

Agile en trabajos de grado de Ingeniería de Sistemas



Ing. Ricardo D. Camargo L.
Facultad de Ingeniería
Programa de Ingeniería de Sistemas
2025

Por una cultura de vida, su calidad y su sentido

Conceptos de Agile





Pregunta #1

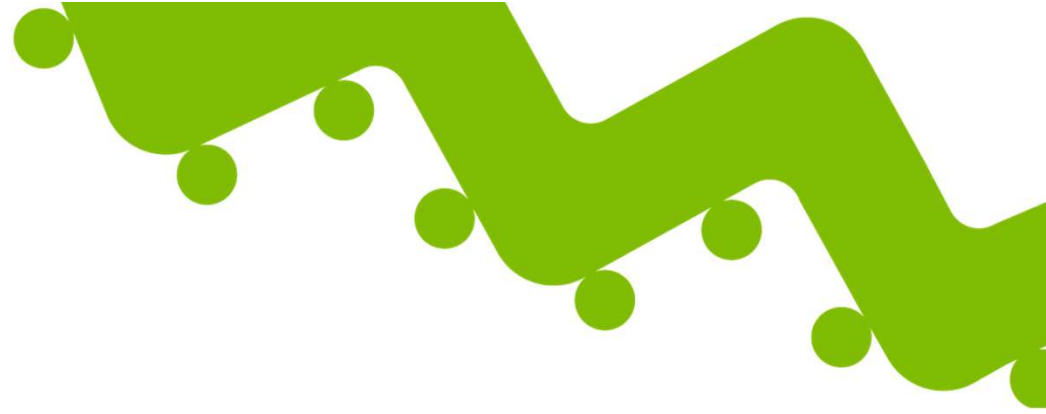


Concepto agile en proyectos de software.

¿Qué es agile?, ¿Significa que puedo
realizar un proyecto de software más
“rapidito”?



Algo de semántica...



Ágil (Del lat. *agilis*. R.A.E):

- “1. adj. Que se mueve con soltura y rapidez. Estuvo muy ágil y esquivó el golpe.*
- 2. adj. Dicho de un movimiento: Hábil y rápido. Camina con paso ágil.*
- 3. adj. Que actúa o se desarrolla con rapidez o prontitud. Tiene una prosa ágil”*

Fuente: <https://dle.rae.es/>



Algo de semántica (cont)...

Ágil (*Agile*)

“Forma de pensamiento y acción basada en principios cuyo foco es el incremento del valor para el cliente y la calidad.”

Fuente: Estay-Niculcar, C (2021). Infografía sobre Agile y todo lo que hoy en día incluye, abarca y le constituye. Recuperado de . <https://www.researchgate.net/publication/355490589>





Algo de semántica (cont)...

Ágilidad (*be agile*)

“Capacidad de la organización para responder, adaptarse y crear cambios para entregar resultados en entornos inciertos, capitalizando oportunidades y esquivando amenazas, y ahora, ante entornos disruptivos de manera productiva y adecuada”.

Fuente: Estay-Niculcar, C (2021). Infografía sobre Agile y todo lo que hoy en día incluye, abarca y le constituye. Recuperado de . <https://www.researchgate.net/publication/355490589>





Algo de semántica (cont)...

Ágilismo (to-do agile)

“Enfoque que agrupa perspectivas y herramientas (Scrum, Kanban, XP, etc.) que permiten una planificación flexible para reaccionar hábilmente a cambios, con objetivos pequeños y de corto plazo, revaluando el producto, y replanificar sobre la marcha, y/o verificar si el camino y el fin son los deseados, sin perder una visión de largo plazo”.

Fuente: Estay-Niculcar, C (2021). Infografía sobre Agile y todo lo que hoy en día incluye, abarca y le constituye. Recuperado de . <https://www.researchgate.net/publication/355490589>



Ciclos de Vida y enfoques de Desarrollo





Pregunta #2



Ciclos de vida de desarrollo de proyectos de software

¿por qué es necesario entender los
modelos y ciclos de vida para un proyecto
de software?





