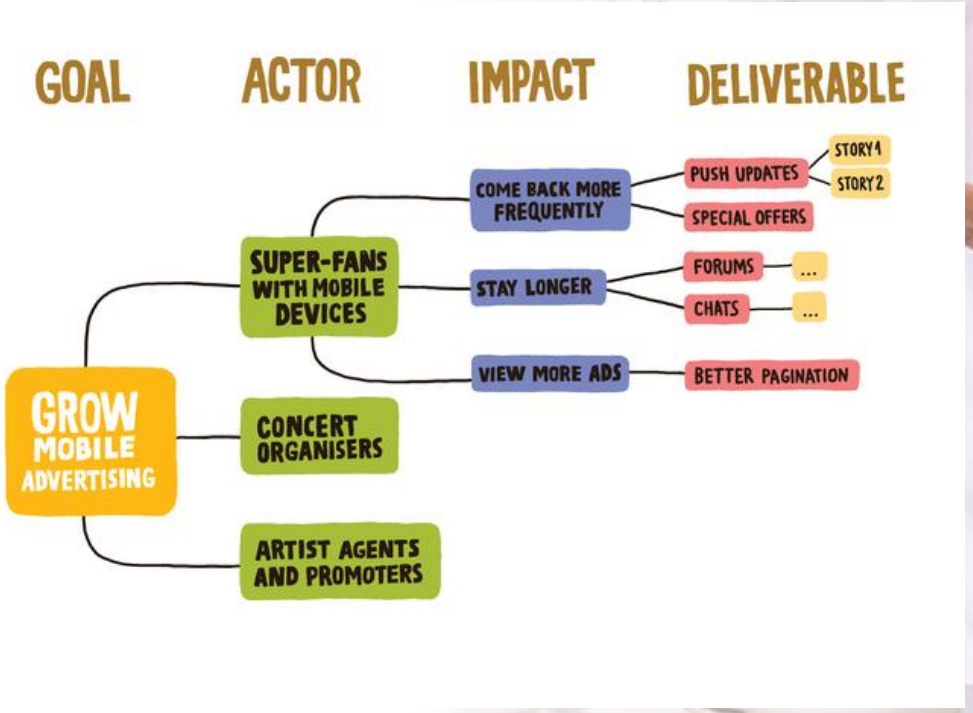
Ayuda a que cada objetivo o historia sea:

— Específico
— Medible
— Alcanzable
— Relevante
— con un Tiempo definido”

1️⃣4️⃣ Priorizar con propósito

💎 **Impact Mapping (opcional)**

“Mapa visual que conecta objetivos → actores → impactos → entregables.”





1️⃣5️⃣ Control empírico del progreso

✔️ Tomar decisiones basadas en **evidencia**, no en deseos.
✔️ Usar datos reales para inspeccionar el progreso.
✔️ Adaptar el plan en función del aprendizaje.
✔️ Mantener transparencia sobre la realidad, aunque sea incómoda.



1️⃣5️⃣ Control empírico del progreso

🟢 **Empirismo bien aplicado:**

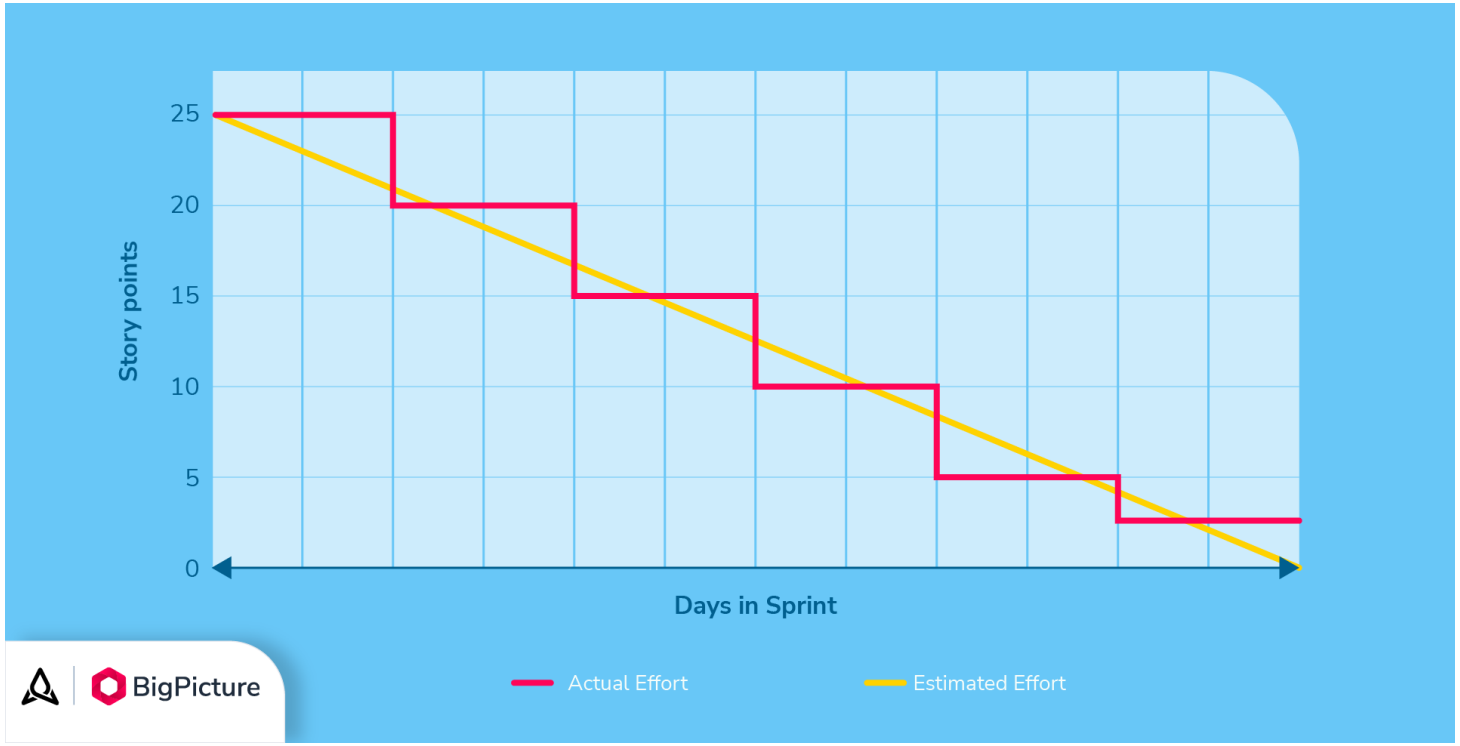
- El equipo detecta en la Daily que está por debajo de ritmo → reorganiza prioridades.
- El burn-down muestra desvío → se revisa el Sprint Backlog.
- El burn-up muestra progreso estable → se discute mejorar capacidad o reducir bloqueos.

🔴 **Mal uso del empirismo:**

- Se usa el burn-down para regañar a un Developer.
- Se manipulan puntos para “cumplir metas”.
- Se reporta avance como horas trabajadas en lugar de Incremento real.

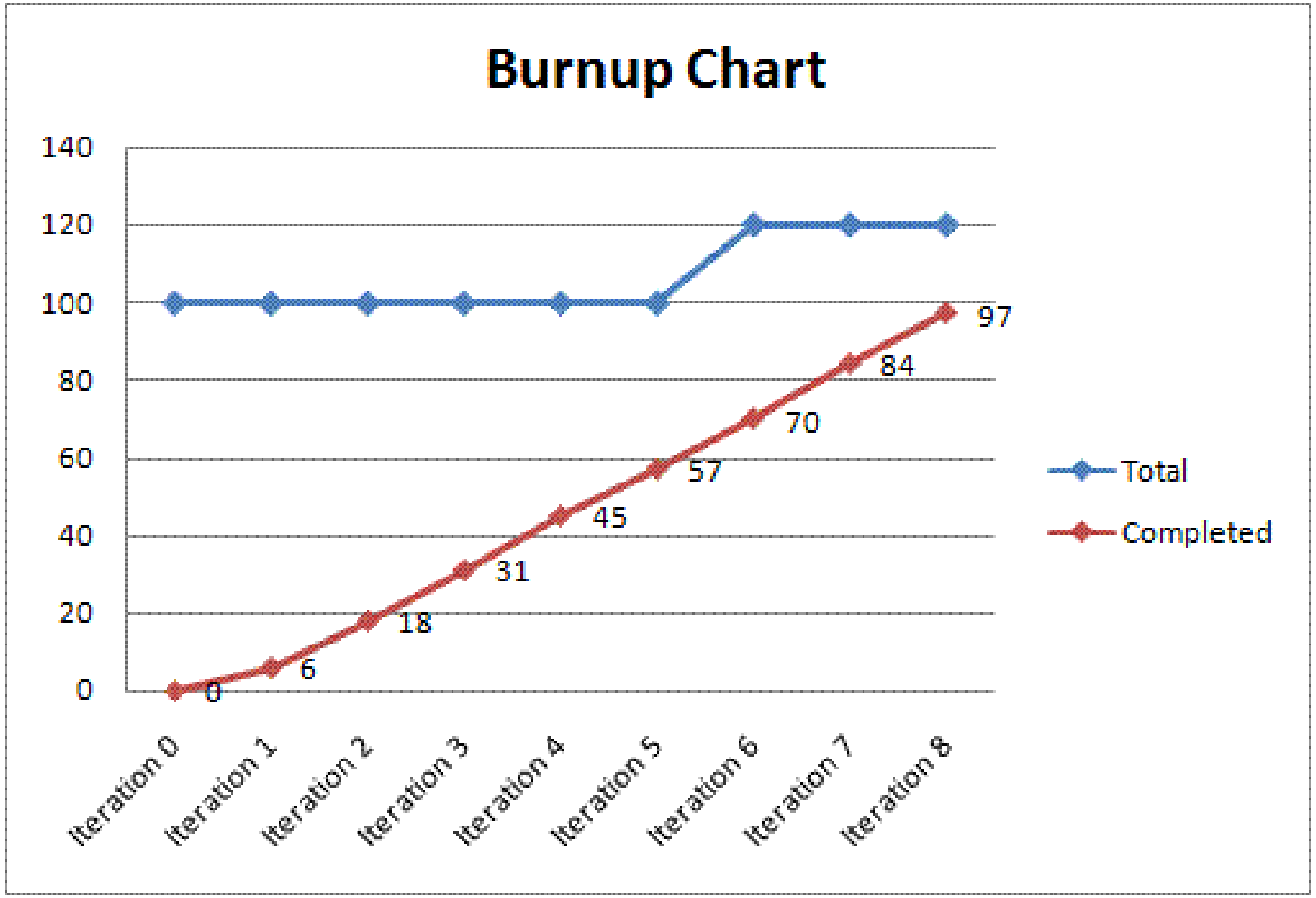
1️⃣6️⃣ Burn-down Chart

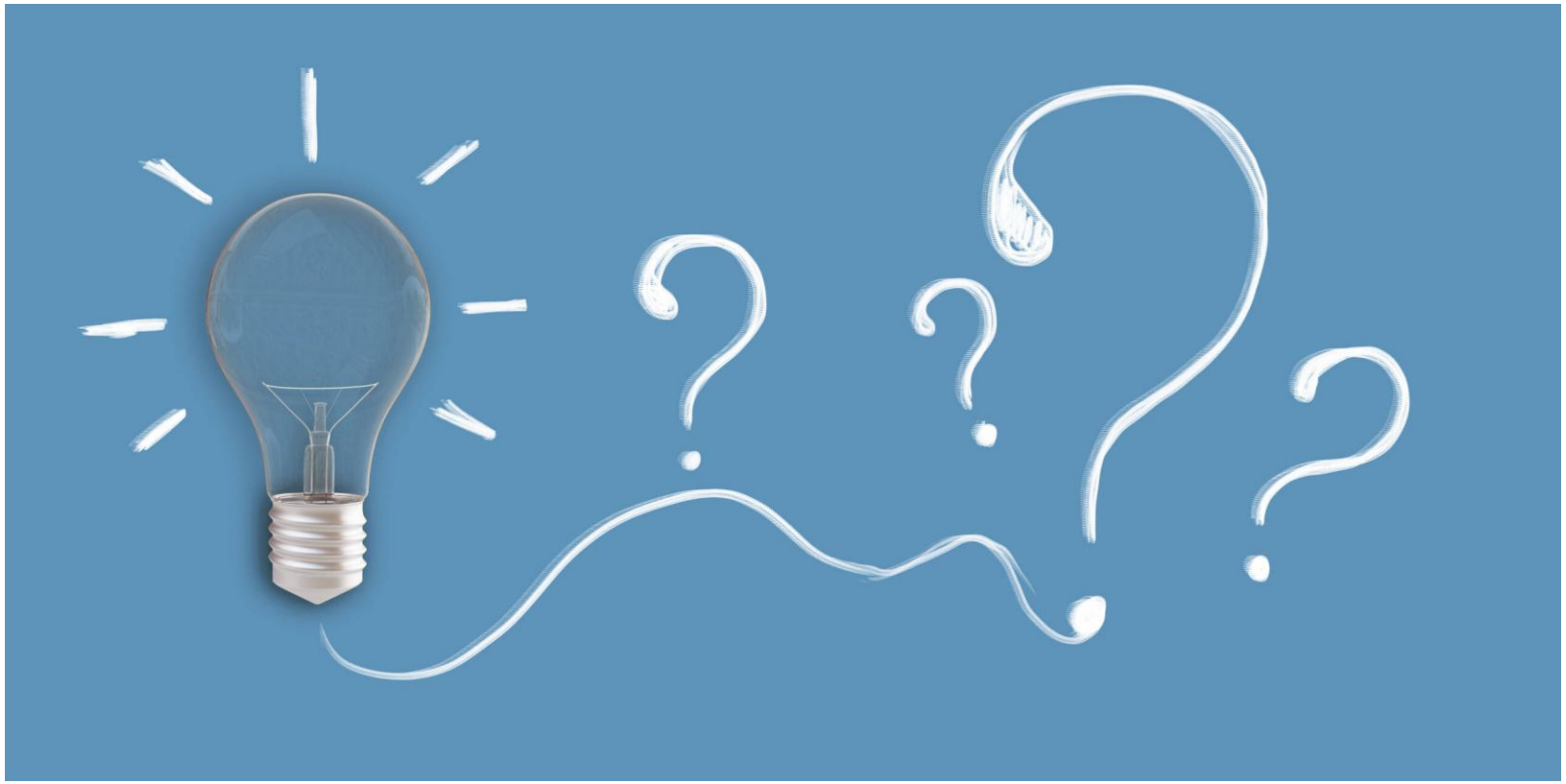
✔️ Un gráfico que muestra **cuánto trabajo queda** en el Sprint.
✔️ Se lee de arriba hacia abajo → de “lo que falta” a “cero pendiente”.
✔️ Ayuda a inspeccionar el **ritmo del equipo**.
✔️ Permite detectar riesgos y desviaciones temprano.



1️⃣7️⃣ Burn-up Chart

✔️ Un gráfico que muestra **cuánto trabajo se ha completado**.
✔️ Incluye la **línea del alcance total**, algo que el burn-down no muestra.
✔️ Hace visibles cambios de alcance, no solo el progreso.
✔️ Es más transparente y menos manipulable que el burn-down.





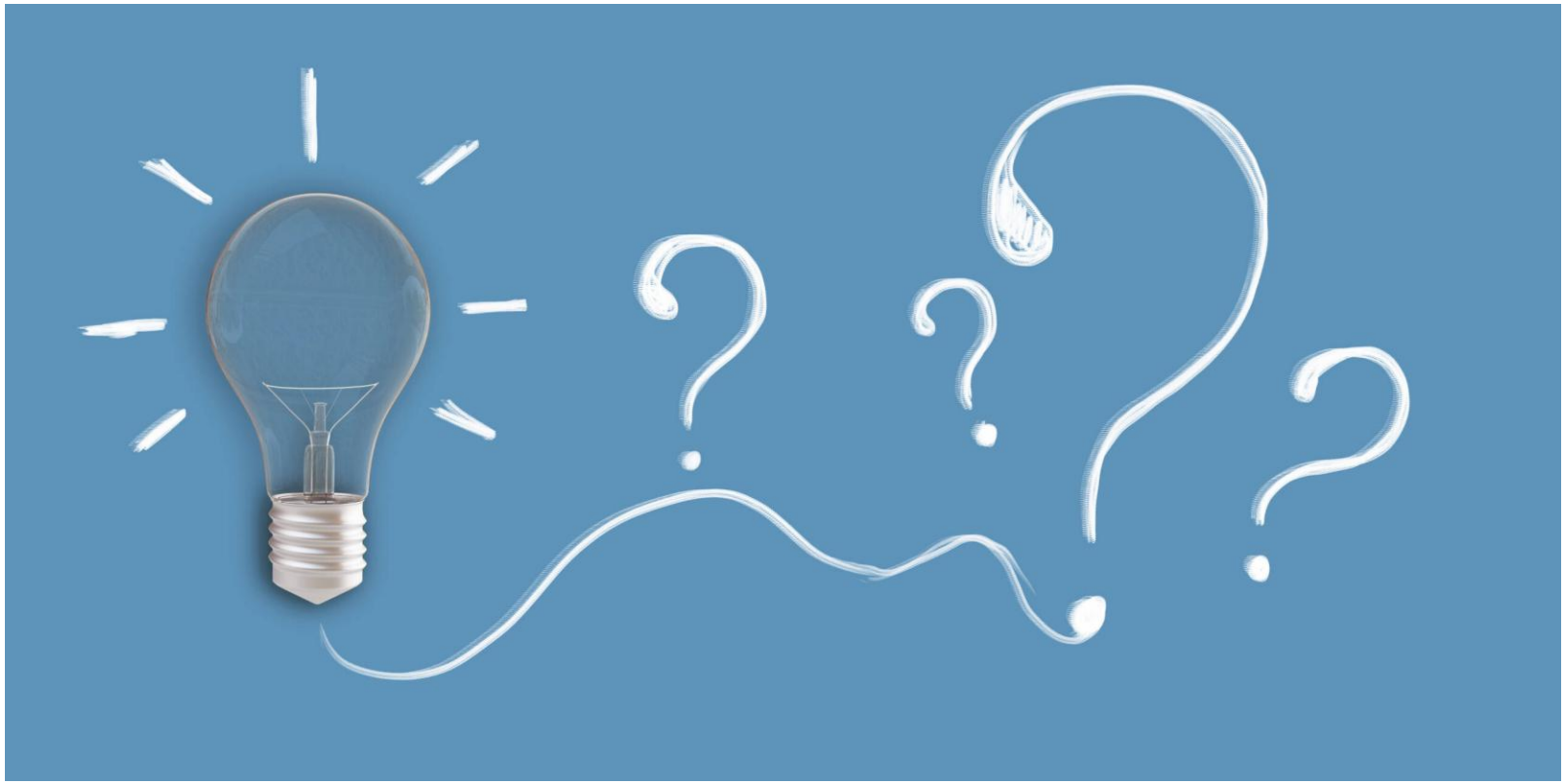
Durante un Sprint, el Burn-down chart muestra que el equipo no está “bajando” la línea de trabajo pendiente, incluso después de varios días. Sin embargo, el Scrum Master observa que los Developers sí están trabajando intensamente en tareas técnicas críticas. ¿Qué puede estar ocurriendo?

A. El equipo no está trabajando en absoluto.

B. El Sprint Backlog no está siendo actualizado con el trabajo real completado.

C. El PO está agregando ítems sin avisar al Scrum Master.

D. El Burn-down chart está mal configurado y debe eliminarse.



En un Burn-up chart, se observa que la línea de alcance total (scope total) sube a mitad del Sprint. ¿Qué indica esto?

A. El equipo incrementó su velocidad sin avisar.

B. Se añadieron elementos al Sprint Backlog sin modificar el Sprint Goal.

C. Se descubrió trabajo adicional necesario y el alcance del Sprint aumentó.

D. Los Developers eliminaron tareas que no querían hacer.



1 8 Práctica: Crear historias de usuario

- ✔ Practicar cómo convertir necesidades reales en historias de usuario.
- ✔ Aplicar el modelo **INVEST** y criterios de aceptación.
- ✔ Detectar ambigüedades, dependencias y riesgos.
- ✔ Crear historias “listas para el backlog”.



El viaje del artefacto durante el Sprint

- **Product Goal** → dirección estratégica
- **Product Backlog** → trabajo priorizado
- **Refinamiento** → historias claras y pequeñas
- **Sprint Planning** → Sprint Goal + selección de trabajo
- **Sprint Backlog** → plan para el Sprint
- **Daily Scrum** → inspección y adaptación del plan
- **Trabajo de Desarrollo** → creación del Increment
- **DoD** → validación de calidad
- **Increment** → valor entregado
- **Sprint Review** → feedback
- **Retrospective** → mejora del proceso

1️⃣9️⃣ Práctica: Estimar con Planning Poker

✔️ Aprender a estimar con **story points** usando Planning Poker.
✔️ Identificar diferencias de entendimiento entre miembros del equipo.
✔️ Exponer riesgos, dependencias y dudas ocultas.
✔️ Lograr consenso basado en conversación, no en votación.





2️⃣0️⃣ Práctica: Construir la Definition of Done

- ✔ Crear una Definition of Done clara y realista.
- ✔ Alinear expectativas entre PO, SM y Developers.
- ✔ Evitar ambigüedad sobre qué significa “terminado”.
- ✔ Identificar criterios de calidad, consistencia y validación.

2️⃣1️⃣ Conclusiones del Módulo 5



✔️ Los tres artefactos:

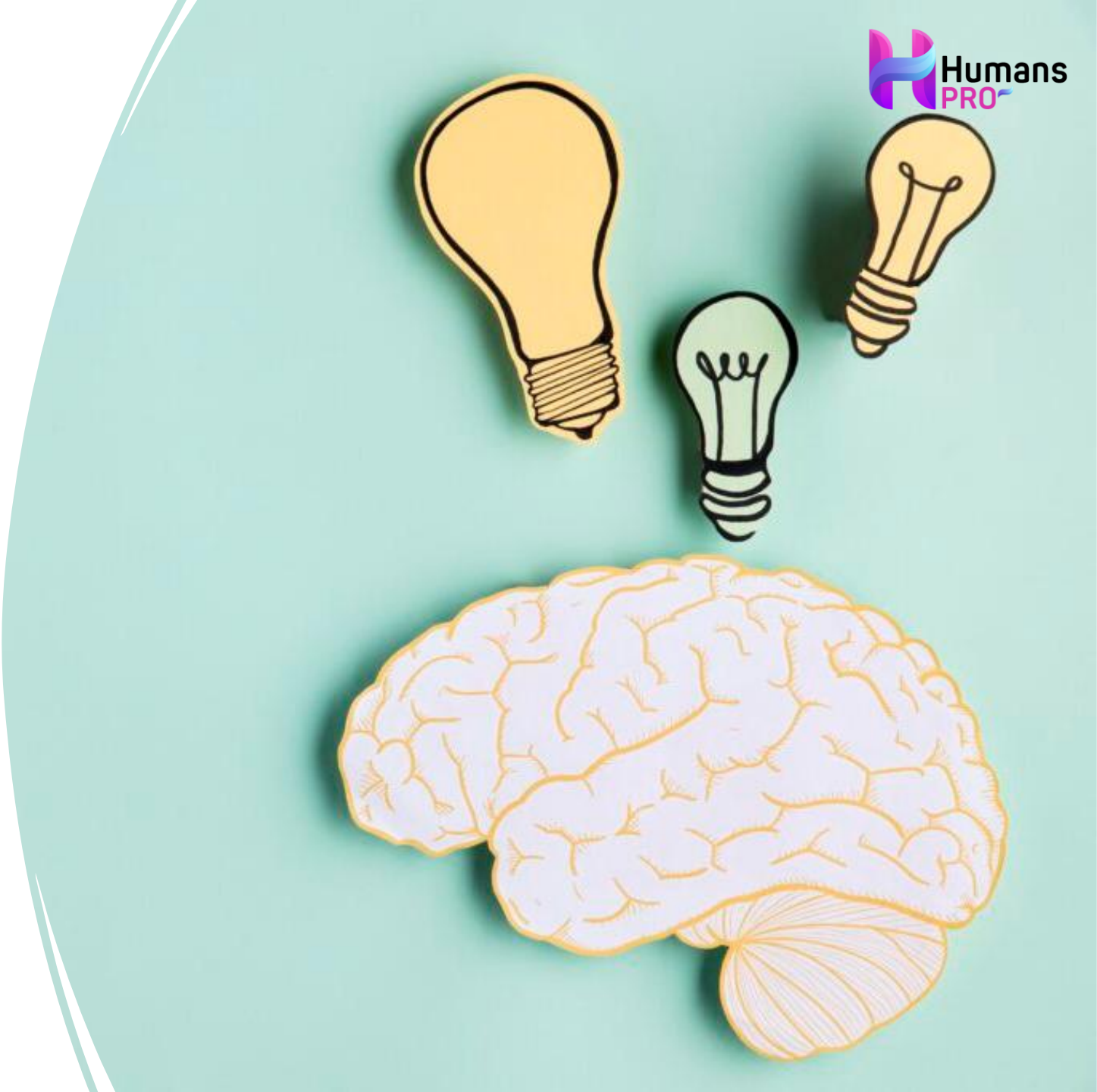
- **Product Backlog** → claridad estratégica
- **Sprint Backlog** → plan táctico
- **Increment** → valor verdadero

✔️ Los tres compromisos:

- **Product Goal**
- **Sprint Goal**
- **Definition of Done**
- ✔️ Herramientas para refinar, estimar y priorizar mejor.
  ✔️ Métricas que permiten inspección y adaptación reales.
  ✔️ Prácticas para escribir, estimar y completar historias de calidad.

2️⃣2️⃣ Reflexión final: ¿Qué cambiarás en tu backlog mañana?

✔️ ¿Qué aprendiste hoy sobre artefactos y compromisos?
✔️ ¿Qué práctica concreta aplicarás en tu equipo?
✔️ ¿Qué mejorarás en tu Product Backlog o Sprint Backlog?
✔️ ¿Qué eliminarás, simplificarás o clarificarás?



Módulo 6. Scrum Mastery y Facilitación Ágil



Scrum Master Profesional

Contenido del Curso (1/3)

- **Módulo 1 - Fundamentos de la Agilidad y de Scrum**
  - El pensamiento ágil y el Manifiesto Ágil.
  - Contexto: complejidad, adaptabilidad y empirismo.
  - Diferencias entre la gestión tradicional y la ágil.
  - ¿Qué es Scrum y qué no es Scrum?
  - Los tres pilares de Scrum: transparencia, inspección y adaptación.
- **Módulo 2 – Valores y Cultura Scrum**
  - Compromiso, Foco, Apertura, Respeto y Coraje.
  - Cómo los valores sustentan la autogestión y la transparencia.
  - Conductas esperadas del Scrum Master para fomentar la cultura Scrum.



✨ ***El conocimiento cobra sentido cuando se convierte en acción con propósito.***

Scrum Master Profesional

Contenido del Curso (2/3)

- **Módulo 3. El Scrum Team y el Rol del Scrum Master**
  - Estructura y responsabilidades del Scrum Team.
  - Qué hace y qué no hace un Scrum Master.
  - El liderazgo servicial: coaching, facilitación y mentoring.
  - Cómo eliminar impedimentos y promover la autogestión.
  - Ejemplos de disfunciones comunes (ScrumBut) y cómo corregirlas.
- **Módulo 4. Los Eventos de Scrum**
  - El Sprint como contenedor de todo.
  - Sprint Planning: propósito, enfoque en el “por qué”, “qué” y “cómo”.
  - Daily Scrum: autogestión, foco y visibilidad del progreso.
  - Sprint Review: transparencia y colaboración con stakeholders.
  - Sprint Retrospective: mejora continua y experimentación.
  - Time-boxing y cómo mantener eventos productivos.

✨ ***El conocimiento cobra sentido cuando se convierte en acción con propósito.***



Scrum Master Profesional

Contenido del Curso (3/3)

- **Módulo 5. Artefactos y Compromisos**
  - Product Backlog, Sprint Backlog e Increment.
  - Compromisos: Objetivo del Producto, Objetivo del Sprint y Definición de Terminado.
  - Refinamiento del backlog.
  - Técnicas ágiles de estimación y priorización (INVEST, Planning Poker, SMART).
  - Control empírico del progreso (Burn-down, Burn-up).
- **Módulo 6. Scrum Mastery y Facilitación Ágil**
  - Habilidades esenciales: escucha activa, coaching, mediación, gestión de conflictos.
  - Cómo crear un entorno de seguridad psicológica.
  - Métricas ágiles: flujo, valor, felicidad del equipo.
  - Introducción a escalamiento y madurez organizacional



✨ ***El conocimiento cobra sentido cuando se convierte en acción con propósito.***



Recordando el Módulo 5

✔ Los tres artefactos:

- **Product Backlog** → claridad estratégica
- **Sprint Backlog** → plan táctico
- **Increment** → valor verdadero

✔ Los tres compromisos:

- **Product Goal**
- **Sprint Goal**
- **Definition of Done**
- ✔ Herramientas para refinar, estimar y priorizar mejor.
  ✔ Métricas que permiten inspección y adaptación reales.
  ✔ Prácticas para escribir, estimar y completar historias de calidad.

Scrum Master Profesional

Contenido del Curso (3/4)

- **Módulo 5. Artefactos y Compromisos**
  - Product Backlog, Sprint Backlog e Increment.
  - Compromisos: Objetivo del Producto, Objetivo del Sprint y Definición de Terminado.
  - Refinamiento del backlog.
  - Técnicas ágiles de estimación y priorización (INVEST, Planning Poker, SMART).
  - Control empírico del progreso (Burn-down, Burn-up).
- **Módulo 6. Scrum Mastery y Facilitación Ágil**
  - Habilidades esenciales: escucha activa, coaching, mediación, gestión de conflictos.
  - Cómo crear un entorno de seguridad psicológica.
  - Métricas ágiles: flujo, valor, felicidad del equipo.
  - Introducción a escalamiento y madurez organizacional



✨ ***El conocimiento cobra sentido cuando se convierte en acción con propósito.***



El rol del Scrum Master más allá de Scrum

- **¿Qué significa “más allá de Scrum”?**

El Scrum Master no es solo facilitador de eventos → es un **agente de cambio**, **coach**, **mentor**, **mediador**, **protector del equipo**, **educador ágil** y **catalizador de mejora continua**.



El rol del Scrum Master más allá de Scrum

1 Coaching del equipo
Facilita reflexión, aprendizaje y crecimiento.
Hace preguntas poderosas en vez de dar órdenes.
Impulsa la autogestión, evitando ser “Scrum Police”.

2 Influencia sin autoridad (liderar sin jerarquía)
Influye por credibilidad, coherencia y confianza.
Navega relaciones difíciles con stakeholders, management y PO.
Facilita acuerdos sin imponer.



El rol del Scrum Master más allá de Scrum

3 Protección del equipo
Bloquea interrupciones externas.
Defiende el foco durante el Sprint.
Escala problemas de manera estratégica.

4 Catalizador de cultura ágil
Promueve transparencia, empirismo y colaboración.
Enseña la agilidad más allá del framework.
Conecta áreas para reducir fricciones.



El rol del Scrum Master más allá de Scrum

5 Identificación de disfunciones sistémicas
El Scrum Master mira más allá del equipo.
Señala dependencias tóxicas.
Identifica cuellos de botella operativos.
Facilita conversaciones difíciles.

6 Impulsor de autogestión y madurez
Ayuda al equipo a pasar de “necesitamos instrucciones” a “nos organizamos”.
Reduce dependencias del Scrum Master con el tiempo.

Un Scrum Master observa que durante el Sprint, el Product Owner insiste en cambiar prioridades y presiona al equipo para incluir trabajo no planificado, generando tensión y ansiedad.
El Scrum Master quiere influir en el PO sin usar autoridad ni confrontación directa.

¿Cuál es la mejor acción para un Scrum Master “más allá de Scrum”?
A. Decirle al PO que “Scrum no permite cambios durante el Sprint” y solicitar que se respete la regla, reforzando la guía del framework.
B. Reunirse con el PO, mostrarle el impacto en el flujo (cycle time y WIP), y facilitar una conversación basada en datos sobre alternativas de menor impacto.
C. Pedir al equipo que rechace cualquier cambio proveniente del PO durante el Sprint, para enseñar por experiencia que debe respetarse el proceso.
D. Escalar inmediatamente el comportamiento del PO a la gerencia, ya que está comprometiendo la estabilidad del Sprint y la salud del equipo.



En una organización con múltiples dependencias, un equipo Scrum está constantemente bloqueado por otro equipo que trabaja en cascada. El PO considera que el problema está en “los developers que no se organizan bien”, mientras que los developers creen que “el PO prioriza mal”.
El Scrum Master quiere atacar la raíz del problema desde una perspectiva de **Scrum Master más allá de Scrum**.
¿Cuál es la acción más adecuada?
A. Facilitar una Retrospectiva enfocada en mejorar la comunicación interna del equipo, sin incluir a otros equipos para mantener la sesión enfocada.
B. Trabajar con el PO para redefinir prioridades y reorganizar el Sprint Backlog según dependencias externas, aceptando que “así funciona la empresa”.
C. Facilitar una conversación conjunta entre ambos equipos usando un mapa de flujo (Value Stream Mapping) para visualizar dependencias y descubrir impedimentos sistémicos.
D. Solicitar a los managers que reasignen personal para que el equipo afectado tenga más capacidad y deje de depender de otros.



Habilidades Esenciales del Scrum Master (Soft Skills Avanzadas)



- 👂 Escucha activa
- ❓ Preguntas poderosas (Coaching Ágil)
- 🕊️ Mediación y gestión de conflictos
- 🤝 Influencia sin autoridad
- 🧭 Liderazgo servicial
- 💬 Comunicación clara y empática
- ⚖️ Inteligencia emocional

Habilidades Esenciales del Scrum Master (Soft Skills Avanzadas)



1️⃣ Escucha Activa (👂)

- La habilidad más subestimada del rol.
- **Principios:**
- Escuchar para entender, no para responder.
- Parafrasear para validar.
- Observar lenguaje corporal.
- Haer silencio intencional.
-

2️⃣ Coaching Ágil: Preguntas Poderosas (❓)

- Guía al equipo sin dar respuestas.
- **Principios:**
- No resolver por ellos.
- Hacer preguntas que inviten a pensar.
- Crear conciencia → generar acción.

Habilidades Esenciales del Scrum Master (Soft Skills Avanzadas)



3️⃣ **Mediación y Gestión de Conflictos (🕊️)**
Un equipo sin conflicto es un equipo sin confianza... o sin sinceridad.
Principios:
Intervenir sin tomar partido.
Facilitar acuerdos sin imponer soluciones.
Enfocar en el problema, no en las personas.
Marco práctico: LASER
Listen — Escuchar
Affirm — Validar
Specify — Desescalar
Explore — Explorar opciones
Resolve — Cerrar acuerdo

4️⃣ **Influencia sin Autoridad (🤝)**
El Scrum Master lidera con credibilidad, no con poder.
Principios:
Construir relaciones antes de usarlas.
Ganar confianza con consistencia.
Alinear expectativas entre áreas sin jerarquía.
Usar datos del flujo como herramienta neutral.

Habilidades Esenciales del Scrum Master (Soft Skills Avanzadas)



5️⃣ Liderazgo Servicial (🧭)
No dirige al equipo → los acompaña a ser mejores.
Principios:
Humildad + valentía.
Servir sin convertirse en asistente del equipo.
Facilitar sin protagonismo.
Enseñar sin humillar.

6️⃣ Comunicación Clara y Empática (💬)
Un Scrum Master es un traductor entre negocio, tecnología y equipo.
Principios:
Mensajes cortos y precisos.
Hablar con intención.
Ajustar la comunicación a la audiencia (técnica vs negocio).
Empatía → escuchar emociones antes de datos.

7️⃣ Inteligencia Emocional (⚖️)
Leer tensiones, gestionar presión, desactivar reacciones impulsivas.
Componentes:
Autoconciencia
Autogestión
Empatía
Habilidad social

Durante una discusión técnica, los developers están bloqueados porque no logran decidir cómo dividir una historia compleja.
Después de 15 minutos de debate, todos se quedan en silencio esperando que el Scrum Master “les diga cómo hacerlo”.

Como Scrum Master con fuertes soft skills, ¿cuál es la mejor intervención?

A. Proponer directamente la mejor opción técnica basada en experiencias previas, para destrabar la conversación y avanzar.
B. Preguntar: “¿Qué información nos falta para decidir?” y facilitar que el equipo genere sus propias opciones y criterios de división.
C. Indicar al Tech Lead que tome la decisión rápidamente para evitar perder más tiempo.
D. Cambiar la conversación hacia otro tema y dejar que el equipo resuelva después en privado para no presionarlos.

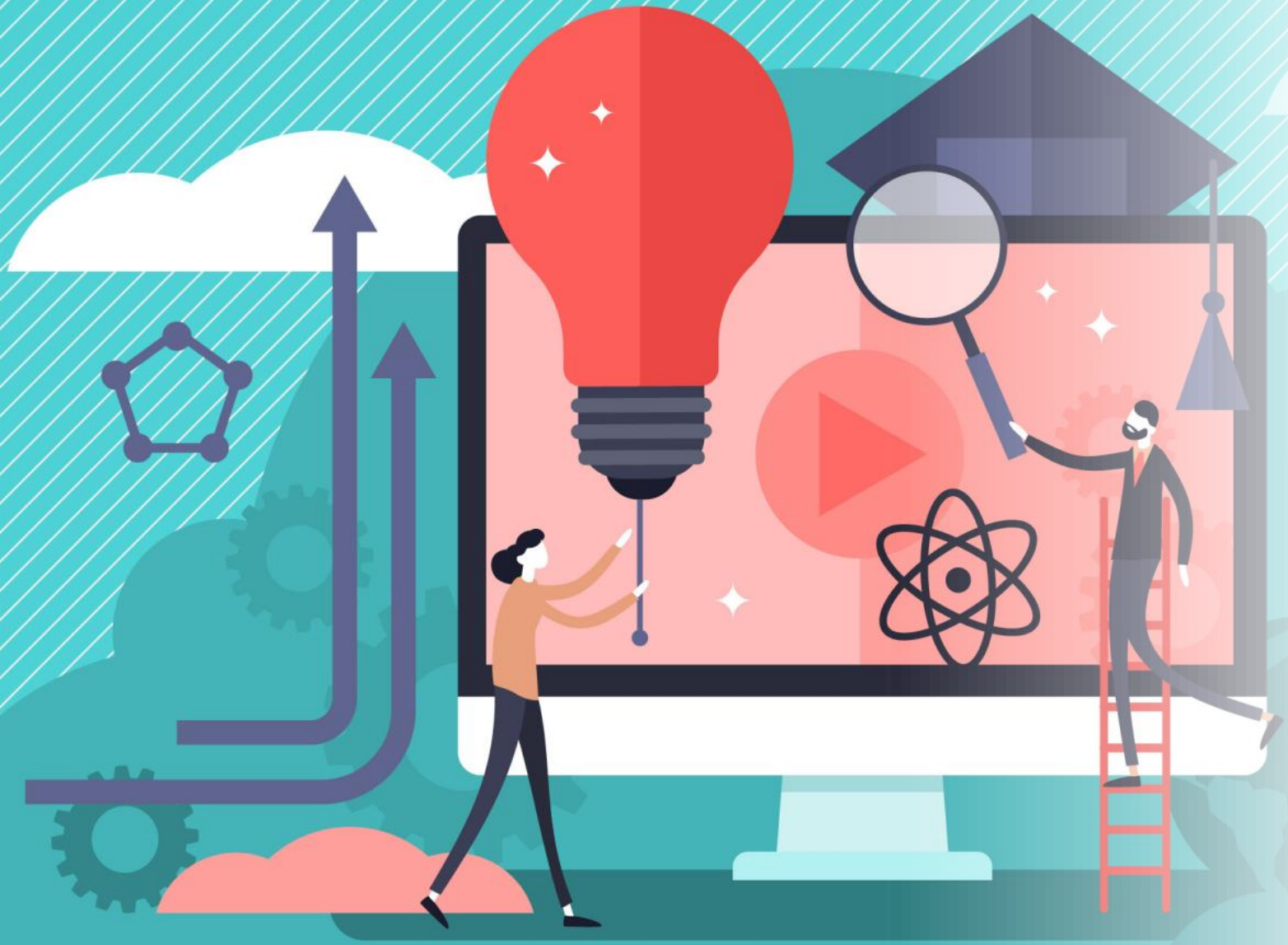


En medio de un Sprint, QA y un developer entran en un conflicto: QA acusa a Dev de “entregar código sin validar”, mientras Dev acusa a QA de “demorar todo y ser demasiado exigente”.
La tensión sube y ambos miran al Scrum Master para que “decida quién tiene razón”.

¿Cuál es la mejor respuesta desde las soft skills avanzadas?

A. Defender a QA, porque la calidad es responsabilidad de todo el equipo y el developer debe ajustarse a los estándares.
B. Defender al developer, porque QA también tiene responsabilidad en el flujo y no debe convertirse en cuello de botella.
C. Aplicar la técnica LASER: escuchar a ambos, validar emociones, desescalar, explorar causas reales y guiar hacia un acuerdo concreto.
D. Proponer dividir las historias para reducir puntos de fricción entre Dev y QA, evitando el conflicto por completo.

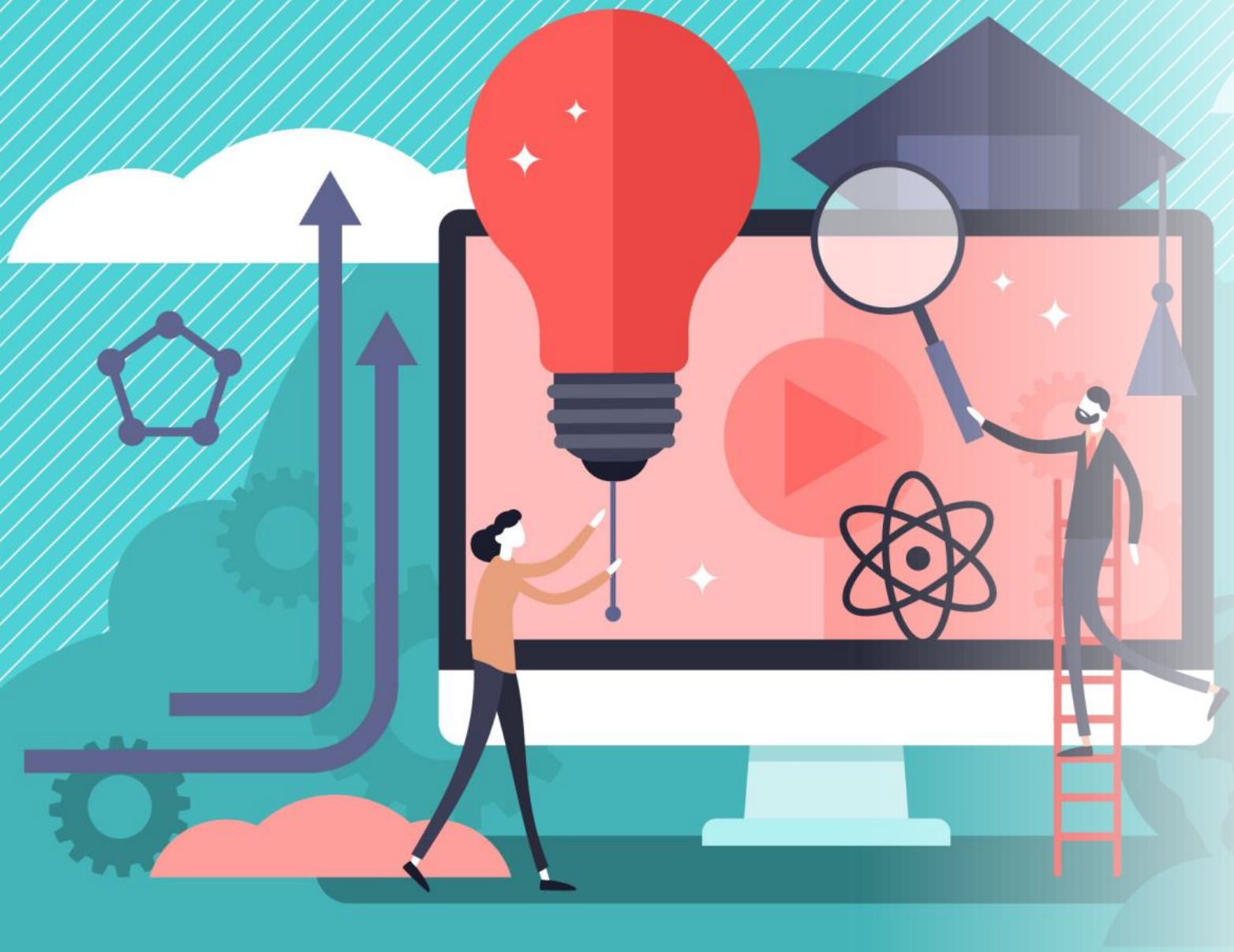




Facilitación Ágil — Técnicas y Herramientas

El Scrum Master diseña, guía y sostiene conversaciones productivas para que el equipo:

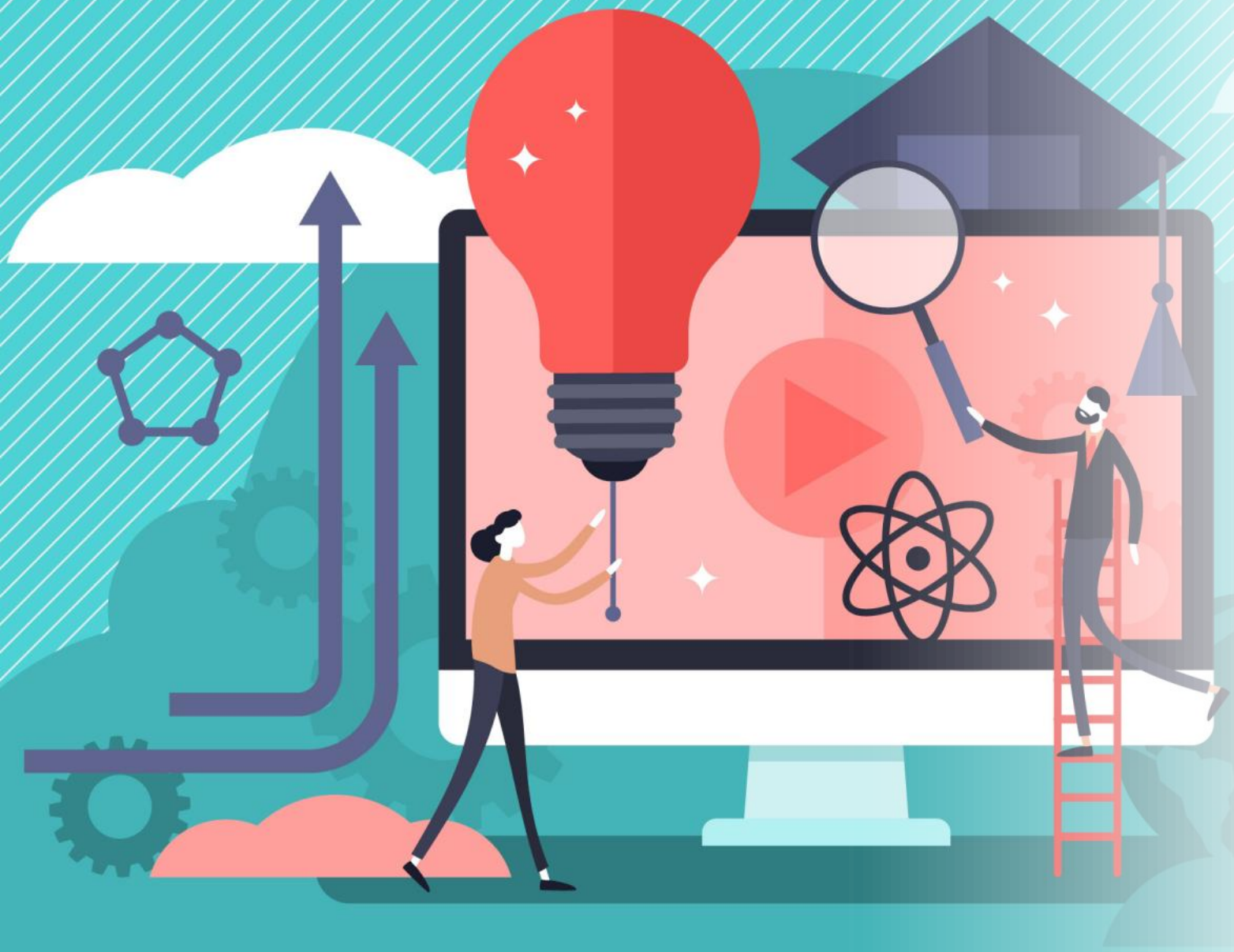
- Colabore major
- Tome decisions
- Resuelva problemas
- Aprenda y mejore continuamente