Ciclos de Vida de Proyectos...

Tipos		Características
Predictivo		<ul style="list-style-type: none">• Requisitos fijos• Actividades realizadas una vez por proyecto• Una sola entrega• Objetivo: Gestionar costos
Adaptativo	Iterativo	<ul style="list-style-type: none">• Requisitos dinámicos• Las actividades se repiten hasta que se corrigen• Una sola entrega• Objetivo: Solución correcta
	Incremental	<ul style="list-style-type: none">• Requisitos dinámicos• Actividades realizadas una vez por incremento• Entregas pequeñas frecuentes• Objetivo: Velocidad (en entrega)
	Ágil	<ul style="list-style-type: none">• Requisitos dinámicos• Combina la repetición iterativa de actividades con entregas incrementales• Objetivo: Valor del cliente
Híbrido		<ul style="list-style-type: none">• Incluye componentes adaptativos y predictivos• Marcos de tiempo más cortos e iterativos• Alta participación de los interesados• Requisitos más detallados

Fuente: Project Management Institute, Inc (2017). Guía del PMBOK®, 6a edición, pág. 19.



Pregunta #3



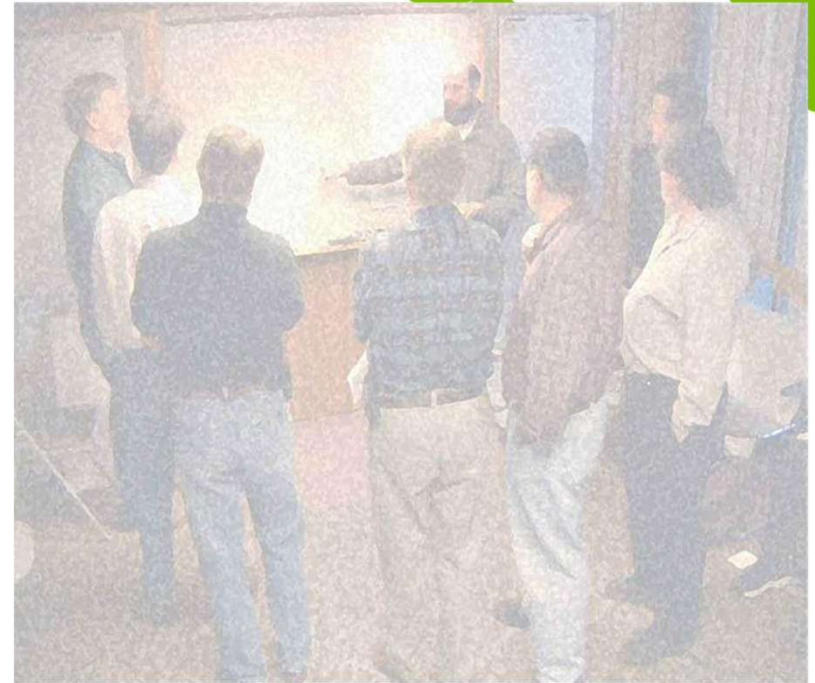
Enfoque de desarrollo Ágil

¿Qué es el enfoque ágil para desarrollar proyectos de software?, ¿Qué criterios debo tener en cuenta?



Manifiesto Ágil

- Surge el 17 de febrero del 2001: reunión de diecisiete críticos del desarrollo de software.
- Acuñaron el término Ágil para definir a las practicas que surgían como alternativa a las metodologías formales.
- “*Estamos descubriendo mejores formas de desarrollar software (..) “*
- Se conforma de 12 principios y 4 valores.