Facilitación Ágil — Técnicas y Herramientas

Competencias clave de facilitación:

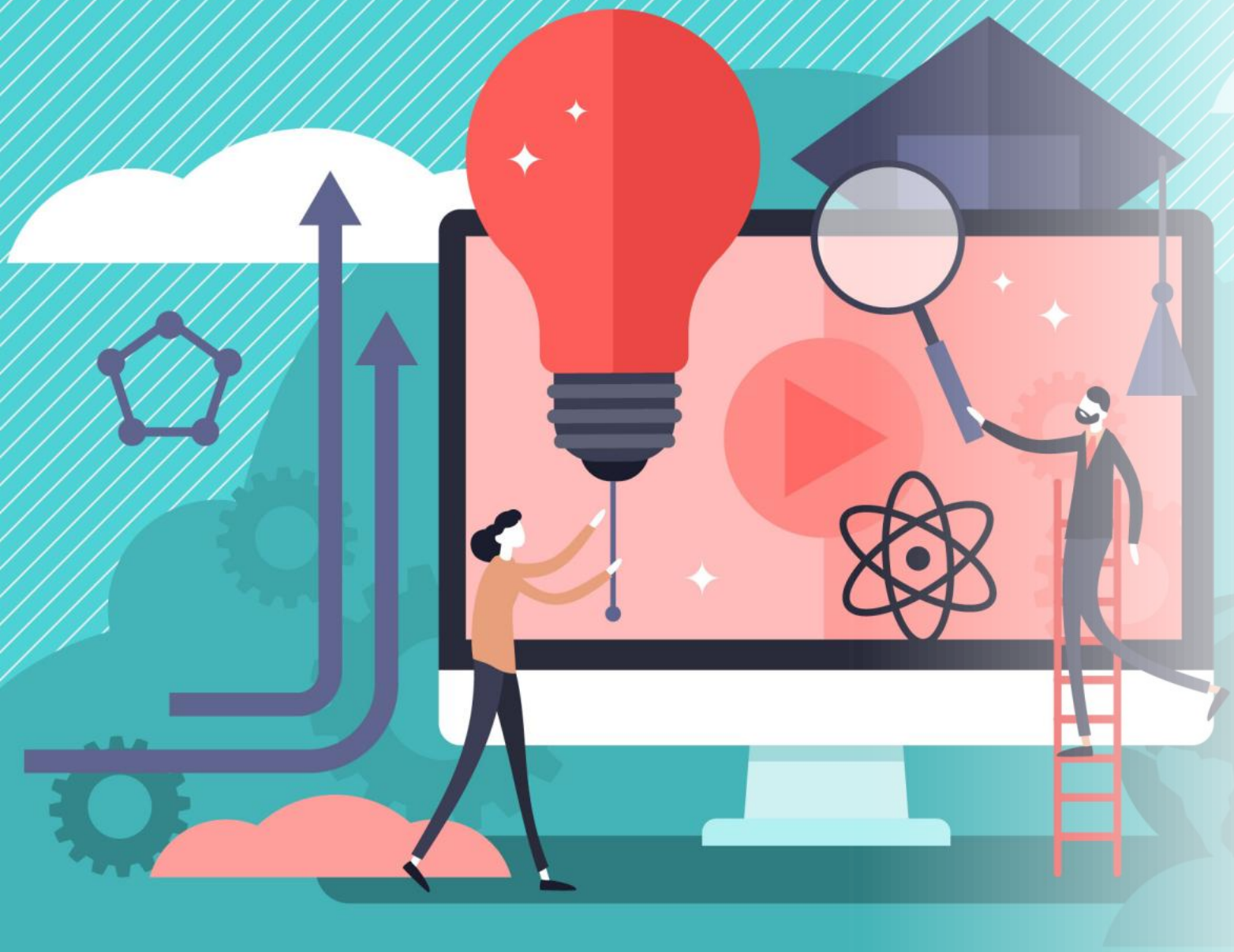
- 🎯 Claridad de propósito
- 🗺️ Diseño de reuniones
- ⏱️ Time-boxing efectivo
- 🧘 Neutralidad del facilitador
- 🗣️ Técnicas de participación
- 🌀 Gestión de la energía del grupo
- 🪞 Visualización del pensamiento



Facilitación Ágil — Técnicas y Herramientas

1️⃣ Claridad de Propósito (🎯)
Cada reunión debe iniciar respondiendo:
¿Para qué estamos aquí?
No “qué haremos”, sino *para qué*.
Buenas prácticas:
Propósito en 1 frase.
Mensaje visible al inicio del evento.
Alinear expectativas antes de comenzar.

2️⃣ Diseño de Reuniones (🗺️)
Una buena reunión tiene:
Propósito
Agenda
Resultado esperado
Dinámica clara
Cierre explícito
Estructura recomendada:
Opening → encuadre
Exploration → ideas, datos, debate
Narrowing → priorizar, decidir
Closing → acciones, responsables, próximos pasos
Marco útil: ORID
• Objective → Datos
• Reflective → Emociones
• Interpretive → Significado
• Decisional → Acción

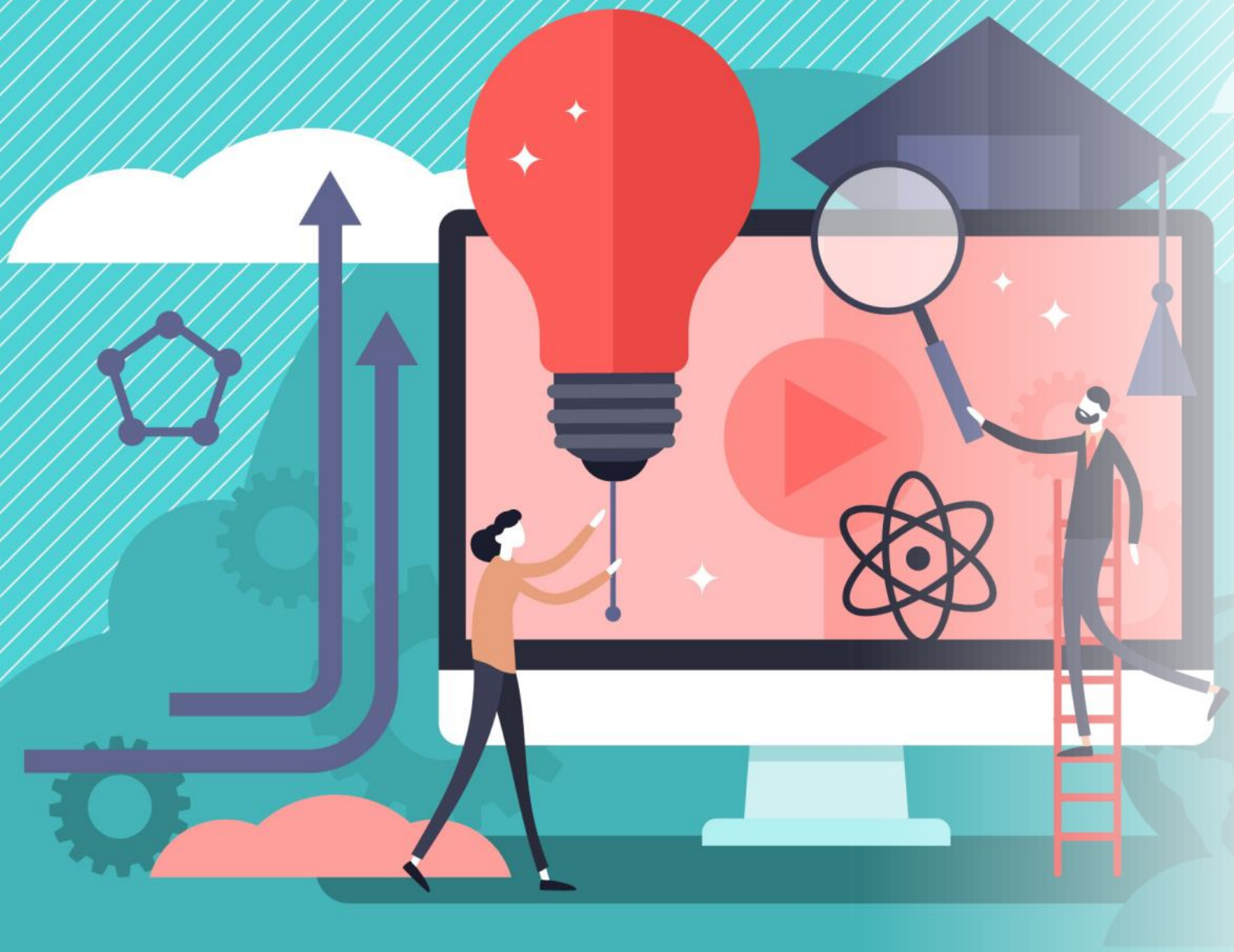


Facilitación Ágil — Técnicas y Herramientas

3️⃣ **Neutralidad del Facilitador (🧘)**
El Scrum Master guía el proceso… **sin opinar del contenido**.
Principios:
No tomar partido
No dirigir decisiones
Evitar expresiones de sesgo (“yo creo que deberíamos…”)

4️⃣ **Time-Boxing Inteligente (⏱️)**
El tiempo protege el foco del equipo.
Claves:
Explicar el tiempo antes de comenzar
Avisos intermedios (“quedan 3 minutos”)
Cerrar cuando el timebox termine, no cuando ‘alguien’ termine

5️⃣ **Técnicas de Participación (🗣️)**
Garantizan que todos hablan, no solo los de siempre.
Técnicas recomendadas:
1–2–4–All (Liberating Structures)
Brainwriting (escribir ideas antes de hablar)
Round Robin (turnos equitativos)
Dot Voting (priorizar rápido)
Silent Sorting (ordenar ideas sin hablar)



Facilitación Ágil — Técnicas y Herramientas

6️⃣ Visualización del Pensamiento (🔮)
Las ideas se vuelven más claras cuando se ven.
Herramientas:
Tableros tipo Miro/Mural
Post-its físicos
Mapas mentales
Matrices (Esfuerzo vs Valor, Eisenhower, etc.)
Diagramas de flujo rápidos
Regla:
Si es importante → debe estar visible.

7️⃣ Gestión de la Energía del Grupo (🌀)
La energía define la productividad.
Técnicas:
Icebreakers cortos
Check-ins emocionales
Pausas activas
Cambios de dinámica cuando cae la atención
Actividades en silencio para descansar la mente



Facilitación Ágil — Técnicas y Herramientas

Técnicas clave para facilitar reflexión profunda y aprendizaje continuo:

⭐ **Starfish**

Estructura de 5 ejes:

Start – More – Less – Continue – Stop

→ Ideal para identificar hábitos y ajustar prácticas.

⭐ **4Ls (Liked, Learned, Lacked, Longed For)**

→ Útil para capturar aprendizajes, necesidades y expectativas del Sprint.

⭐ **Mad – Sad – Glad**

→ Basada en emociones: ayuda a destapar tensiones y celebrar logros.

⭐ **Retrospective Circles**

→ Formato circular para explorar problemas complejos o sensibles en capas.

⭐ **Root Cause Analysis**

→ Para ir más allá del síntoma:

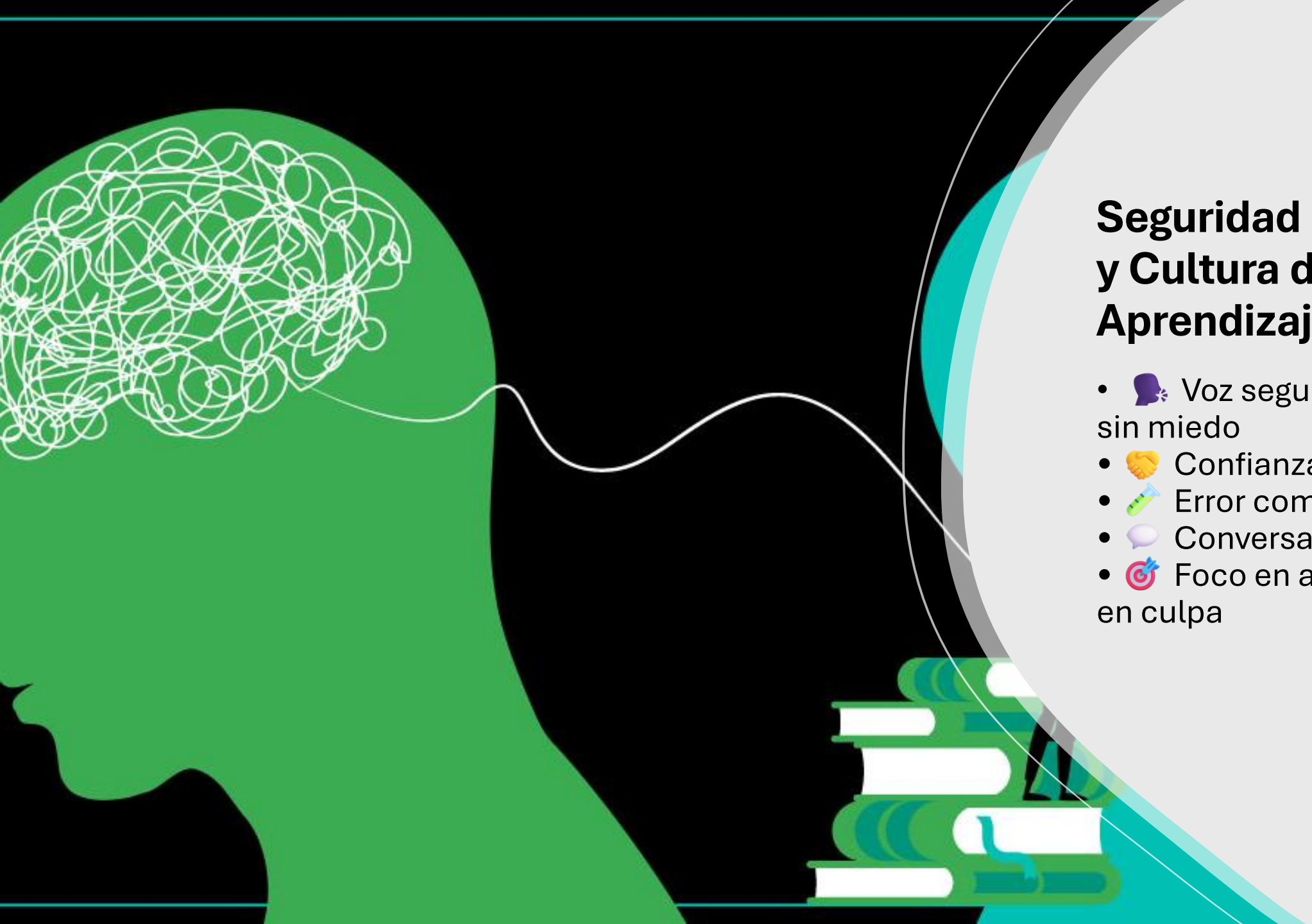
- **5 Whys** → indagar hasta la causa raíz
- **Fishbone (Ishikawa)** → visualizar causas por categorías

Durante una sesión de Sprint Planning, el equipo debate entre dos enfoques para implementar una historia.
La discusión se vuelve circular y el Scrum Master identifica que el tiempo del evento se está agotando.
Uno de los developers le pregunta directamente:
“¿Qué opinan? ¿Cuál opción es la mejor?”
Como facilitador ágil, ¿cuál es la mejor respuesta?
A. Elegir la opción más eficiente técnicamente para evitar que el equipo se quede sin decisión dentro del timebox.
B. Responder que ambas opciones pueden funcionar y sugerir analizar pro y contras, dejando que el equipo decida libremente.
C. Mantener la neutralidad y facilitar un método de toma de decisiones rápido (p. ej. matriz de decisión, dot voting guiado, o “decidir por criterios”), sin compartir opinión personal.
D. Pedir una pausa y consultar con un arquitecto o experto externo para traer una decisión “objetiva” y evitar responsabilidad sobre la decisión.



En una Retrospectiva con 9 personas, el Scrum Master nota que solo 3 participantes aportan ideas, mientras que el resto permanece silencioso.
Ya se ha intentado un brainstorming tradicional y no funcionó.
El SM quiere aumentar la participación sin exponer a quienes se sienten incómodos hablando.
¿Cuál técnica de facilitación avanzada es más adecuada?
A. Round Robin, pidiendo que cada persona hable uno por uno para garantizar igualdad de participación.
B. Silent Brainwriting, permitiendo que cada persona genere ideas en silencio antes de compartirlas de manera estructurada.
C. 1-2-4-All, porque obliga a las personas a hablar en parejas y luego en grupos más grandes, incrementando su exposición progresiva.
D. Hacer una lluvia de ideas abierta nuevamente, pero asignando un “moderador” dentro del equipo para incentivar mejores aportes.





Seguridad Psicológica y Cultura de Aprendizaje

- 🗣️ Voz segura: puedo hablar sin miedo
- 🤝 Confianza interpersonal
- 🧪 Error como oportunidad
- 💬 Conversaciones abiertas
- 🎯 Foco en aprendizaje, no en culpa



Seguridad Psicológica y Cultura de Aprendizaje

1️⃣ ¿Qué es realmente la Seguridad Psicológica? (🗣️)
Más que un concepto, es un clima.
Significa que el equipo siente que puede:

- Hacer preguntas sin parecer incompetente.
- Cometer errores sin ser criticado.
- Proponer ideas sin ser ridiculizado.
- Dar feedback sin temor a represalias.
- Discutir temas sensibles sin tensión.



Seguridad Psicológica y Cultura de Aprendizaje

2️⃣ Señales de Alta y Baja Seguridad Psicológica (🔍)

Alta seguridad psicológica (🟢):

- Se pide ayuda fácilmente.
- Todos participan en Daily y Retro.
- Se dicen los problemas temprano.
- Hay desacuerdos sanos.
- El equipo admite errores rápidamente.



Seguridad Psicológica y Cultura de Aprendizaje

3️⃣ Cómo construye seguridad psicológica un Scrum Master (🧩)
El Scrum Master modela comportamientos, no los pide.
Acciones concretas:

- Normalizar el error: “¿Qué aprendimos?”
- Mostrar vulnerabilidad: “No entendí eso, ¿puedes explicarlo?”
- Facilitar turnos equitativos en reuniones.
- Reflejar emociones: “Suena a que esto generó frustración.”
- Detener comportamientos tóxicos.
- Proteger de interrupciones y presiones externas.
- Promover retrospectivas profundas (no superficiales).
- Usar el lenguaje de aprendizaje: “¿Qué hipótesis probamos esta semana?”



Seguridad Psicológica y Cultura de Aprendizaje

5️⃣ Herramientas prácticas para construir cultura (🔧)

a) Check-in emocional (2 minutos)
"Usen una palabra para describir cómo llegan hoy."

b) Technique: "Fail but Learn"
Cada Sprint → un error, una enseñanza, una acción.

c) Working Agreements
Normas co-creadas por el equipo.

d) Rondas equitativas
Todos hablan antes de discutir.

e) Preguntas del Scrum Master



Seguridad Psicológica y Cultura de Aprendizaje

6️⃣ **Cómo manejar conversaciones difíciles (🔥)**
El Scrum Master no evita conversaciones incómodas: las hace seguras.
Marco recomendado:
3C – Claridad, Contexto, Cuidado
Claridad → “Observé que…”
Contexto → “El impacto fue…”
Cuidado → “Quiero que esto funcione para todos.”

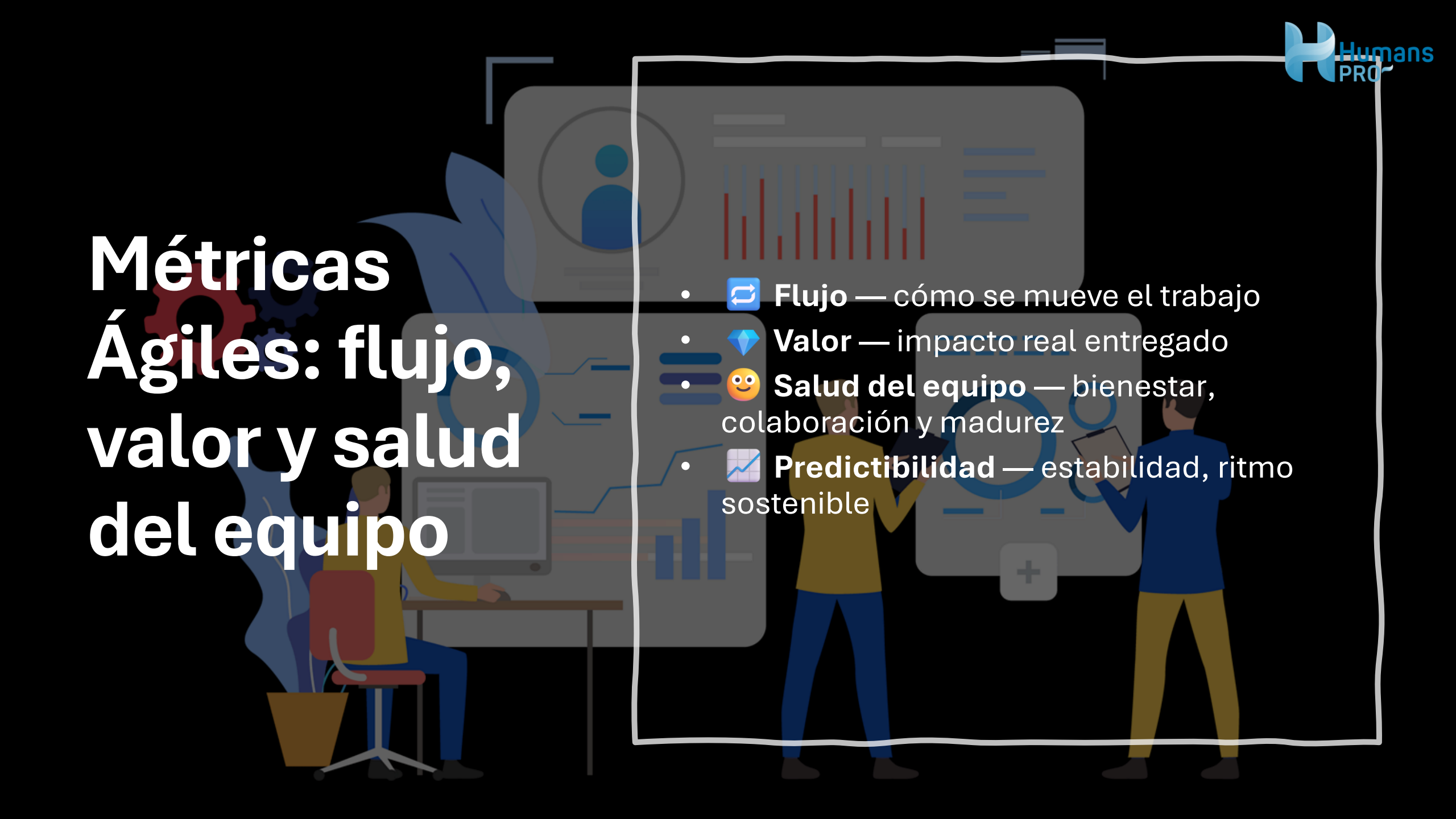
Durante una Retrospectiva, el equipo se muestra silencioso y evasivo.
El Scrum Master nota gestos de incomodidad cuando se mencionan errores del Sprint.
Finalmente, un developer dice en voz baja:
“La verdad, prefiero no hablar porque luego esto me lo sacan en cara en la semana.”
¿Cuál es la mejor intervención del Scrum Master para restaurar seguridad psicológica?
A. Decirle al equipo que Scrum requiere hablar abiertamente y que todos deben aportar, reforzando que es parte del proceso.
B. Preguntar quién “saca en cara” los errores para manejar el conflicto directamente y dejarlo claro en la reunión.
C. Validar la emoción, reforzar la no-culpa y guiar al equipo hacia acuerdos explícitos de seguridad mediante normas de trabajo (Working Agreements).
D. Cambiar el enfoque de la Retro hacia un análisis de datos objetivos, para evitar emociones que podrían bloquear la conversación.



El equipo ha cometido un error que causó retrasos en producción.
En la reunión post-mortem, el manager dice:
“Necesito saber quién fue el responsable para evitar que esto pase otra vez.”
El Scrum Master quiere proteger la cultura de aprendizaje sin confrontar directamente al manager.
¿Cuál es la intervención más efectiva?
A. Explicar al manager que en Scrum no se buscan culpables y que ese tipo de preguntas dañan la psicología del equipo.
B. Responder: “Todos fuimos responsables”, para evitar que se señale a alguien y mantener la cohesión del equipo.
C. Redirigir la conversación hacia el sistema y el flujo, mostrando la cadena completa de eventos y facilitando análisis causal sin personalizar.
D. Solicitar que el manager abandone la reunión para proteger emocionalmente al equipo y seguir con el análisis de forma segura.



Métricas Ágiles: flujo, valor y salud del equipo



- 🔁 **Flujo** — cómo se mueve el trabajo
- 💎 **Valor** — impacto real entregado
- 🫠 **Salud del equipo** — bienestar, colaboración y madurez
- 📈 **Predictibilidad** — estabilidad, ritmo sostenible

Métricas Ágiles: flujo, valor y salud del equipo



1 Métricas de Flujo (🔁)
Miden cómo se mueve el trabajo desde "Pendiente" → "Hecho".

a) Lead Time
Tiempo desde que surge una necesidad hasta que se entrega valor.
Pregunta clave: ¿Qué tan rápido entregamos valor?

b) Cycle Time
Tiempo desde que el equipo empieza el trabajo hasta que lo termina.
Pregunta clave: ¿Cuánto nos toma completar una historia?

c) Throughput
Cantidad de ítems completados por Sprint o semana.
Pregunta clave: ¿Cuál es nuestra capacidad real?

d) Cumulative Flow Diagram (CFD)
Visualiza cuellos de botella.

Métricas Ágiles: flujo, valor y salud del equipo



2️⃣ Métricas de Valor (💎)
No miden actividad, miden impacto.
Ejemplos de métricas de valor:

- Valor entregado por release
- Uso de funcionalidades (¿la gente realmente las usa?)
- Métricas del negocio (NPS, conversión, ingresos, retención)
- ROI del incremento
- % de entregables alineados al Objetivo del Producto

3️⃣ Métricas de Salud del Equipo (🥹)
El equipo es un sistema humano: su salud influye directamente en su rendimiento.
Indicadores:

- Team Happiness Index
- Team Health Check (Spotify model adaptado)
- Rotación / desgaste del equipo
- Niveles de colaboración
- Frecuencia de impedimentos
- Seguridad psicológica

Métricas Ágiles: flujo, valor y salud del equipo



4️⃣ **Métricas de Predictibilidad (📈)**
¿Cómo de confiables son nuestras entregas?
Indicadores clave:
- Porcentaje de historias completadas vs. comprometidas
- Estabilidad del throughput en 3–5 Sprints
- Variación del cycle time
- Desviación en la entrega de releases

No debe interpretarse como "cumplir por cumplir":
La predictibilidad ayuda a tomar mejores decisiones, NO a presionar al equipo.

5️⃣ **Lo que NO son métricas ágiles (🛑)**
Estas prácticas generan miedo, manipulación o comportamientos tóxicos:
- 🛑 Velocity comparada entre equipos
- 🛑 Medir performance individual
- 🛑 Contar "líneas de código"
- 🛑 Usar métricas para castigar
- 🛑 Medir tiempo sentado en reuniones
- 🛑 Forzar throughput a subir artificialmente

Regla de oro:
Las métricas nunca deben usarse contra las personas.

Un equipo muestra el siguiente comportamiento en las últimas 4 semanas:
El **Throughput** ha aumentado un 15%.
El **Cycle Time promedio** también ha aumentado un 25%.
El **WIP** ha crecido ligeramente en la columna “In Progress”.
El PO interpreta este comportamiento como “el equipo está entregando más y volviéndose más productivo”.
Como Scrum Master, ¿cómo interpretarías correctamente este patrón?
A. El equipo es más productivo porque el throughput subió; el aumento de cycle time es normal cuando aumenta la producción.
B. El throughput subió artificialmente debido al aumento del WIP; el ciclo de trabajo más largo indica menor fluidez del sistema.
C. La productividad del equipo se mantiene estable; el aumento del cycle time es irrelevante mientras el throughput suba.
D. El equipo está tomando historias más complejas y por eso aumentó el cycle time; el throughput es evidencia de mejor rendimiento.



Un equipo completó 28 historias en el último trimestre, manteniendo una velocidad estable.
Sin embargo, el área de negocio comenta que “el producto no está avanzando como esperábamos”.
El Scrum Master analiza las métricas y descubre:
Las historias entregadas fueron mayoritariamente **técnicas o internas**.
El uso real de las nuevas funcionalidades por parte de usuarios finales es del **3%**.
Las métricas del negocio (NPS, conversión, retención) no mostraron cambios.
El Product Backlog no prioriza impacto, sino “lo que está listo para desarrollar”.
¿Cuál es la interpretación correcta desde una perspectiva profesional?
A. El equipo es altamente productivo, pero el negocio debe ajustar sus expectativas porque velocidad y valor no siempre están correlacionados.
B. El rendimiento del equipo es excelente y la falta de impacto es responsabilidad exclusiva del Product Owner por priorizar mal.
C. El equipo está optimizando por actividad, no por valor; el Scrum Master debe facilitar una conversación profunda sobre priorización basada en impacto.
D. No hay problema real: si la velocidad es estable, el impacto del producto llegará eventualmente.





Introducción al Escalamiento y Madurez Organizacional

- No todas las organizaciones están listas para escalar.

El Scrum Master debe entender **cuándo**, **por qué** y **cómo** escalar sin generar caos.



Introducción al Escalamiento y Madurez Organizacional

📌 ¿Cuándo tiene sentido escalar?
📌 Riesgos del escalamiento prematuro
📌 Principales marcos de escalamiento (solo visión general)
📌 Madurez del equipo + madurez organizacional
📌 Señales de que un equipo está listo para escalar
📌 Cómo influye el Scrum Master en un ecosistema escalado



Introducción al Escalamiento y Madurez Organizacional

1️⃣ ¿Cuándo tiene sentido escalar? (📌)
Escalar solo cuando existe:
Un producto demasiado grande para un solo equipo
Dependencias inevitables entre componentes
Necesidad real de múltiples equipos trabajando hacia un único Product Goal
Capacidad demostrada de un equipo funcionando bien





Introducción al Escalamiento y Madurez Organizacional

Tuckman



Introducción al Escalamiento y Madurez Organizacional

2️⃣ Riesgos del escalamiento prematuro (⚠️)
- Dependencias explosivas
- Más reuniones, menos entrega
- PO sobrecargado
- Arquitectura frágil
- Falta de claridad entre equipos
- Caída de la calidad
- Velocidad ilusoria (“hacemos más, entregamos menos”)
- Más burocracia que agilidad



Introducción al Escalamiento y Madurez Organizacional

3️⃣ **Visión general de frameworks de escalamiento (📚)**
No se profundiza — solo un mapa mental.
Scrum@Scale
Simplicidad, dos ciclos: Scrum Master Cycle y Product Owner Cycle.
Enfoque en des-silenciar y escalamiento orgánico.
LeSS (Large Scale Scrum)
Menos procesos, más puro Scrum.
Equipos multidisciplinarios, foco en aprendizaje y simplicidad.
Nexus
Extiende Scrum con un rol adicional y un evento centralizado.
Gestión explícita de dependencias.
SAFe (Scaled Agile Framework)
Más estructurado.
Orientado a empresas grandes → ARTs, PI Planning, roles extendidos.
Riesgo: burocratizar si no se implementa con sentido ágil.



Introducción al Escalamiento y Madurez Organizacional

4️⃣ **Madurez del Equipo (🚀)**
Un equipo se puede mapear en niveles:
Nivel 1 — Dependiente:
PO dicta todo, Scrum Master dirige, equipo espera instrucciones.
Nivel 2 — Contribuyente:
Hay colaboración, pero dependen del SM y del PO para todo.
Nivel 3 — Autogestionado:
El equipo decide, se organiza y resuelve sin esperar órdenes.
Nivel 4 — Alto rendimiento:
Entrega continua de valor, aprendizaje constante, decisiones rápidas.



Introducción al Escalamiento y Madurez Organizacional

5️⃣ **Madurez Organizacional (🏛️)**
La organización también debe madurar.
Elementos clave:

- Arquitectura modular
- Cultura de transparencia
- Liderazgo que entiende agilidad
- Product Ownership fuerte
- Cultura que tolera experimentación
- Reducción de silos y política
- Mecanismos para alinear a múltiples equipos



Introducción al Escalamiento y Madurez Organizacional

6️⃣ Señales de que una organización está lista para escalar (🟢)
- Un equipo ya funciona con alta calidad
- Backlog claro, priorizado y estable
- PO experimentado y con disponibilidad
- Arquitectura soporta trabajo paralelo
- Hay mecanismos para gestión de dependencias
- Management entiende empirismo
- Flujo de trabajo predecible
- Equipos con seguridad psicológica (del módulo 6.4)

7️⃣ El rol del Scrum Master en entornos escalados (🧭)
El Scrum Master se convierte en:
Facilitador entre equipos
Conector entre Scrum Masters
Gestor de impedimentos sistémicos
Educador en empirismo
Protector del Product Goal
Diseñador de mecanismos de coordinación
Coach para múltiples PO y stakeholders
Guardián de la simplicidad

Una organización quiere pasar de 1 equipo Scrum a 6 equipos trabajando sobre el mismo producto.
El liderazgo dice:
“Con más equipos podremos entregar más rápido.”
Sin embargo, el Scrum Master observa:
La arquitectura es un monolito difícil de dividir.
El PO está sobrecargado y lucha por mantener un backlog claro.
El equipo actual tiene baja predictibilidad y frecuente WIP excesivo.
Existen dependencias con áreas tradicionales (QA, Infra, Seguridad).
¿Cuál es la intervención más adecuada del Scrum Master desde el entendimiento de escalamiento y madurez organizacional?
A. Aceptar el escalamiento y preparar sesiones de PI Planning para coordinar a los 6 equipos y evitar caos.
B. Proponer entrenar a más POs para dividir las responsabilidades y así balancear la carga de trabajo antes de escalar.
C. Mostrar evidencia de que el sistema actual no está listo para escalar (flujo, dependencias, madurez) y facilitar una conversación sobre estabilizar un equipo antes de multiplicarlo.
D. Solicitar contratar un equipo de arquitectura para refactorizar el monolito mientras se hace el escalamiento en paralelo, para no frenar la velocidad del negocio.



En un entorno con 4 equipos Scrum, se observa que:
Los equipos dependen del mismo analista de negocio para aclarar requisitos.
Hay conflictos frecuentes entre Product Owners en la priorización.
Los equipos escalan demasiados impedimentos porque “nadie tiene autoridad para decidir”.
Las retrospectivas identifican problemas, pero no generan cambios fuera de cada equipo.

¿Cuál es la intervención del Scrum Master que demuestra comprensión avanzada del rol en un ecosistema escalado?

A. Facilitar una gran retrospectiva conjunta (Scrum of Scrums Retro) para que los 4 equipos alineen problemas y acuerden soluciones globales.
B. Recomendar al liderazgo crear un rol adicional de “Coordinador Ágil” para tomar decisiones transversales y evitar escalaciones innecesarias.
C. Pedir a cada PO que defina claramente sus límites y su parte del backlog, para evitar conflictos entre ellos.
D. Enfocar la mejora únicamente dentro de cada equipo, ya que cambiar el sistema completo está fuera del control del Scrum Master.





1. Liderar sin autoridad
2. Facilitar, no dirigir
3. Construir seguridad psicológica
4. Medir para aprender, no para controlar
5. Trabajar en el sistema, no solo en el equipo



Cierre del Módulo 6 — Scrum Mastery en la Práctica

- Comprensión profunda del rol más allá del marco.
- Soft skills avanzadas para liderazgo servicial.
- Técnicas de facilitación para conversaciones efectivas.
- Herramientas para construir seguridad psicológica.
- Métricas para aprender y mejorar el flujo.
- Visión sistémica para trabajar en entornos escalados.

Módulo 7. Preparación para la Certificación





Pregunta 1

Un equipo Scrum está teniendo problemas recurrentes para completar el Sprint Backlog. El Product Owner propone aumentar la supervisión diaria para “asegurar que todos entreguen lo que prometen”. El Scrum Master identifica que el verdadero problema es que el equipo no está planificando la capacidad correctamente ni negociando el alcance durante el Sprint.

¿Cuál debe ser la acción MÁS adecuada del Scrum Master?

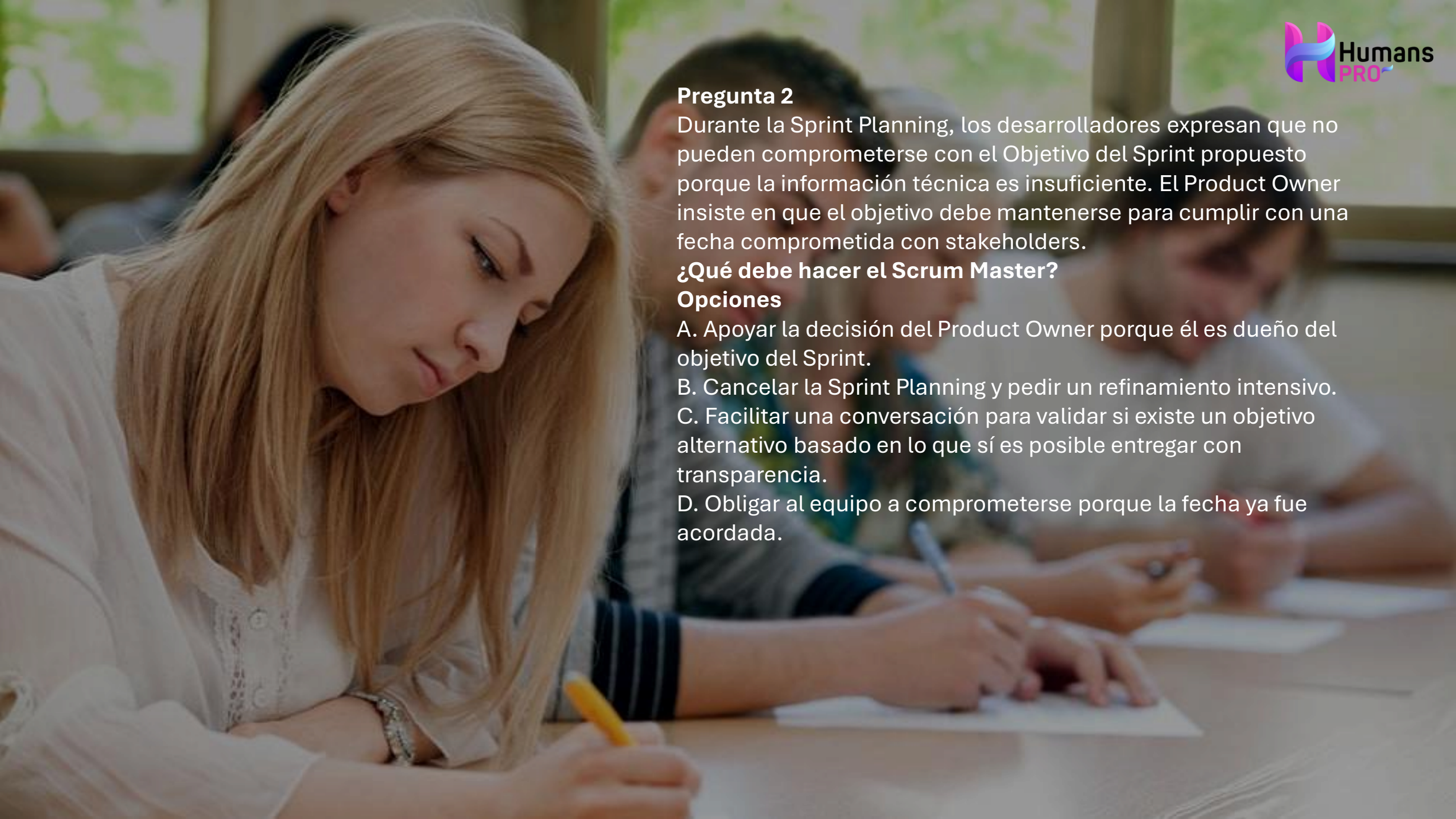
Opciones

A. Aceptar la propuesta del Product Owner para asegurar la entrega del Sprint.

B. Implementar métricas de productividad individuales para entender quién está bajando el rendimiento.

C. Facilitar una sesión de coaching para que el equipo comprenda cómo planificar la capacidad y adaptar el alcance según el progreso.

D. Solicitar a la gerencia que intervenga para corregir la falta de compromiso del equipo.



Pregunta 2

Durante la Sprint Planning, los desarrolladores expresan que no pueden comprometerse con el Objetivo del Sprint propuesto porque la información técnica es insuficiente. El Product Owner insiste en que el objetivo debe mantenerse para cumplir con una fecha comprometida con stakeholders.

¿Qué debe hacer el Scrum Master?

Opciones

A. Apoyar la decisión del Product Owner porque él es dueño del objetivo del Sprint.

B. Cancelar la Sprint Planning y pedir un refinamiento intensivo.

C. Facilitar una conversación para validar si existe un objetivo alternativo basado en lo que sí es posible entregar con transparencia.

D. Obligar al equipo a comprometerse porque la fecha ya fue acordada.



Pregunta 3

En varias retrospectivas, el equipo señala problemas, pero nunca se ejecutan los experimentos acordados. Esto frena la mejora continua y genera frustración.

¿Cuál es la mejor intervención del Scrum Master?

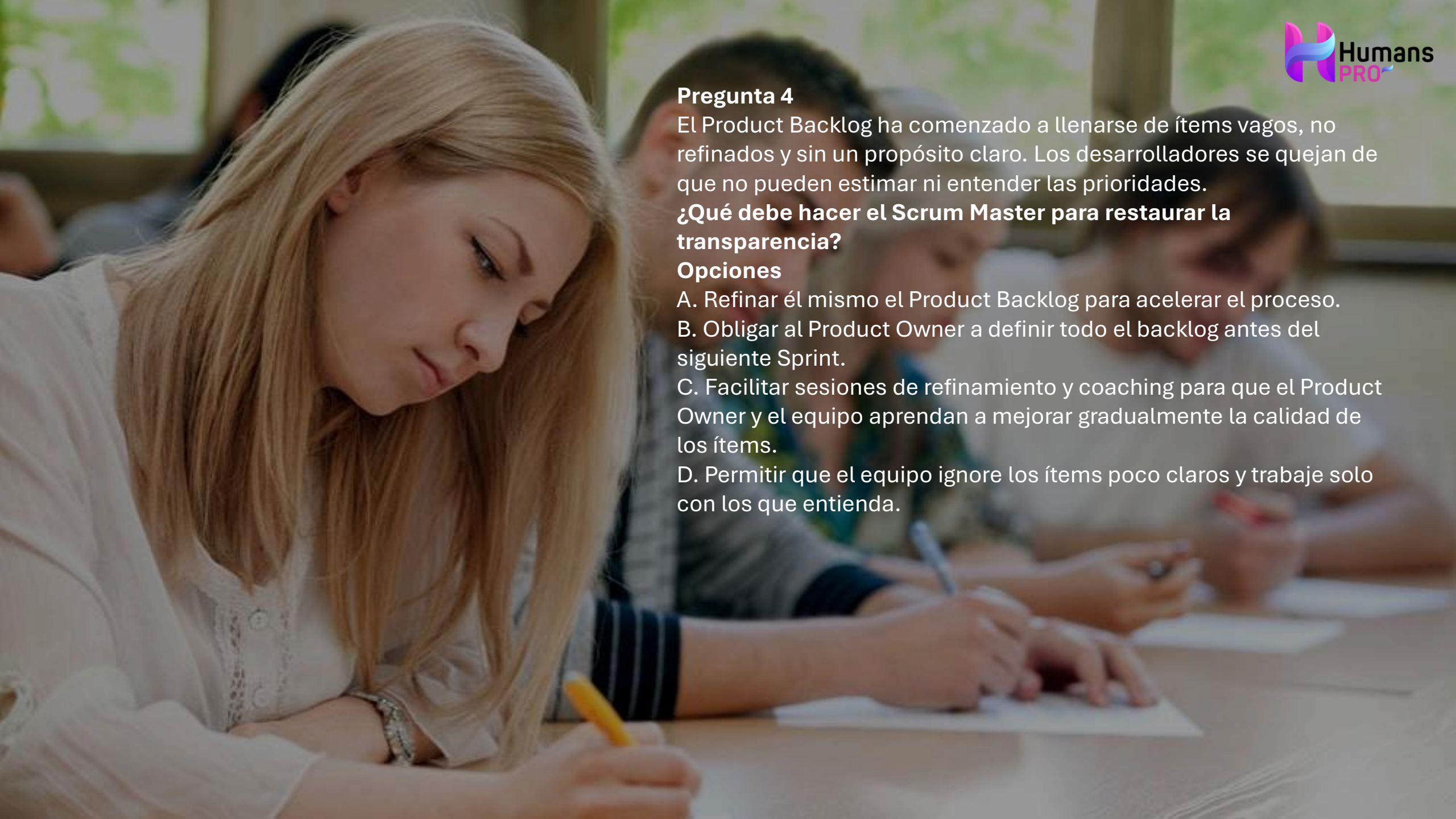
Opciones

A. Hacer un reporte formal a la gerencia para obligar al equipo a cumplir los experimentos.

B. Convertir los experimentos en ítems visibles dentro del Sprint Backlog con un dueño claro.

C. Eliminar la retrospectiva hasta que el equipo esté más maduro.

D. Imponer un formato más estricto para la retrospectiva.



Pregunta 4

El Product Backlog ha comenzado a llenarse de ítems vagos, no refinados y sin un propósito claro. Los desarrolladores se quejan de que no pueden estimar ni entender las prioridades.

¿Qué debe hacer el Scrum Master para restaurar la transparencia?

Opciones

A. Refinar él mismo el Product Backlog para acelerar el proceso.

B. Obligar al Product Owner a definir todo el backlog antes del siguiente Sprint.

C. Facilitar sesiones de refinamiento y coaching para que el Product Owner y el equipo aprendan a mejorar gradualmente la calidad de los ítems.

D. Permitir que el equipo ignore los ítems poco claros y trabaje solo con los que entienda.



Pregunta 5

Un director pide al Scrum Master métricas para identificar qué miembro del equipo es el menos productivo. Esto va en contra de los valores Scrum, pero la solicitud viene de arriba.

¿Qué debe hacer el Scrum Master?