REF: <http://agilemanifesto.org/iso/es/manifesto.html>



Principios del Manifiesto Ágil

Satisfacer al usuario  Nuestra mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor.	Aceptamos los cambios  Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos Ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente.	Entregas frecuentes  Entregamos software funcional frecuentemente, entre dos semanas y dos meses, con preferencia al periodo de tiempo más corto posible.	Negocio e IT juntos  Los responsables de negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto.
Personas motivadas  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.	Comunicación cara a cara  El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.	Producto funcionando  El software funcionando es la medida principal de progreso.	Ritmo sostenible  Los procesos Ágiles promueven el desarrollo sostenible. Los promotores, desarrolladores y usuarios debemos ser capaces de mantener un ritmo constante de forma indefinida.
Excelencia técnica  La atención continua a la excelencia técnica y al buen diseño mejora la Agilidad.	Sencillez  La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.	Arquitectura y diseño  Las mejores arquitecturas, requisitos y diseños emergen de equipos auto-organizados.	El equipo reflexiona  A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para a continuación ajustar y perfeccionar su comportamiento en consecuencia.

<https://www.gjaviermartinc.com/12-principios-del-manifiesto-agil-infografia-infographic-metodologiasagiles/>



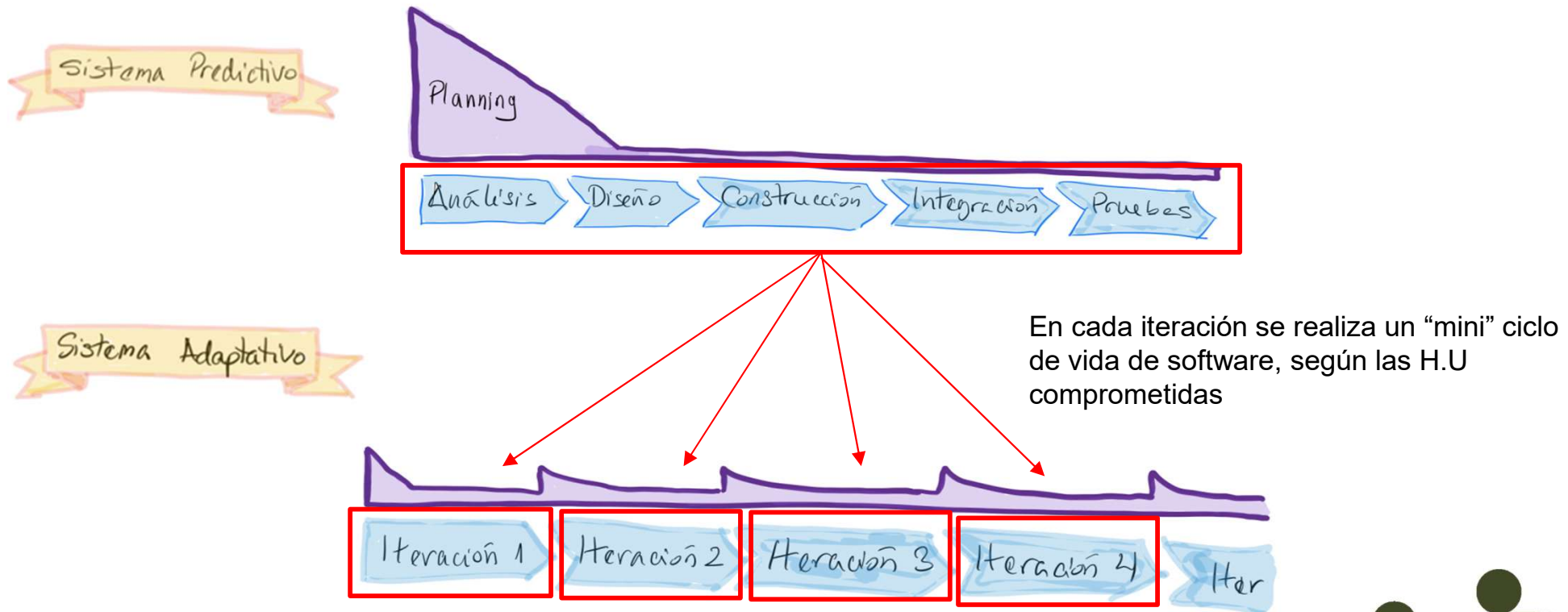
Valores del Manifiesto Agile

Individuos e interacciones sobre procesos y herramientas.
Software funcionando sobre documentación extensiva.
Colaboración con el cliente sobre negociación contractual.
Respuesta ante el cambio sobre seguir un plan.