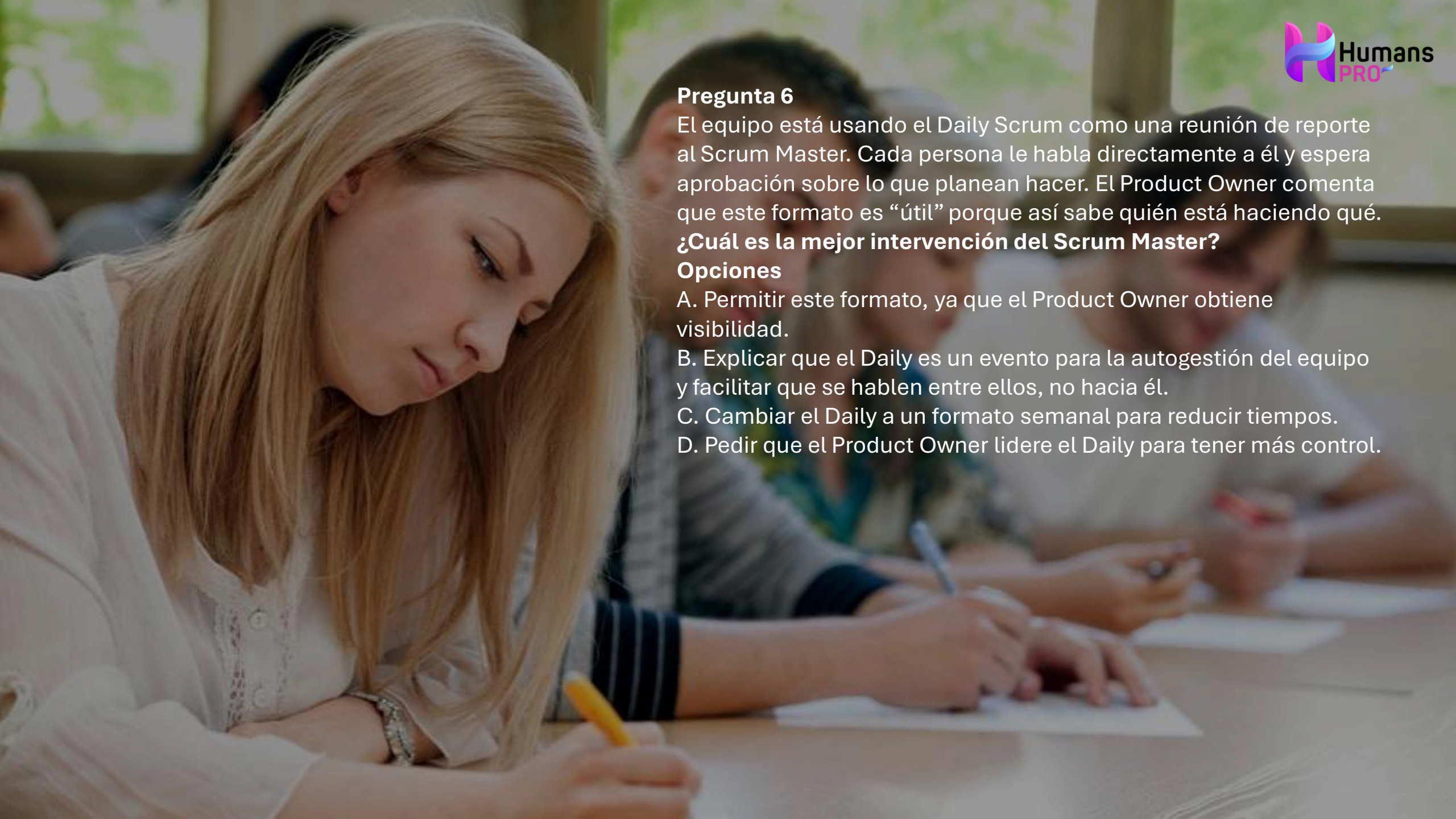
Opciones

A. Entregar las métricas individualizadas para mantener una buena relación con la gerencia.

B. Negarse completamente y citar la Guía Scrum de forma literal.

C. Educar a la gerencia sobre métricas de flujo, valor y equipo, explicando por qué medir individuos genera comportamientos disfuncionales.

D. Pedir al equipo que vote quién es el menos productivo para evitar dar métricas objetivas.



Pregunta 6

El equipo está usando el Daily Scrum como una reunión de reporte al Scrum Master. Cada persona le habla directamente a él y espera aprobación sobre lo que planean hacer. El Product Owner comenta que este formato es “útil” porque así sabe quién está haciendo qué.

¿Cuál es la mejor intervención del Scrum Master?

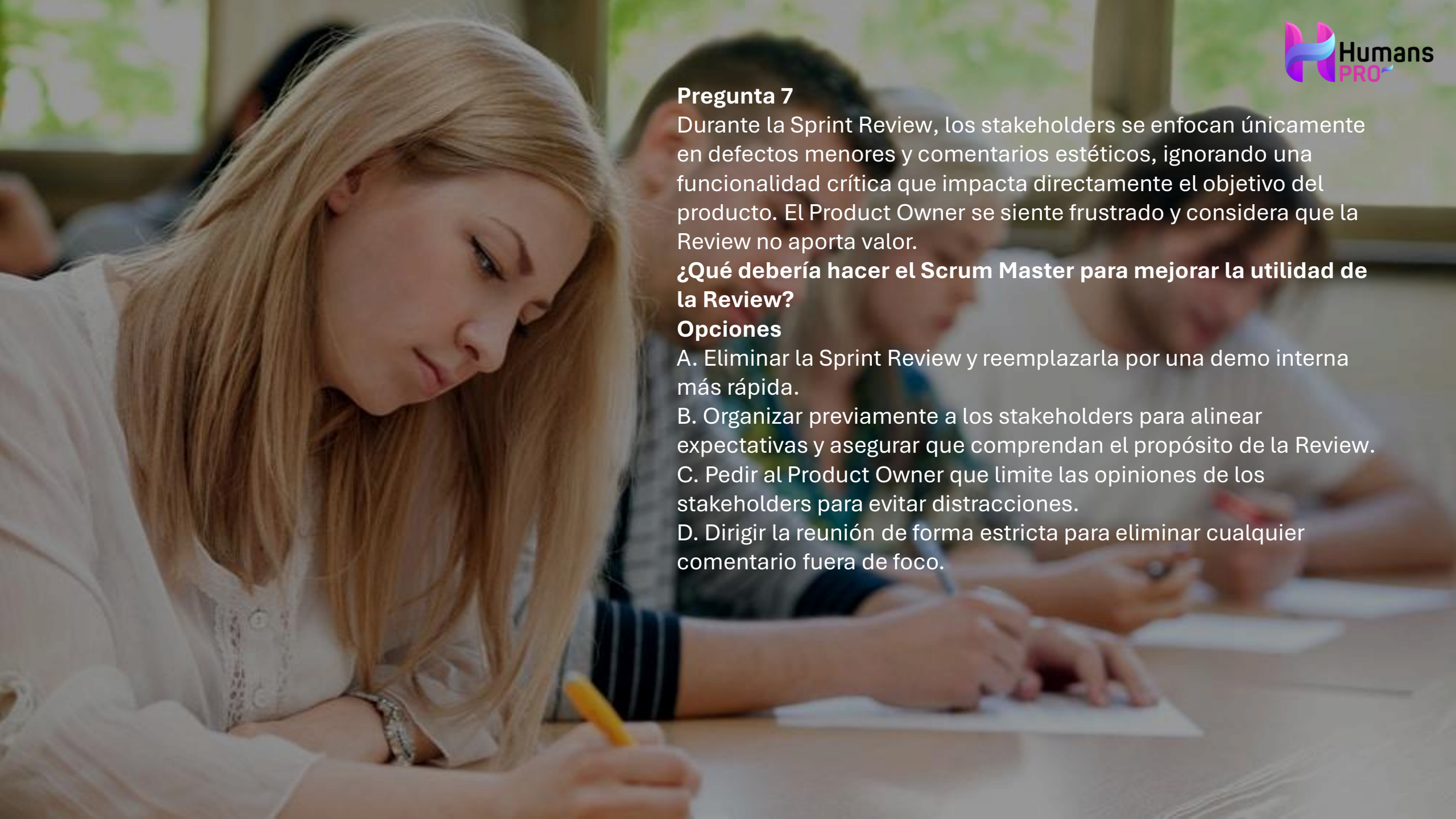
Opciones

A. Permitir este formato, ya que el Product Owner obtiene visibilidad.

B. Explicar que el Daily es un evento para la autogestión del equipo y facilitar que se hablen entre ellos, no hacia él.

C. Cambiar el Daily a un formato semanal para reducir tiempos.

D. Pedir que el Product Owner lidere el Daily para tener más control.



Pregunta 7

Durante la Sprint Review, los stakeholders se enfocan únicamente en defectos menores y comentarios estéticos, ignorando una funcionalidad crítica que impacta directamente el objetivo del producto. El Product Owner se siente frustrado y considera que la Review no aporta valor.

¿Qué debería hacer el Scrum Master para mejorar la utilidad de la Review?

Opciones

A. Eliminar la Sprint Review y reemplazarla por una demo interna más rápida.

B. Organizar previamente a los stakeholders para alinear expectativas y asegurar que comprendan el propósito de la Review.

C. Pedir al Product Owner que limite las opiniones de los stakeholders para evitar distracciones.

D. Dirigir la reunión de forma estricta para eliminar cualquier comentario fuera de foco.



Pregunta 8

Un miembro del equipo siempre acude al Scrum Master para resolver problemas técnicos, incluso cuando otros desarrolladores podrían ayudar. La dependencia está ralentizando el progreso del equipo y afectando la autogestión.

¿Qué acción es la más adecuada para el Scrum Master?

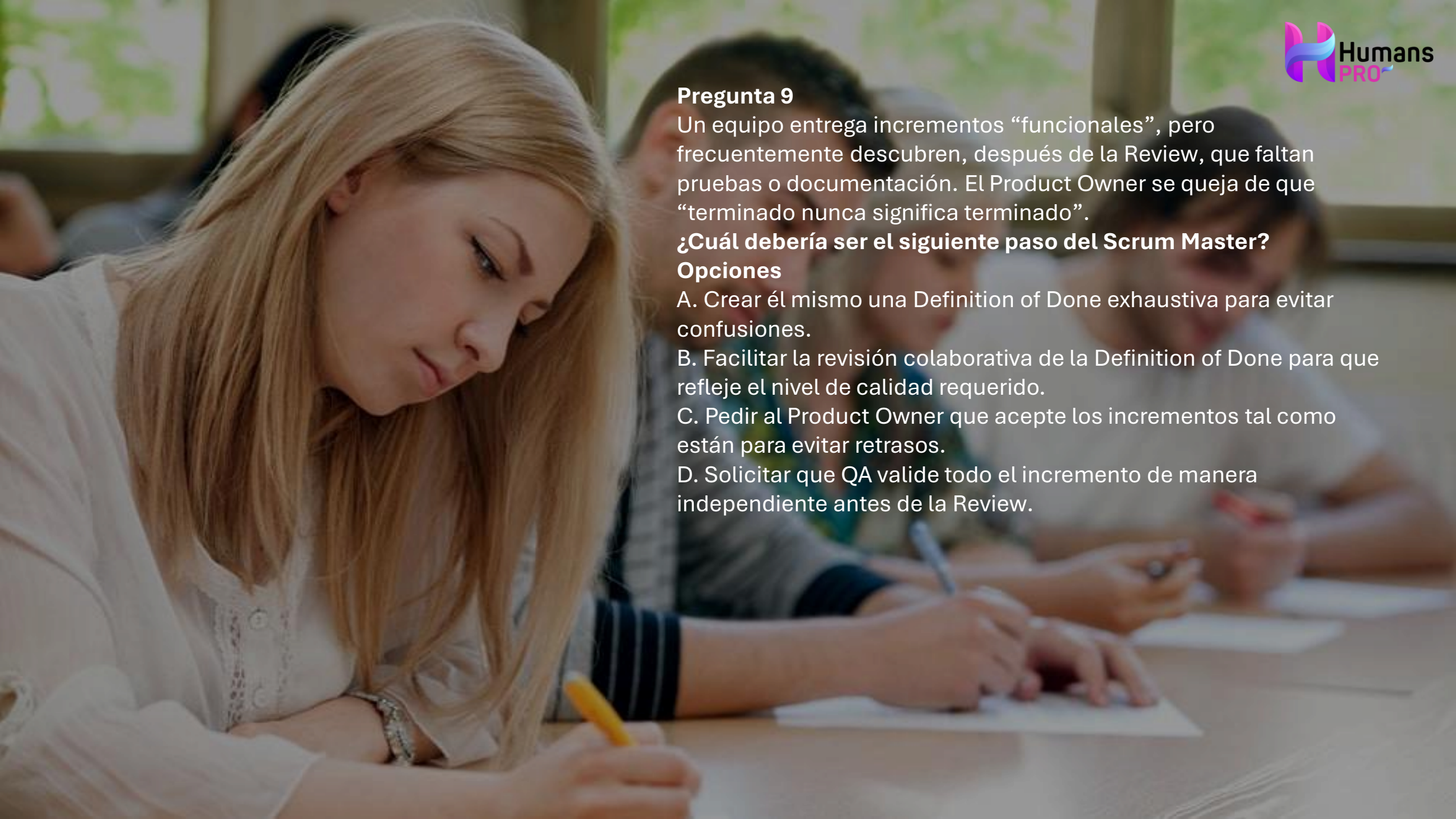
Opciones

A. Resolver los problemas cada vez, para no bloquear al equipo.

B. Negarse a ayudar y pedir siempre que lo resuelvan solos.

C. Facilitar un espacio donde el equipo comparta conocimiento y reduzca la dependencia hacia el Scrum Master.

D. Escalar la situación a la gerencia como “riesgo de incapacidad técnica”.



Pregunta 9

Un equipo entrega incrementos “funcionales”, pero frecuentemente descubren, después de la Review, que faltan pruebas o documentación. El Product Owner se queja de que “terminado nunca significa terminado”.

¿Cuál debería ser el siguiente paso del Scrum Master?

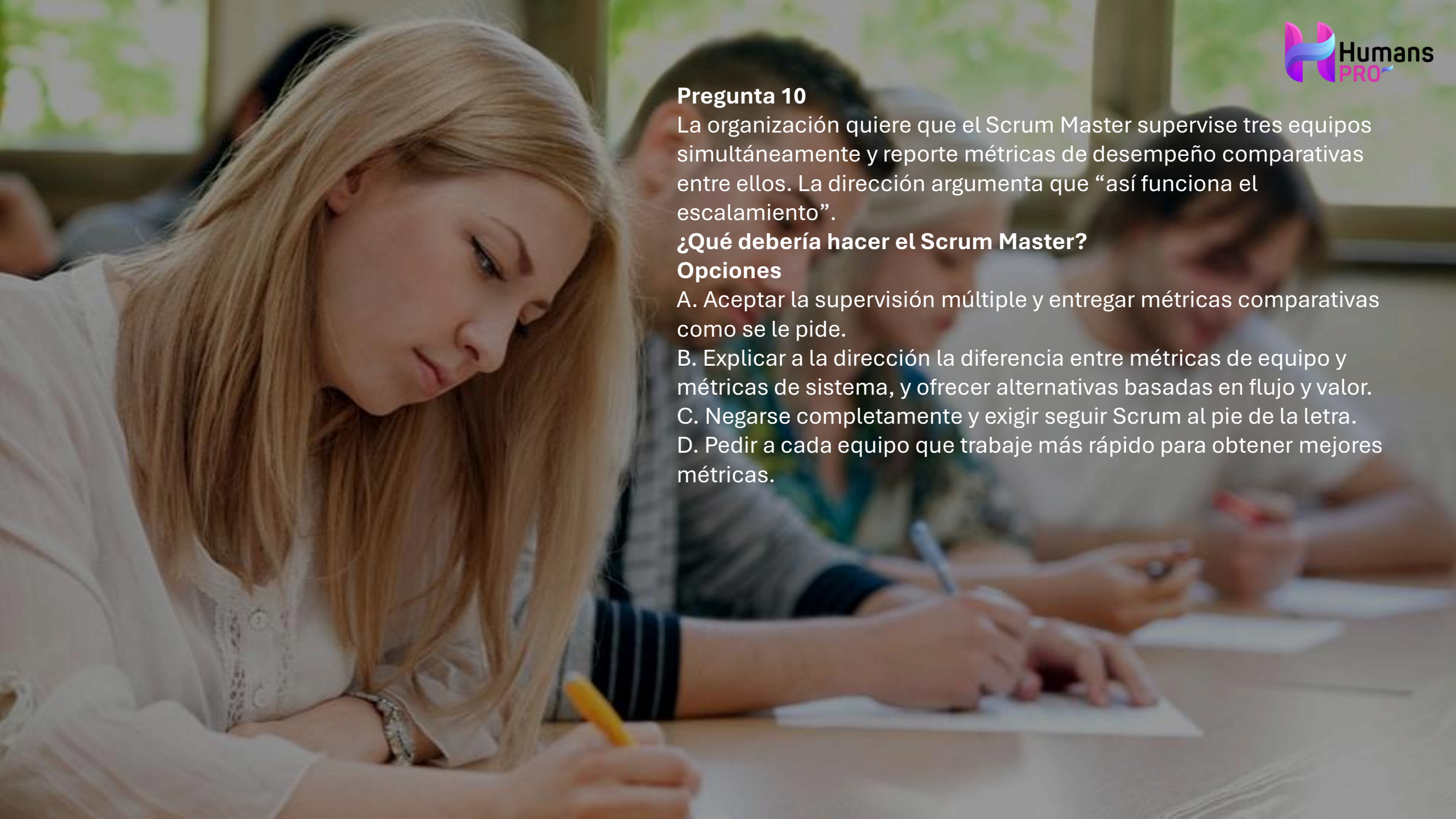
Opciones

A. Crear él mismo una Definition of Done exhaustiva para evitar confusiones.

B. Facilitar la revisión colaborativa de la Definition of Done para que refleje el nivel de calidad requerido.

C. Pedir al Product Owner que acepte los incrementos tal como están para evitar retrasos.

D. Solicitar que QA valide todo el incremento de manera independiente antes de la Review.



Pregunta 10

La organización quiere que el Scrum Master supervise tres equipos simultáneamente y reporte métricas de desempeño comparativas entre ellos. La dirección argumenta que “así funciona el escalamiento”.

¿Qué debería hacer el Scrum Master?

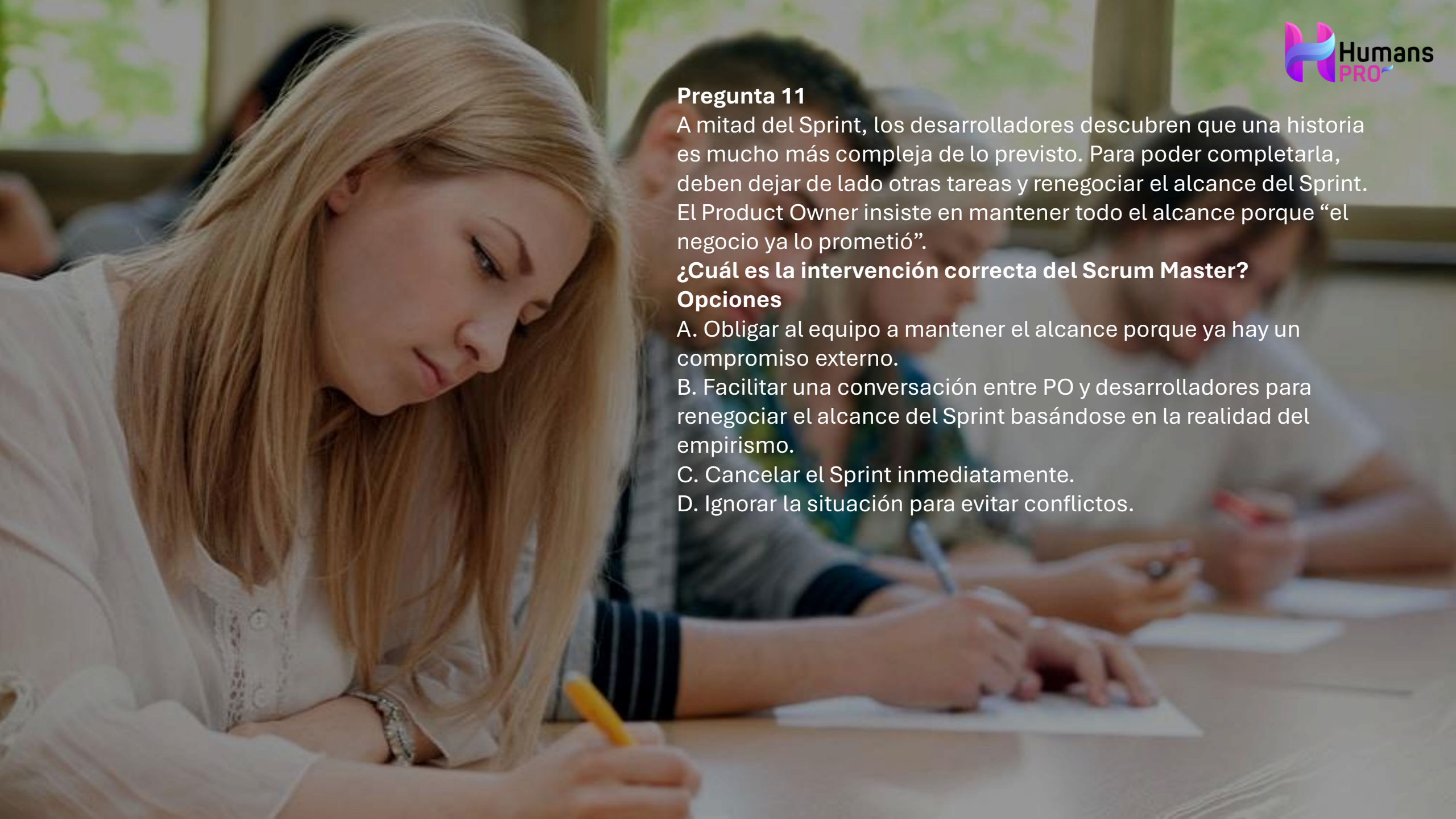
Opciones

A. Aceptar la supervisión múltiple y entregar métricas comparativas como se le pide.

B. Explicar a la dirección la diferencia entre métricas de equipo y métricas de sistema, y ofrecer alternativas basadas en flujo y valor.

C. Negarse completamente y exigir seguir Scrum al pie de la letra.

D. Pedir a cada equipo que trabaje más rápido para obtener mejores métricas.



Pregunta 11

A mitad del Sprint, los desarrolladores descubren que una historia es mucho más compleja de lo previsto. Para poder completarla, deben dejar de lado otras tareas y renegociar el alcance del Sprint. El Product Owner insiste en mantener todo el alcance porque “el negocio ya lo prometió”.

¿Cuál es la intervención correcta del Scrum Master?

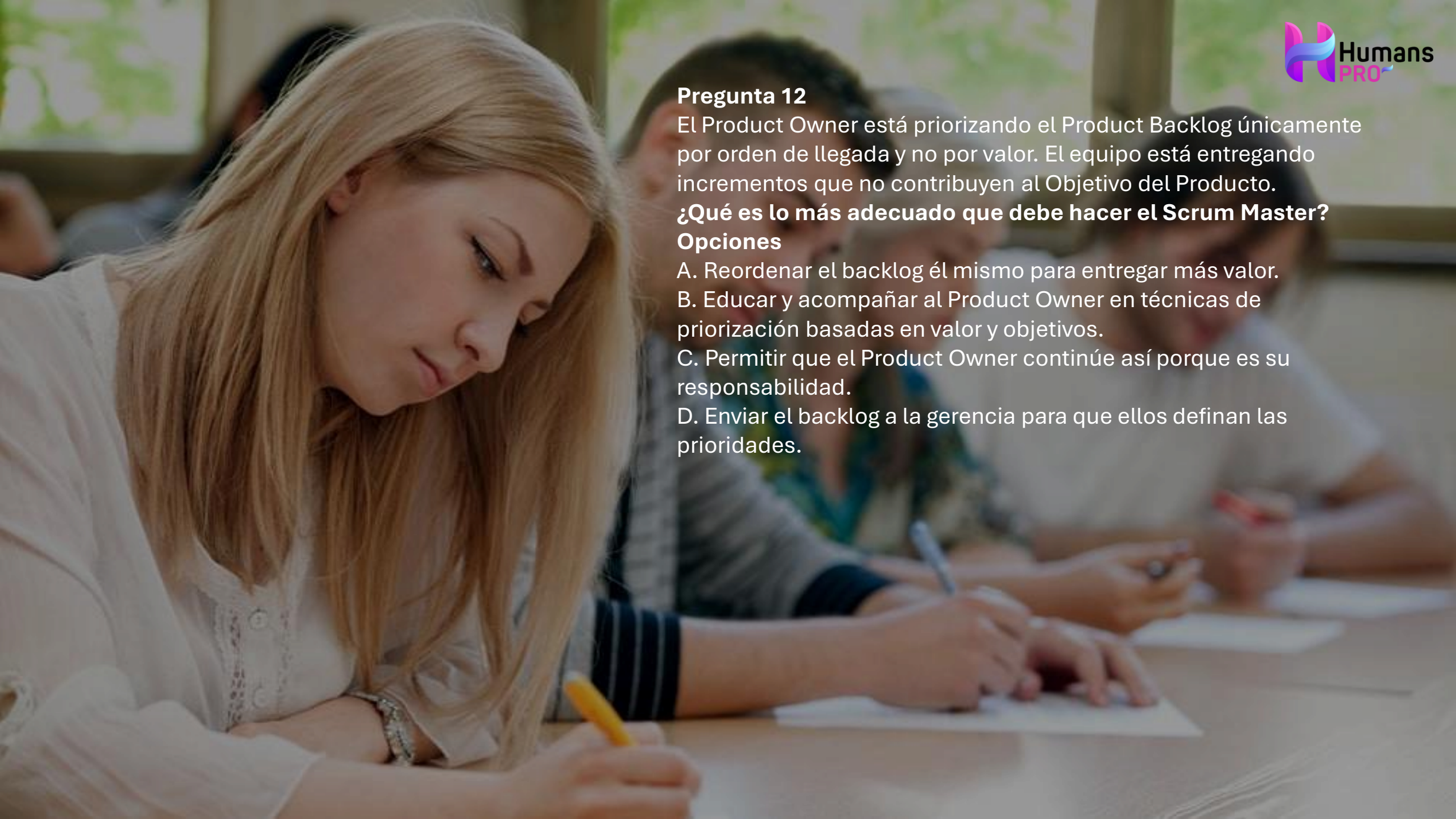
Opciones

A. Obligar al equipo a mantener el alcance porque ya hay un compromiso externo.

B. Facilitar una conversación entre PO y desarrolladores para renegociar el alcance del Sprint basándose en la realidad del empirismo.

C. Cancelar el Sprint inmediatamente.

D. Ignorar la situación para evitar conflictos.



Pregunta 12

El Product Owner está priorizando el Product Backlog únicamente por orden de llegada y no por valor. El equipo está entregando incrementos que no contribuyen al Objetivo del Producto.

¿Qué es lo más adecuado que debe hacer el Scrum Master?

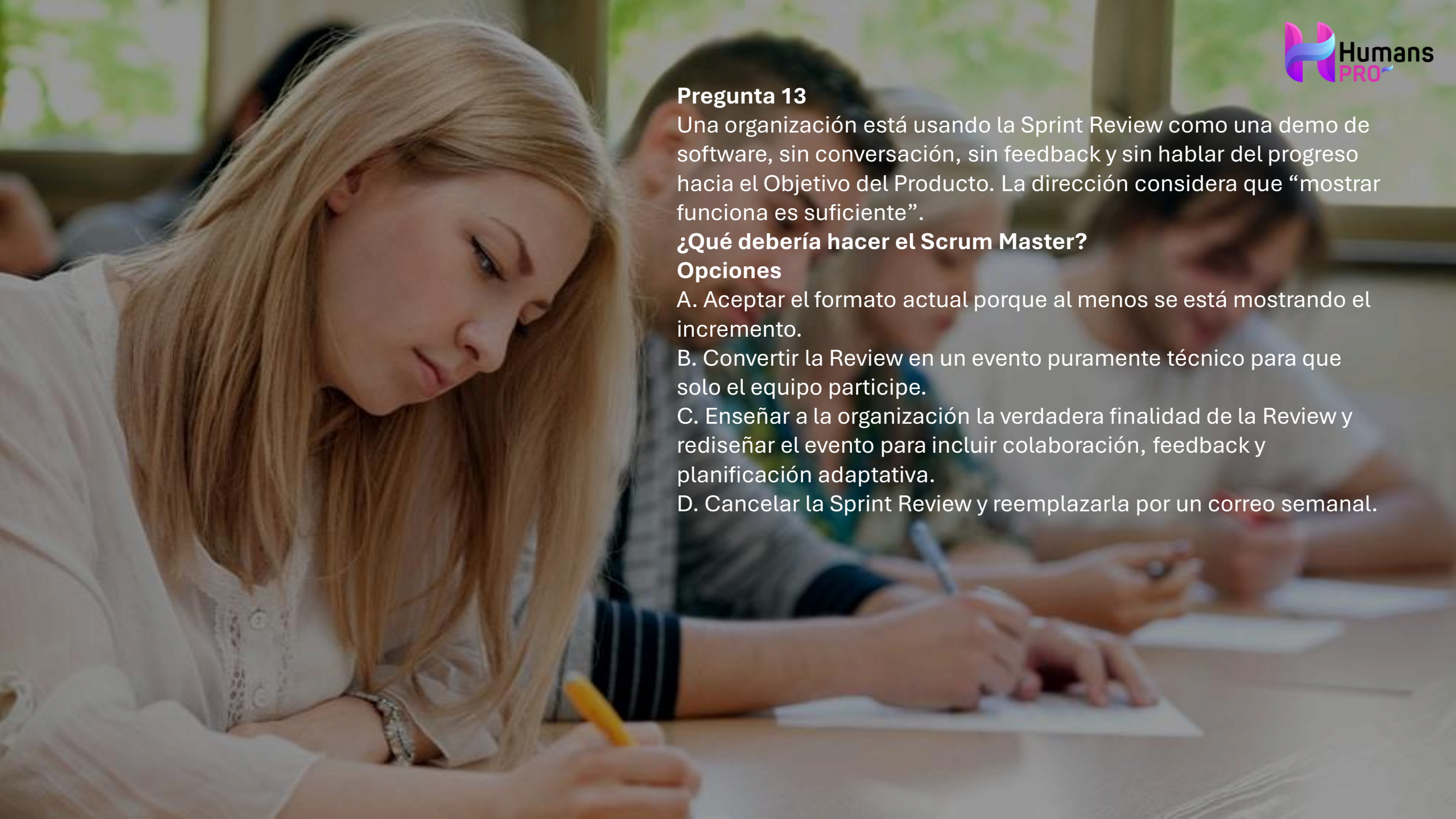
Opciones

A. Reordenar el backlog él mismo para entregar más valor.

B. Educar y acompañar al Product Owner en técnicas de priorización basadas en valor y objetivos.

C. Permitir que el Product Owner continúe así porque es su responsabilidad.

D. Enviar el backlog a la gerencia para que ellos definan las prioridades.



Pregunta 13

Una organización está usando la Sprint Review como una demo de software, sin conversación, sin feedback y sin hablar del progreso hacia el Objetivo del Producto. La dirección considera que “mostrar funciona es suficiente”.

¿Qué debería hacer el Scrum Master?

Opciones

A. Aceptar el formato actual porque al menos se está mostrando el incremento.

B. Convertir la Review en un evento puramente técnico para que solo el equipo participe.

C. Enseñar a la organización la verdadera finalidad de la Review y rediseñar el evento para incluir colaboración, feedback y planificación adaptativa.

D. Cancelar la Sprint Review y reemplazarla por un correo semanal.



Pregunta 14

Últimamente, el equipo llega tarde a los eventos, participa poco y se muestran desconectados del Objetivo del Sprint. No hay conflictos visibles, simplemente apatía.

¿Cuál sería la acción más efectiva del Scrum Master?

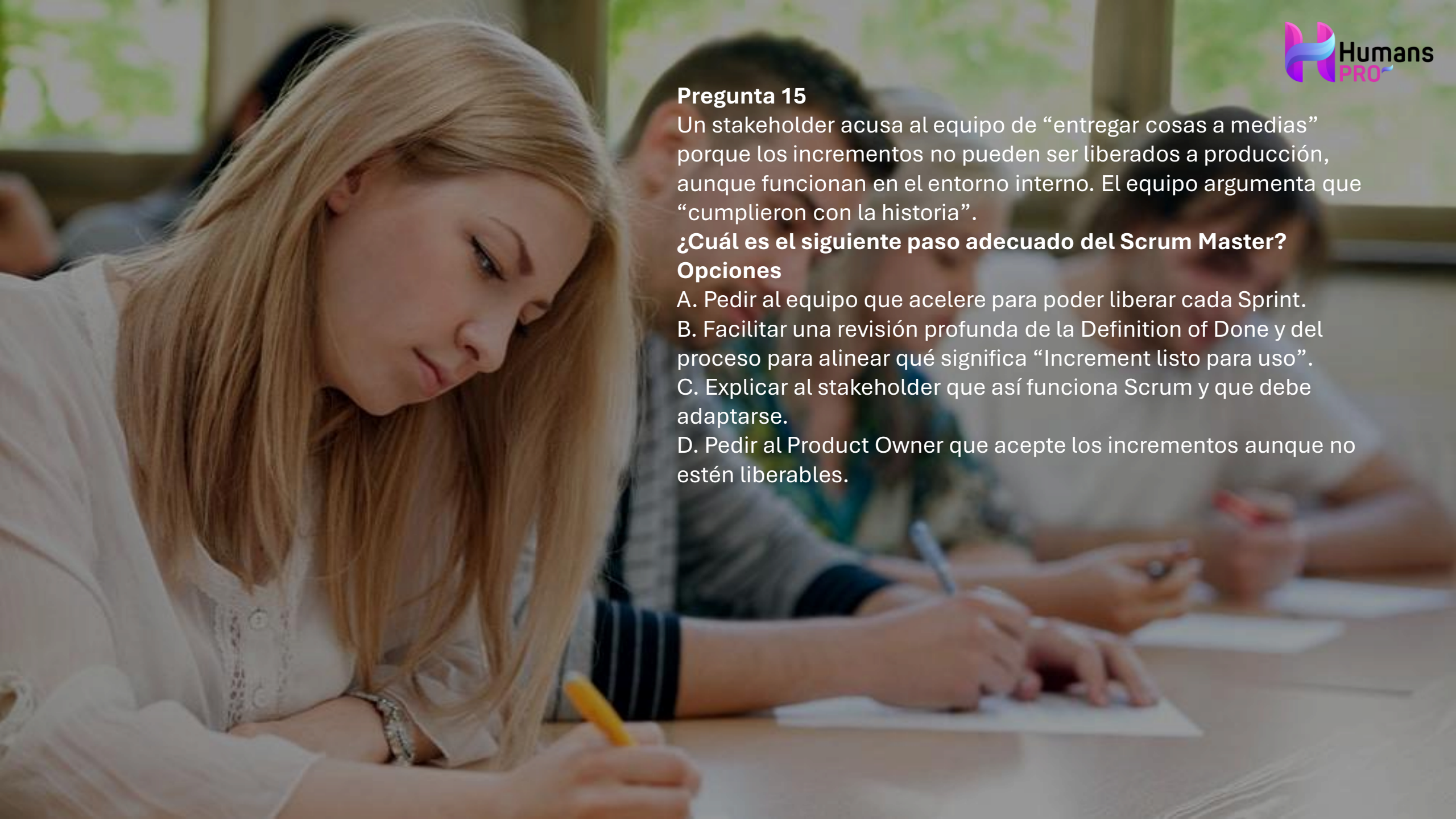
Opciones

A. Llamar la atención del equipo y recordarles que deben ser más disciplinados.

B. Forzar retrospectivas más largas hasta identificar el problema.

C. Explorar, mediante coaching y técnicas de facilitación (como 1–2–4-All, Mad-Sad-Glad), las causas subyacentes relacionadas con motivación, seguridad o claridad.

D. Solicitar a Recursos Humanos que haga una evaluación de desempeño.



Pregunta 15

Un stakeholder acusa al equipo de “entregar cosas a medias” porque los incrementos no pueden ser liberados a producción, aunque funcionan en el entorno interno. El equipo argumenta que “cumplieron con la historia”.

¿Cuál es el siguiente paso adecuado del Scrum Master?

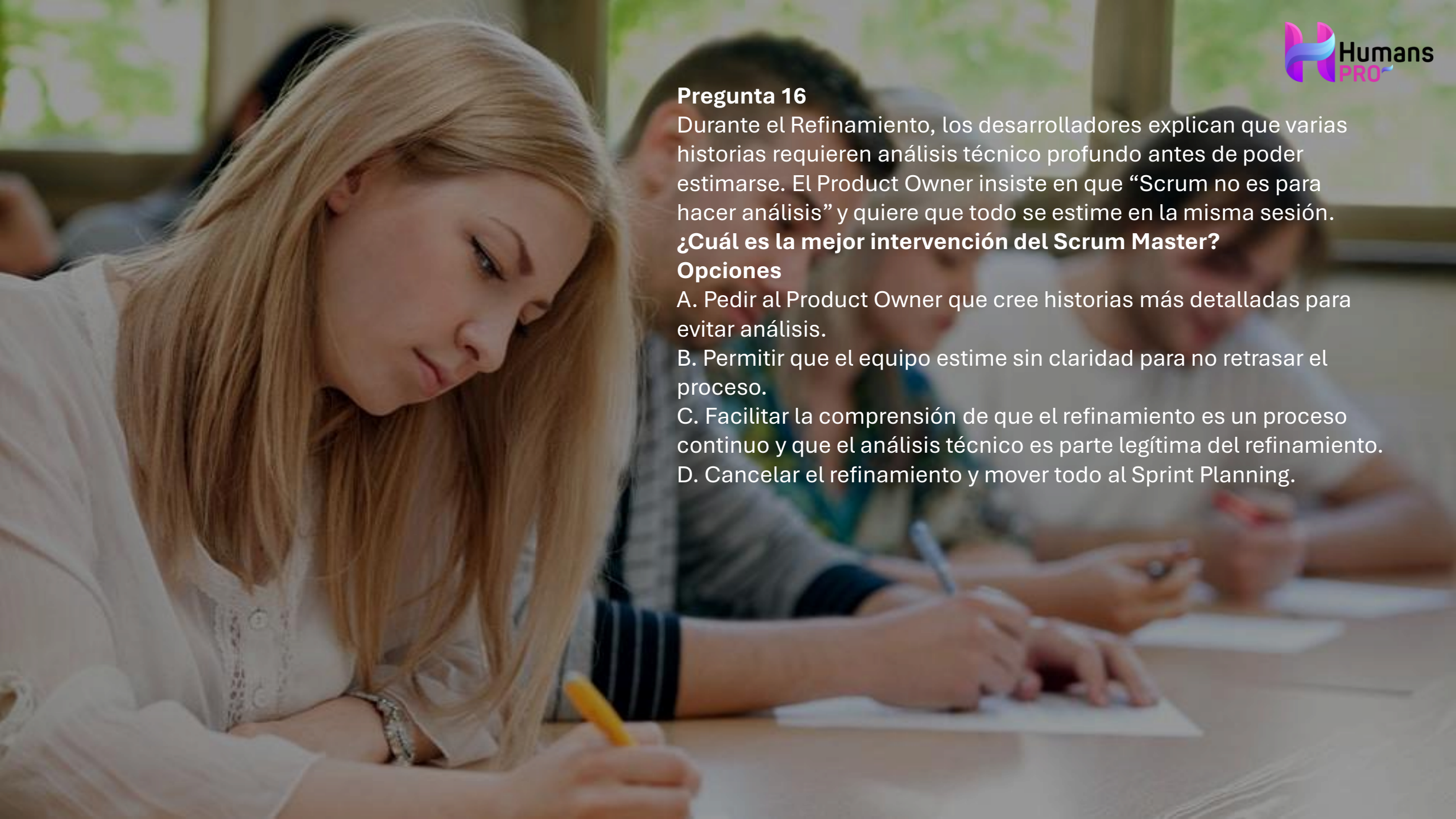
Opciones

A. Pedir al equipo que acelere para poder liberar cada Sprint.

B. Facilitar una revisión profunda de la Definition of Done y del proceso para alinear qué significa “Increment listo para uso”.

C. Explicar al stakeholder que así funciona Scrum y que debe adaptarse.

D. Pedir al Product Owner que acepte los incrementos aunque no estén liberables.



Pregunta 16

Durante el Refinamiento, los desarrolladores explican que varias historias requieren análisis técnico profundo antes de poder estimarse. El Product Owner insiste en que “Scrum no es para hacer análisis” y quiere que todo se estime en la misma sesión.

¿Cuál es la mejor intervención del Scrum Master?

Opciones

A. Pedir al Product Owner que cree historias más detalladas para evitar análisis.

B. Permitir que el equipo estime sin claridad para no retrasar el proceso.

C. Facilitar la comprensión de que el refinamiento es un proceso continuo y que el análisis técnico es parte legítima del refinamiento.

D. Cancelar el refinamiento y mover todo al Sprint Planning.



Pregunta 17

En varios Sprints, el equipo está eligiendo un Objetivo del Sprint que simplemente agrupa tareas (“Avanzar en backend”, “Refactorizar módulos”) sin un propósito claro o conexión con el Objetivo del Producto.

¿Qué debería hacer el Scrum Master?

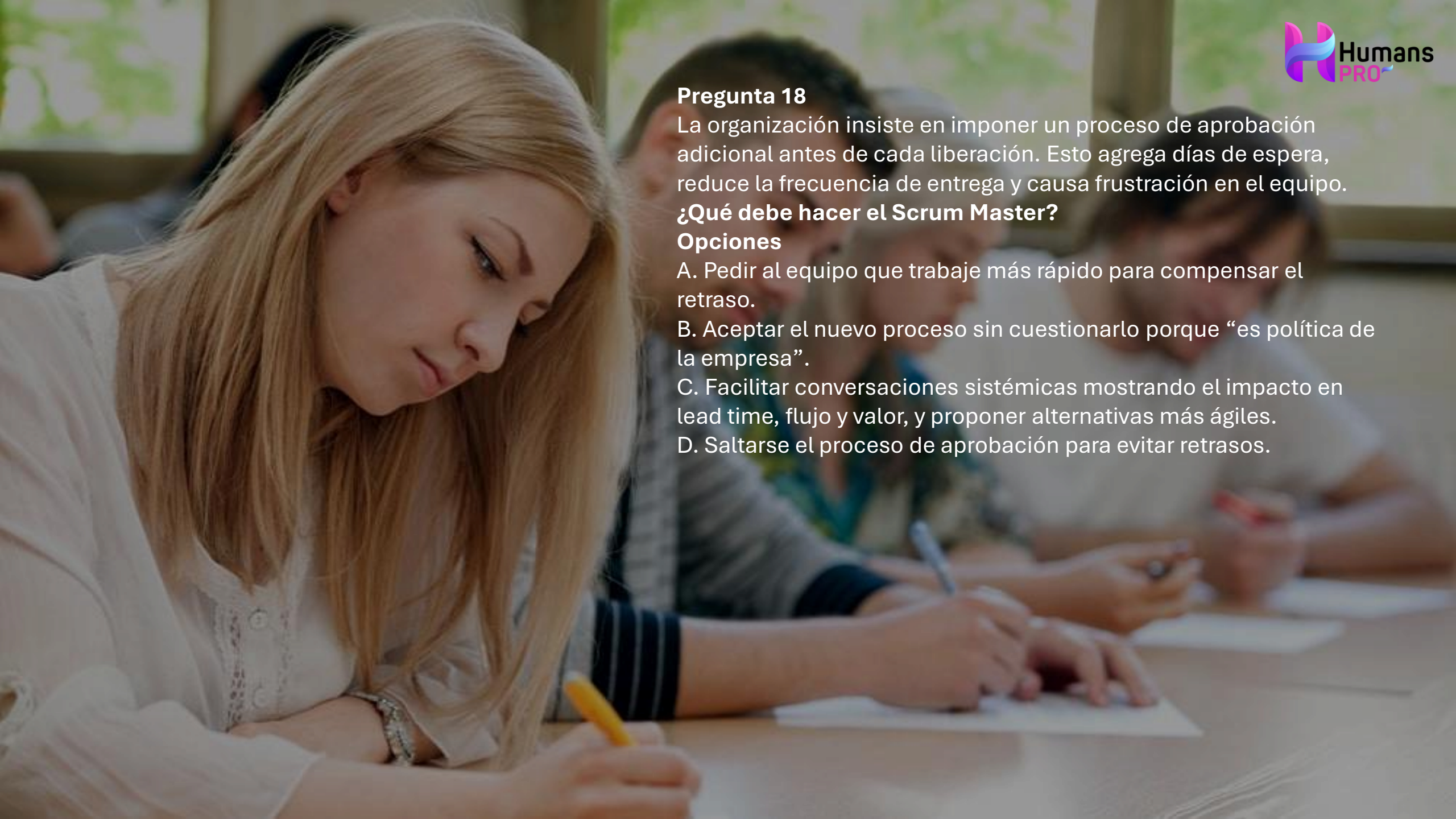
Opciones

A. Forzar al Product Owner a definir un objetivo más valioso.

B. Enseñar cómo crear Sprint Goals orientados a valor y facilitar discusiones para identificar resultados, no actividades.

C. Permitir esos objetivos mientras no afecten la velocidad.

D. Eliminar el Objetivo del Sprint si el equipo no está maduro para usarlo.



Pregunta 18

La organización insiste en imponer un proceso de aprobación adicional antes de cada liberación. Esto agrega días de espera, reduce la frecuencia de entrega y causa frustración en el equipo.

¿Qué debe hacer el Scrum Master?

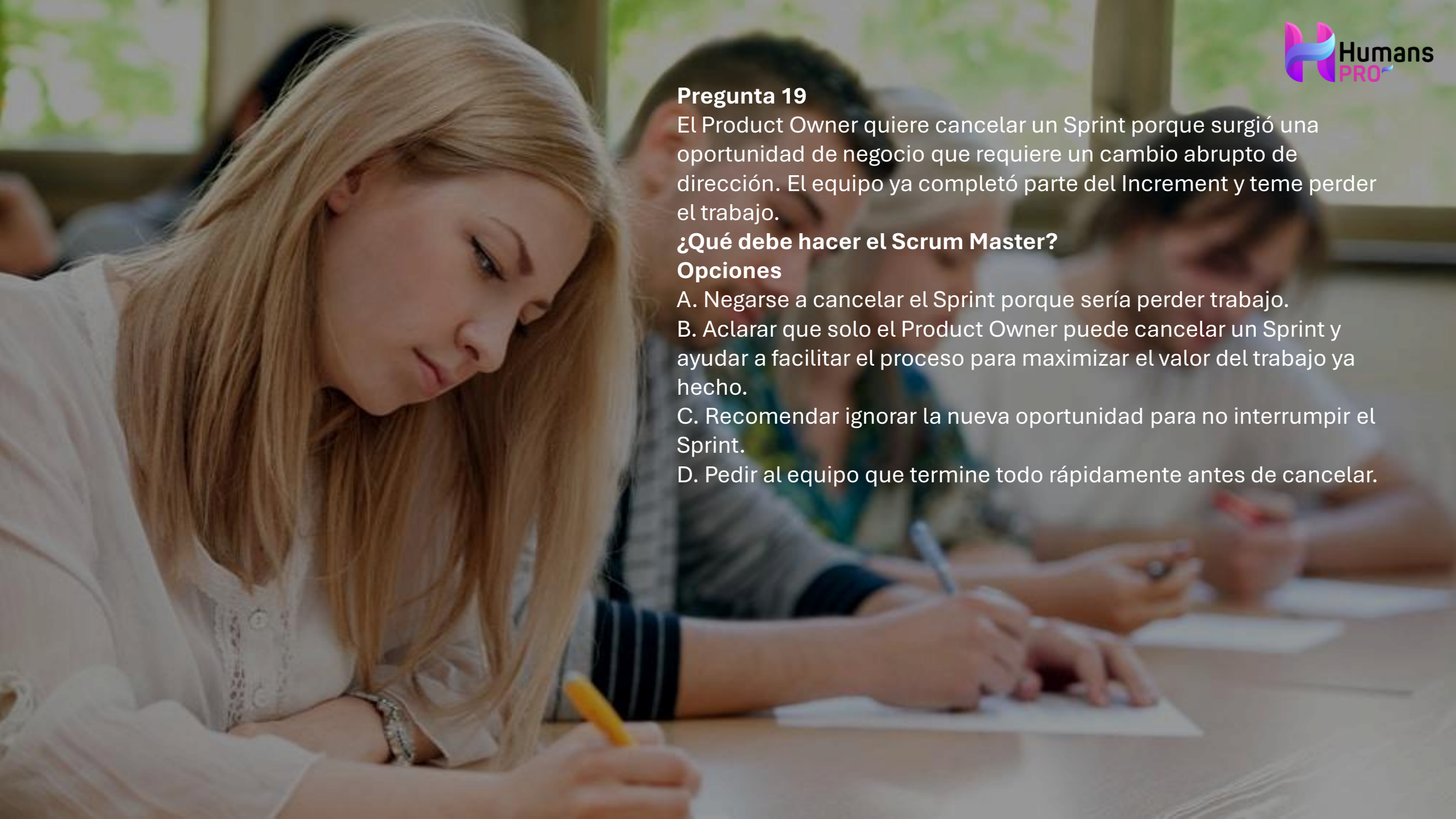
Opciones

A. Pedir al equipo que trabaje más rápido para compensar el retraso.

B. Aceptar el nuevo proceso sin cuestionarlo porque “es política de la empresa”.

C. Facilitar conversaciones sistémicas mostrando el impacto en lead time, flujo y valor, y proponer alternativas más ágiles.

D. Saltarse el proceso de aprobación para evitar retrasos.



Pregunta 19

El Product Owner quiere cancelar un Sprint porque surgió una oportunidad de negocio que requiere un cambio abrupto de dirección. El equipo ya completó parte del Increment y teme perder el trabajo.

¿Qué debe hacer el Scrum Master?

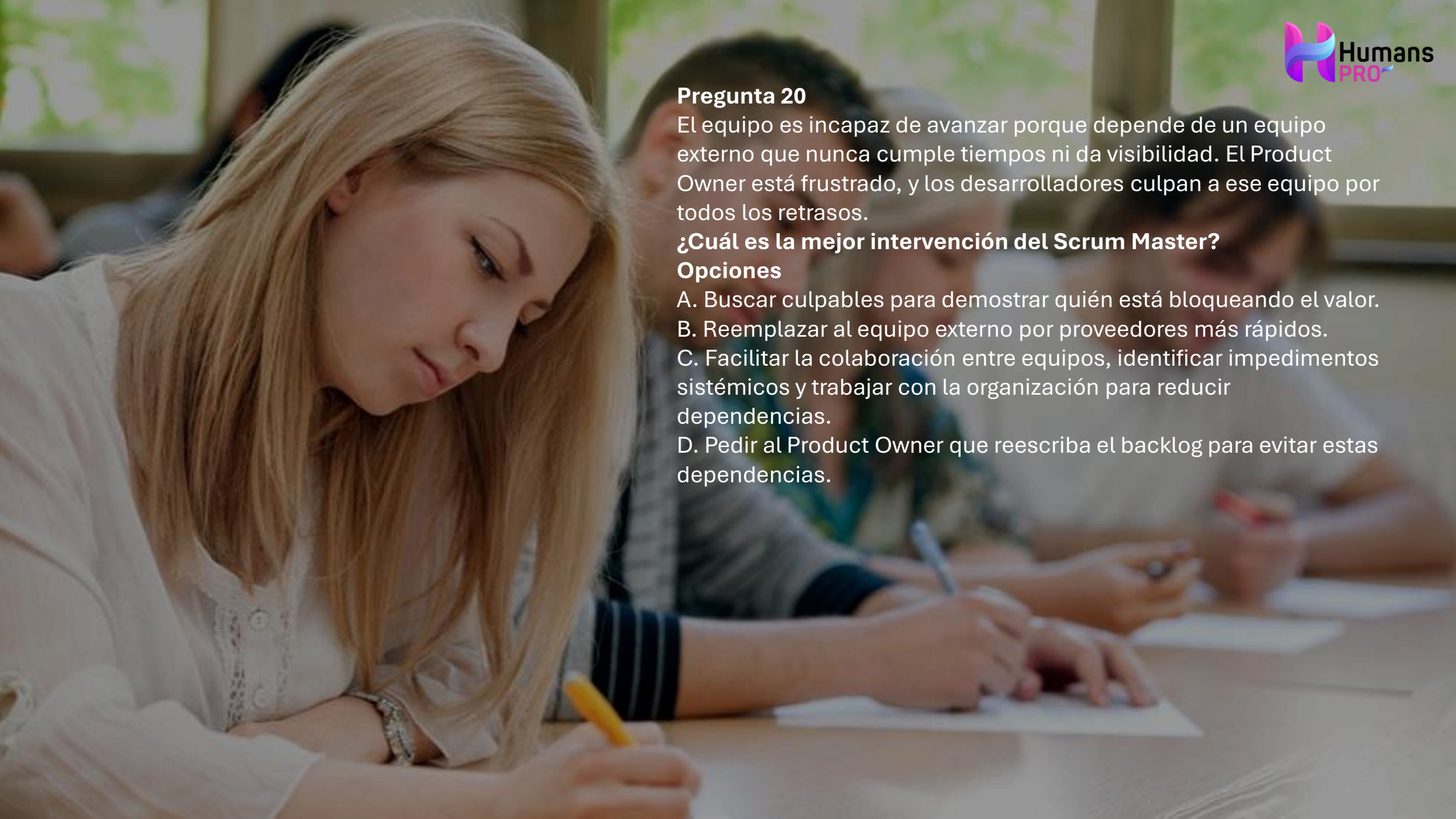
Opciones

A. Negarse a cancelar el Sprint porque sería perder trabajo.

B. Aclarar que solo el Product Owner puede cancelar un Sprint y ayudar a facilitar el proceso para maximizar el valor del trabajo ya hecho.

C. Recomendar ignorar la nueva oportunidad para no interrumpir el Sprint.

D. Pedir al equipo que termine todo rápidamente antes de cancelar.



Pregunta 20

El equipo es incapaz de avanzar porque depende de un equipo externo que nunca cumple tiempos ni da visibilidad. El Product Owner está frustrado, y los desarrolladores culpan a ese equipo por todos los retrasos.

¿Cuál es la mejor intervención del Scrum Master?

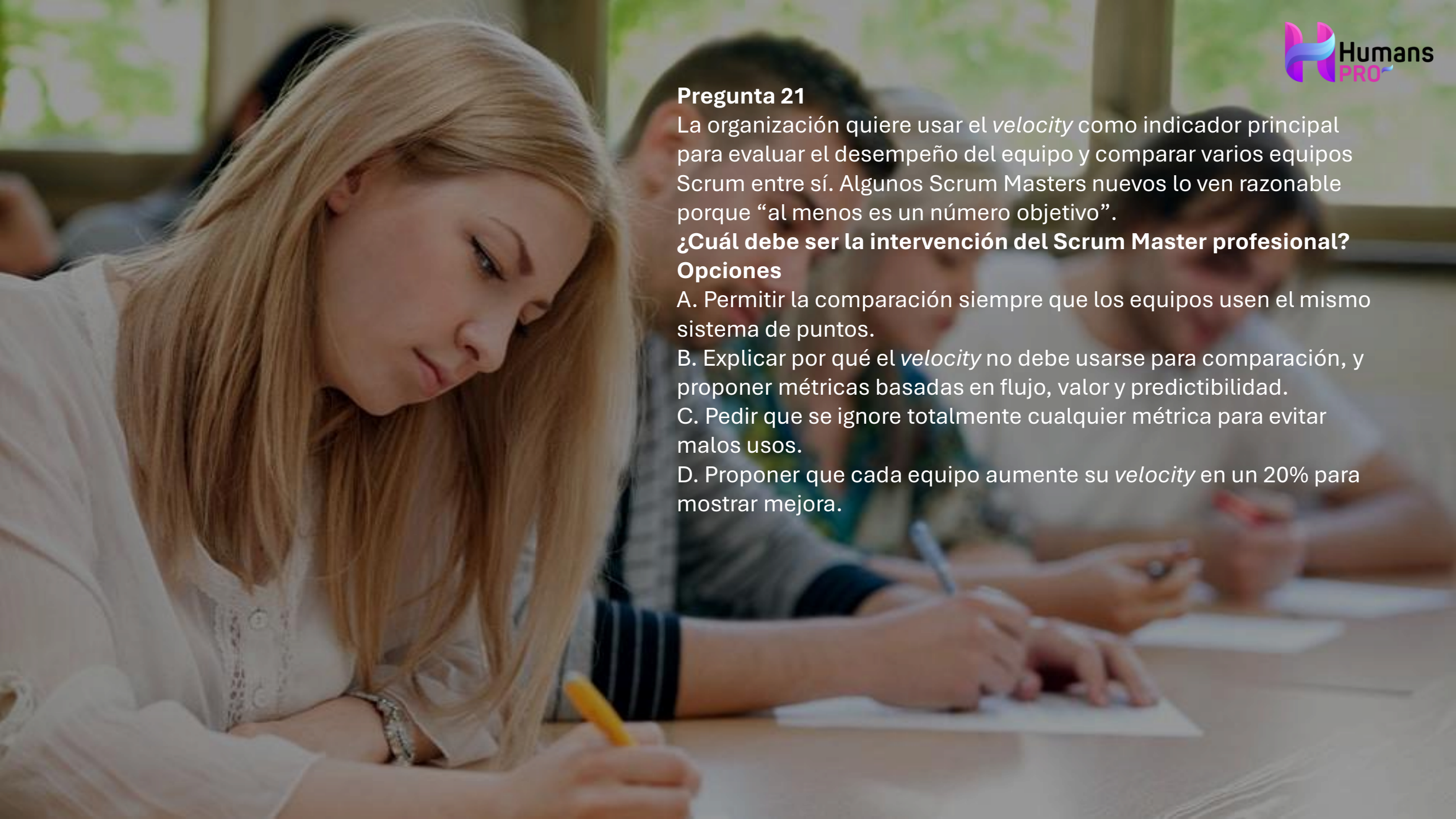
Opciones

A. Buscar culpables para demostrar quién está bloqueando el valor.

B. Reemplazar al equipo externo por proveedores más rápidos.

C. Facilitar la colaboración entre equipos, identificar impedimentos sistémicos y trabajar con la organización para reducir dependencias.

D. Pedir al Product Owner que reescriba el backlog para evitar estas dependencias.



Pregunta 21

La organización quiere usar el *velocity* como indicador principal para evaluar el desempeño del equipo y comparar varios equipos Scrum entre sí. Algunos Scrum Masters nuevos lo ven razonable porque “al menos es un número objetivo”.

¿Cuál debe ser la intervención del Scrum Master profesional?

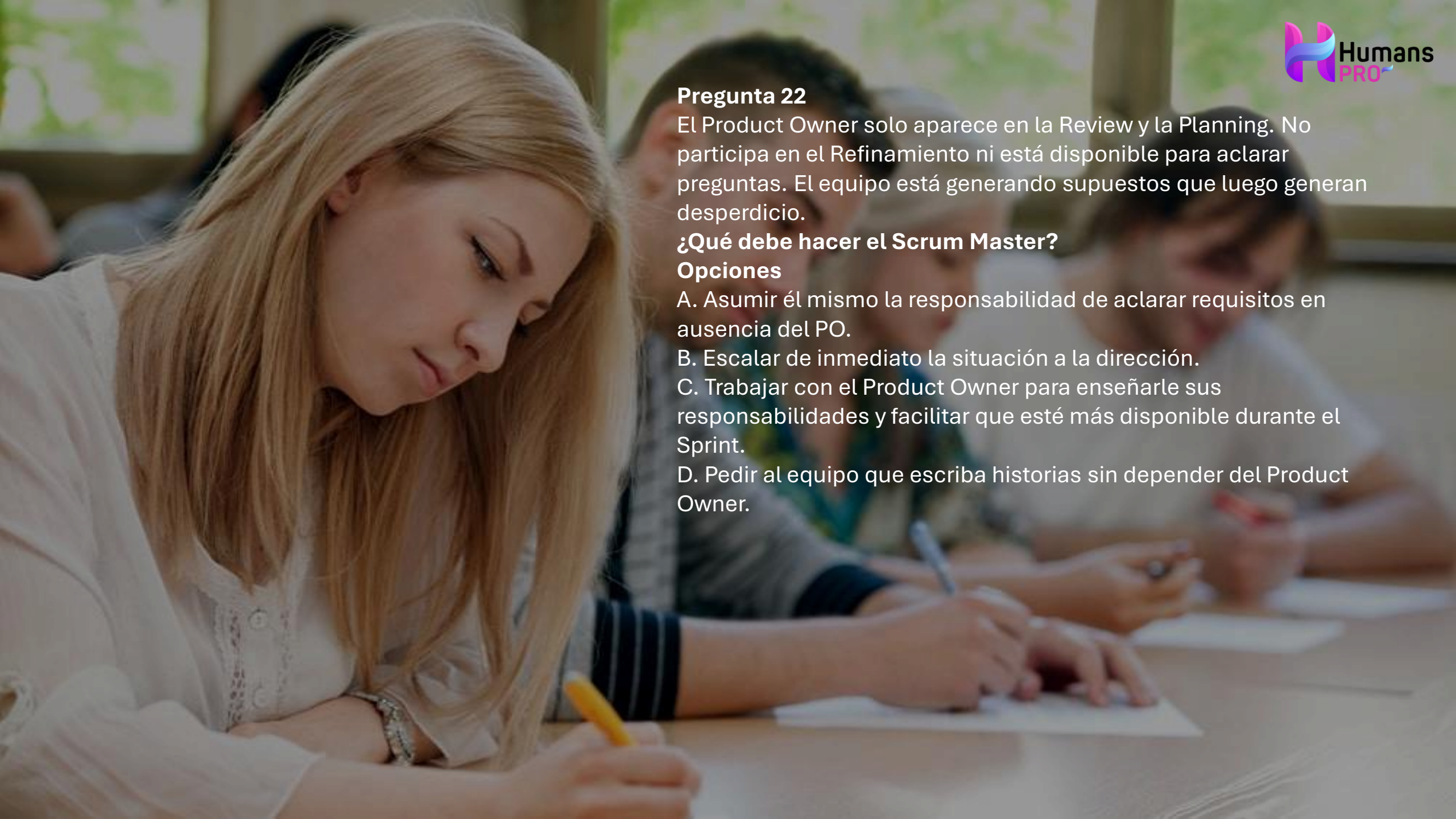
Opciones

A. Permitir la comparación siempre que los equipos usen el mismo sistema de puntos.

B. Explicar por qué el *velocity* no debe usarse para comparación, y proponer métricas basadas en flujo, valor y predictibilidad.

C. Pedir que se ignore totalmente cualquier métrica para evitar malos usos.

D. Proponer que cada equipo aumente su *velocity* en un 20% para mostrar mejora.



Pregunta 22

El Product Owner solo aparece en la Review y la Planning. No participa en el Refinamiento ni está disponible para aclarar preguntas. El equipo está generando supuestos que luego generan desperdicio.

¿Qué debe hacer el Scrum Master?

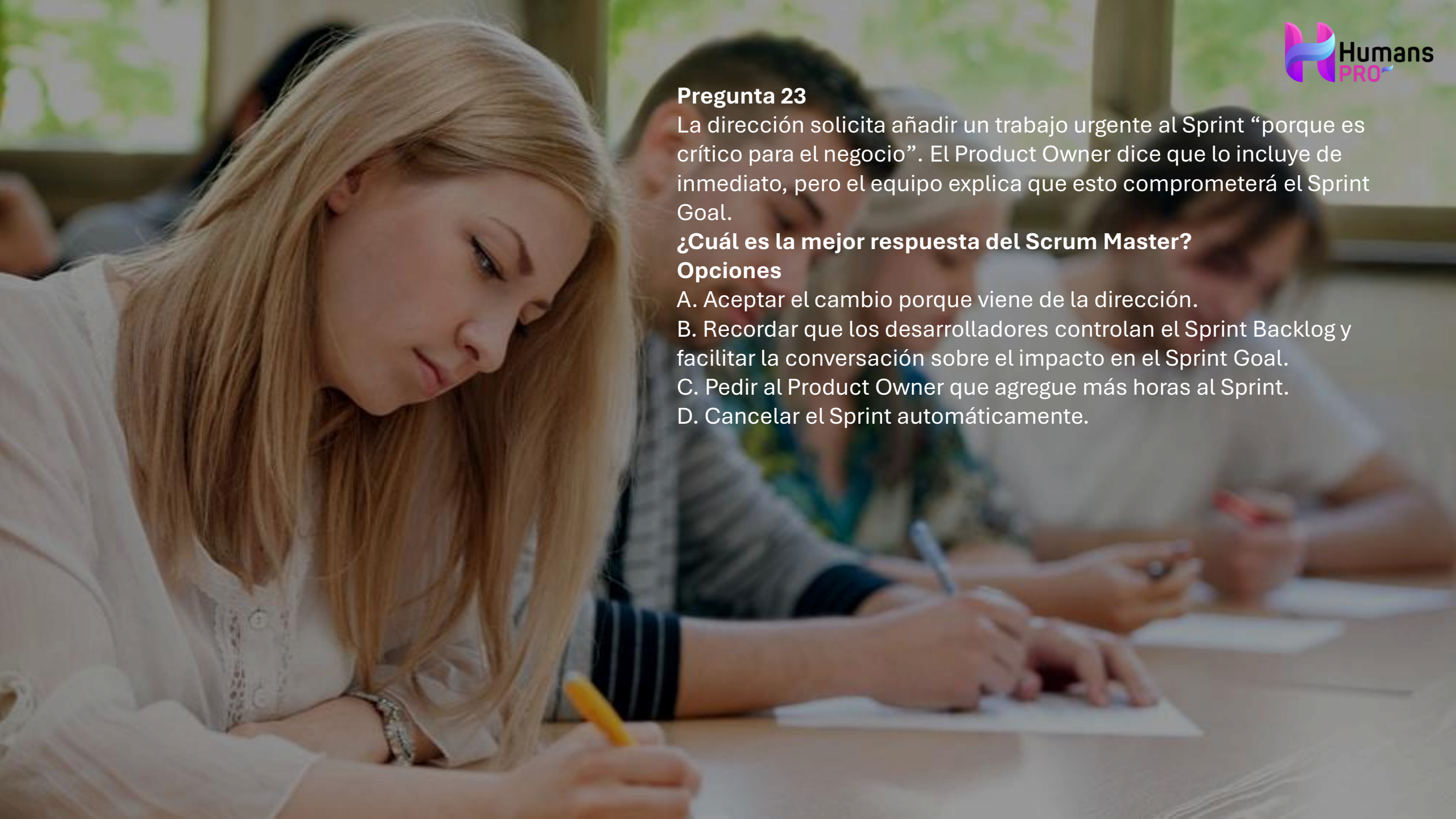
Opciones

A. Asumir él mismo la responsabilidad de aclarar requisitos en ausencia del PO.

B. Escalar de inmediato la situación a la dirección.

C. Trabajar con el Product Owner para enseñarle sus responsabilidades y facilitar que esté más disponible durante el Sprint.

D. Pedir al equipo que escriba historias sin depender del Product Owner.



Pregunta 23

La dirección solicita añadir un trabajo urgente al Sprint “porque es crítico para el negocio”. El Product Owner dice que lo incluye de inmediato, pero el equipo explica que esto comprometerá el Sprint Goal.

¿Cuál es la mejor respuesta del Scrum Master?

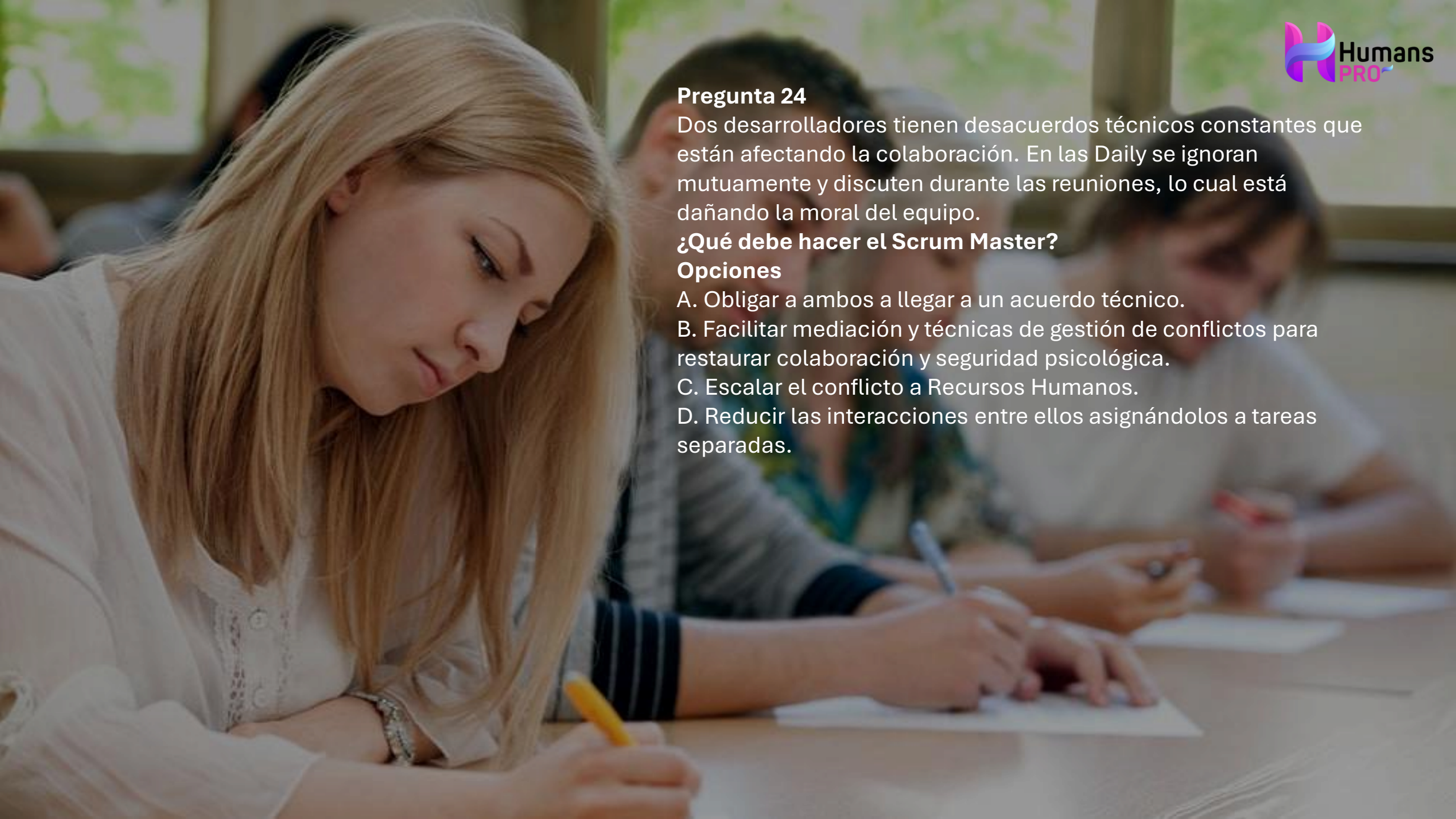
Opciones

A. Aceptar el cambio porque viene de la dirección.

B. Recordar que los desarrolladores controlan el Sprint Backlog y facilitar la conversación sobre el impacto en el Sprint Goal.

C. Pedir al Product Owner que agregue más horas al Sprint.

D. Cancelar el Sprint automáticamente.



Pregunta 24

Dos desarrolladores tienen desacuerdos técnicos constantes que están afectando la colaboración. En las Daily se ignoran mutuamente y discuten durante las reuniones, lo cual está dañando la moral del equipo.

¿Qué debe hacer el Scrum Master?

Opciones

A. Obligar a ambos a llegar a un acuerdo técnico.

B. Facilitar mediación y técnicas de gestión de conflictos para restaurar colaboración y seguridad psicológica.

C. Escalar el conflicto a Recursos Humanos.

D. Reducir las interacciones entre ellos asignándolos a tareas separadas.



Pregunta 25

En varias Sprint Reviews, los stakeholders dan feedback contradictorio o demasiado tardío, generando re-trabajo. El Product Owner está frustrado y dice que “Scrum no está funcionando”.

¿Cuál es la intervención más efectiva del Scrum Master?

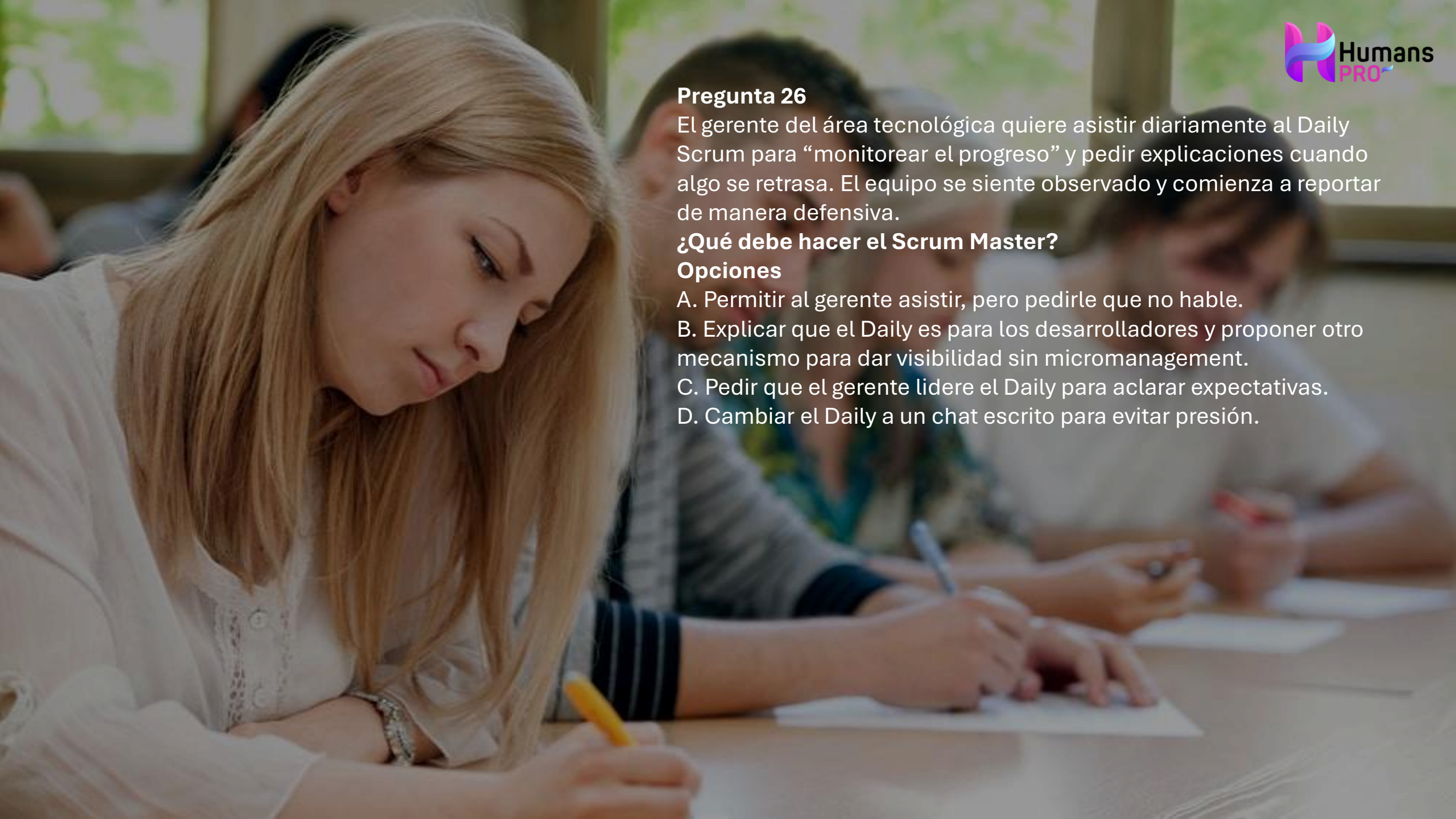
Opciones

A. Reducir la Review a una demo técnica para minimizar feedback.

B. Pedir a los stakeholders que envíen feedback por correo después del Sprint.

C. Trabajar con stakeholders y PO para mejorar la preparación previa y la claridad del Objetivo del Producto, y asegurar participación consistente.

D. Cancelar las Reviews y presentar un informe mensual.



Pregunta 26

El gerente del área tecnológica quiere asistir diariamente al Daily Scrum para “monitorear el progreso” y pedir explicaciones cuando algo se retrasa. El equipo se siente observado y comienza a reportar de manera defensiva.

¿Qué debe hacer el Scrum Master?

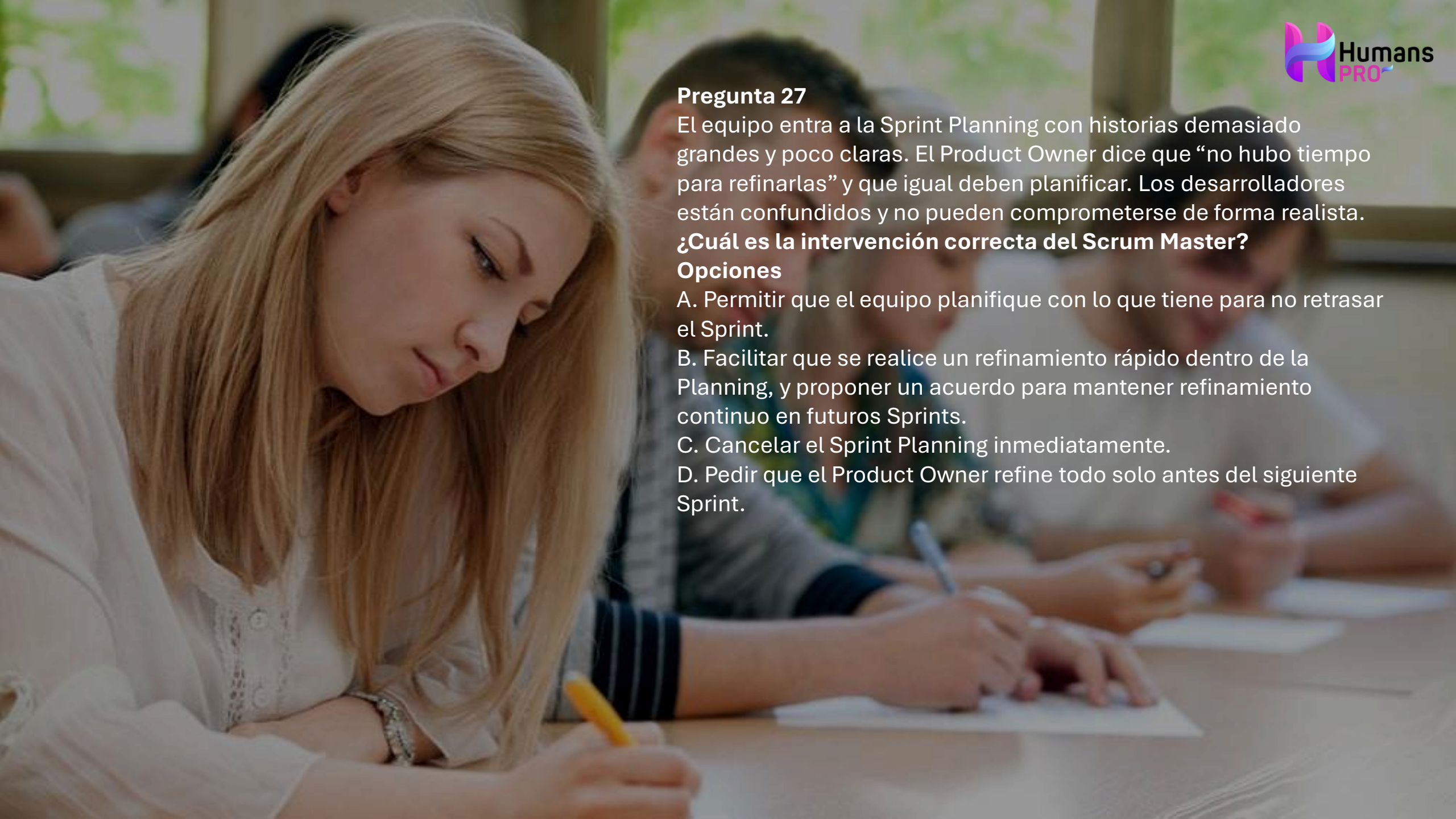
Opciones

A. Permitir al gerente asistir, pero pedirle que no hable.

B. Explicar que el Daily es para los desarrolladores y proponer otro mecanismo para dar visibilidad sin micromanagement.

C. Pedir que el gerente lidere el Daily para aclarar expectativas.

D. Cambiar el Daily a un chat escrito para evitar presión.



Pregunta 27

El equipo entra a la Sprint Planning con historias demasiado grandes y poco claras. El Product Owner dice que “no hubo tiempo para refinarlas” y que igual deben planificar. Los desarrolladores están confundidos y no pueden comprometerse de forma realista.

¿Cuál es la intervención correcta del Scrum Master?

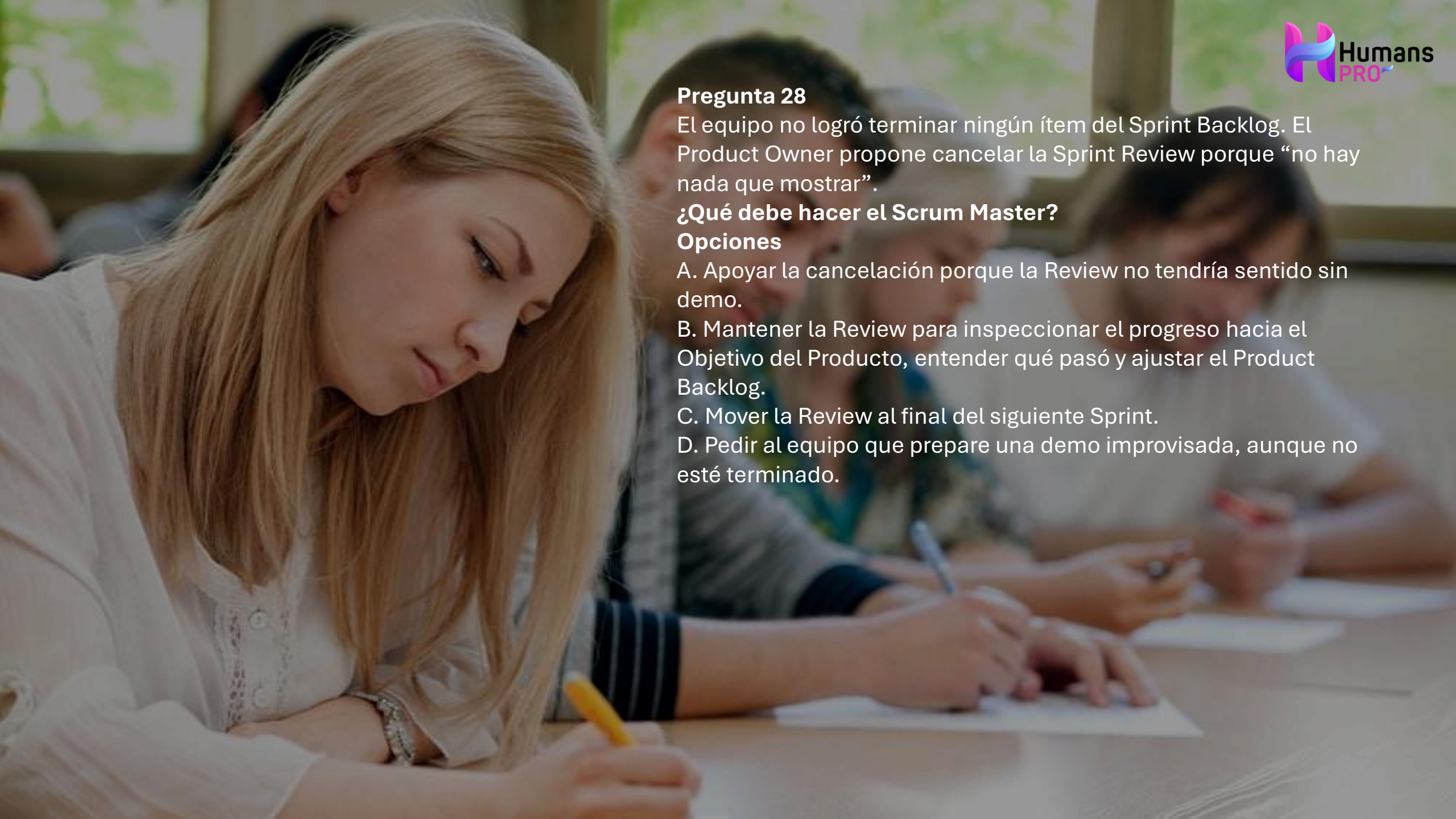
Opciones

A. Permitir que el equipo planifique con lo que tiene para no retrasar el Sprint.

B. Facilitar que se realice un refinamiento rápido dentro de la Planning, y proponer un acuerdo para mantener refinamiento continuo en futuros Sprints.

C. Cancelar el Sprint Planning inmediatamente.

D. Pedir que el Product Owner refine todo solo antes del siguiente Sprint.



Pregunta 28

El equipo no logró terminar ningún ítem del Sprint Backlog. El Product Owner propone cancelar la Sprint Review porque “no hay nada que mostrar”.

¿Qué debe hacer el Scrum Master?

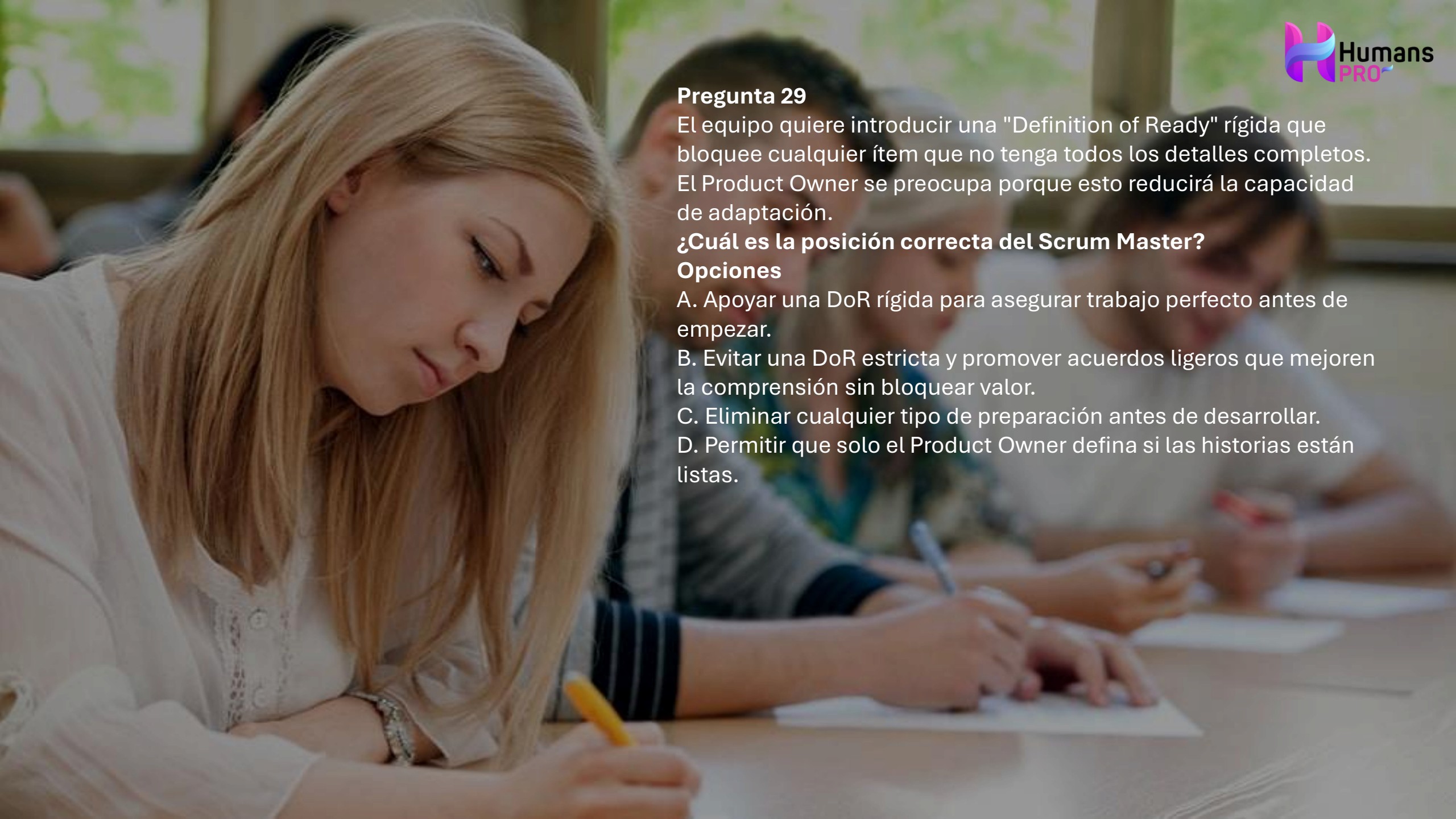
Opciones

A. Apoyar la cancelación porque la Review no tendría sentido sin demo.

B. Mantener la Review para inspeccionar el progreso hacia el Objetivo del Producto, entender qué pasó y ajustar el Product Backlog.

C. Mover la Review al final del siguiente Sprint.

D. Pedir al equipo que prepare una demo improvisada, aunque no esté terminado.



Pregunta 29

El equipo quiere introducir una "Definition of Ready" rígida que bloquee cualquier ítem que no tenga todos los detalles completos. El Product Owner se preocupa porque esto reducirá la capacidad de adaptación.

¿Cuál es la posición correcta del Scrum Master?

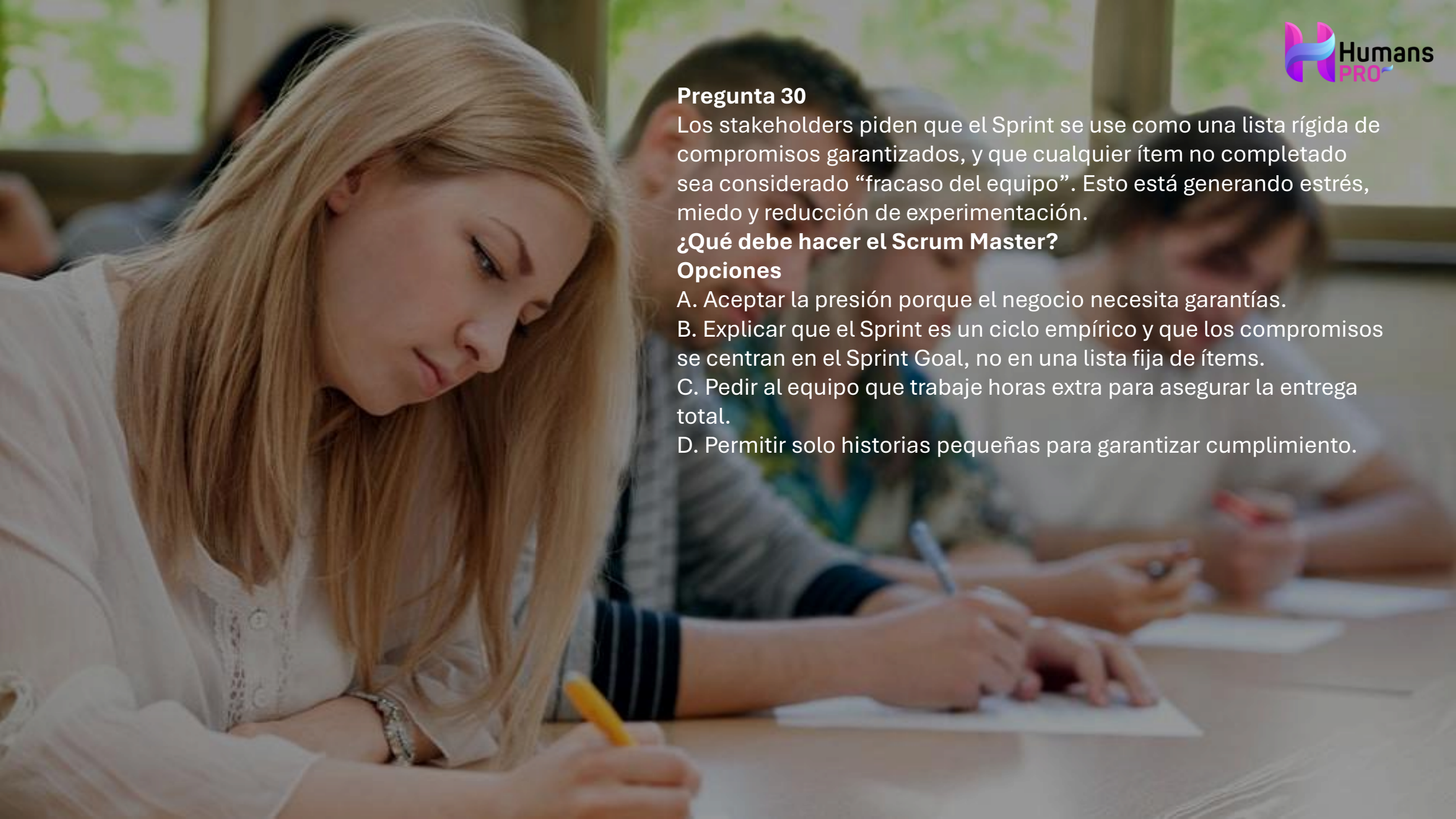
Opciones

A. Apoyar una DoR rígida para asegurar trabajo perfecto antes de empezar.

B. Evitar una DoR estricta y promover acuerdos ligeros que mejoren la comprensión sin bloquear valor.

C. Eliminar cualquier tipo de preparación antes de desarrollar.

D. Permitir que solo el Product Owner defina si las historias están listas.



Pregunta 30

Los stakeholders piden que el Sprint se use como una lista rígida de compromisos garantizados, y que cualquier ítem no completado sea considerado “fracaso del equipo”. Esto está generando estrés, miedo y reducción de experimentación.

¿Qué debe hacer el Scrum Master?

Opciones

A. Aceptar la presión porque el negocio necesita garantías.

B. Explicar que el Sprint es un ciclo empírico y que los compromisos se centran en el Sprint Goal, no en una lista fija de ítems.

C. Pedir al equipo que trabaje horas extra para asegurar la entrega total.

D. Permitir solo historias pequeñas para garantizar cumplimiento.

Humans PRO

Ente de acreditación internacional

SM PROFESIONAL



Muchas gracias

DIPLOMADO

Formación Profesional en Proyectos



UNIVERSIDAD DE SAN BUENAVENTURA

THINK DESIGN BUILD ING | FACULTAD DE INGENIERÍA

Humans PRO