Esto es, aunque valoramos los elementos de la derecha,
valoramos más los de la izquierda.



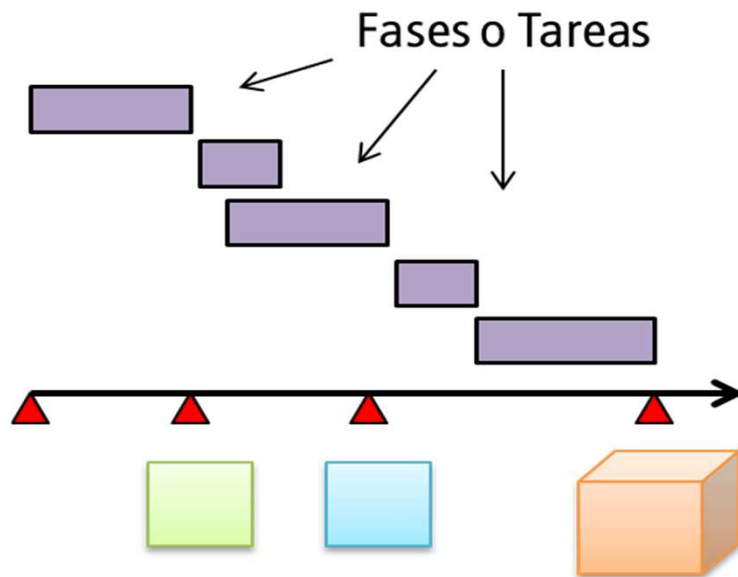
Proyectos predictivos Vs. Adaptativos



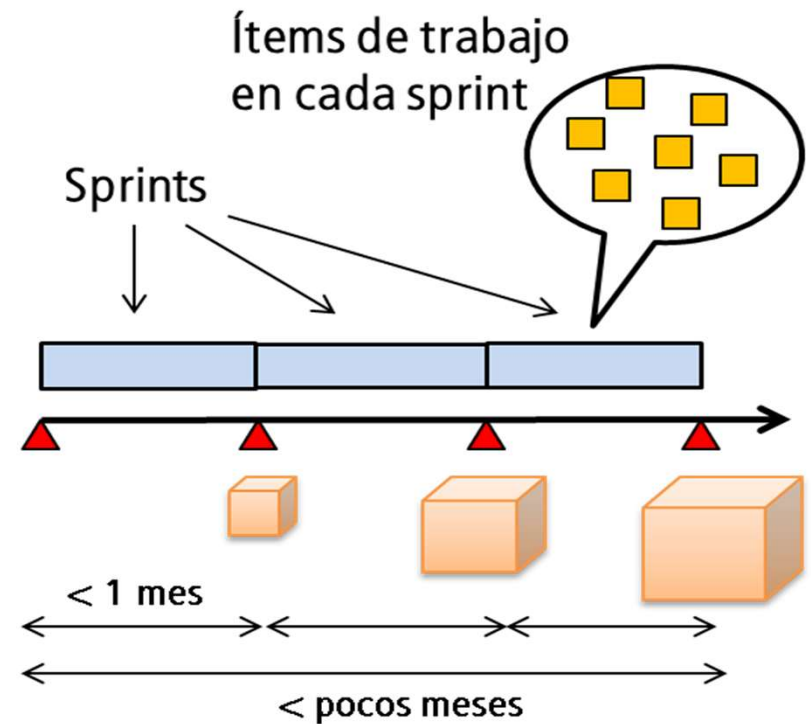


Construcción de Software Tradicional Vs. Ágil

Plan Tradicional



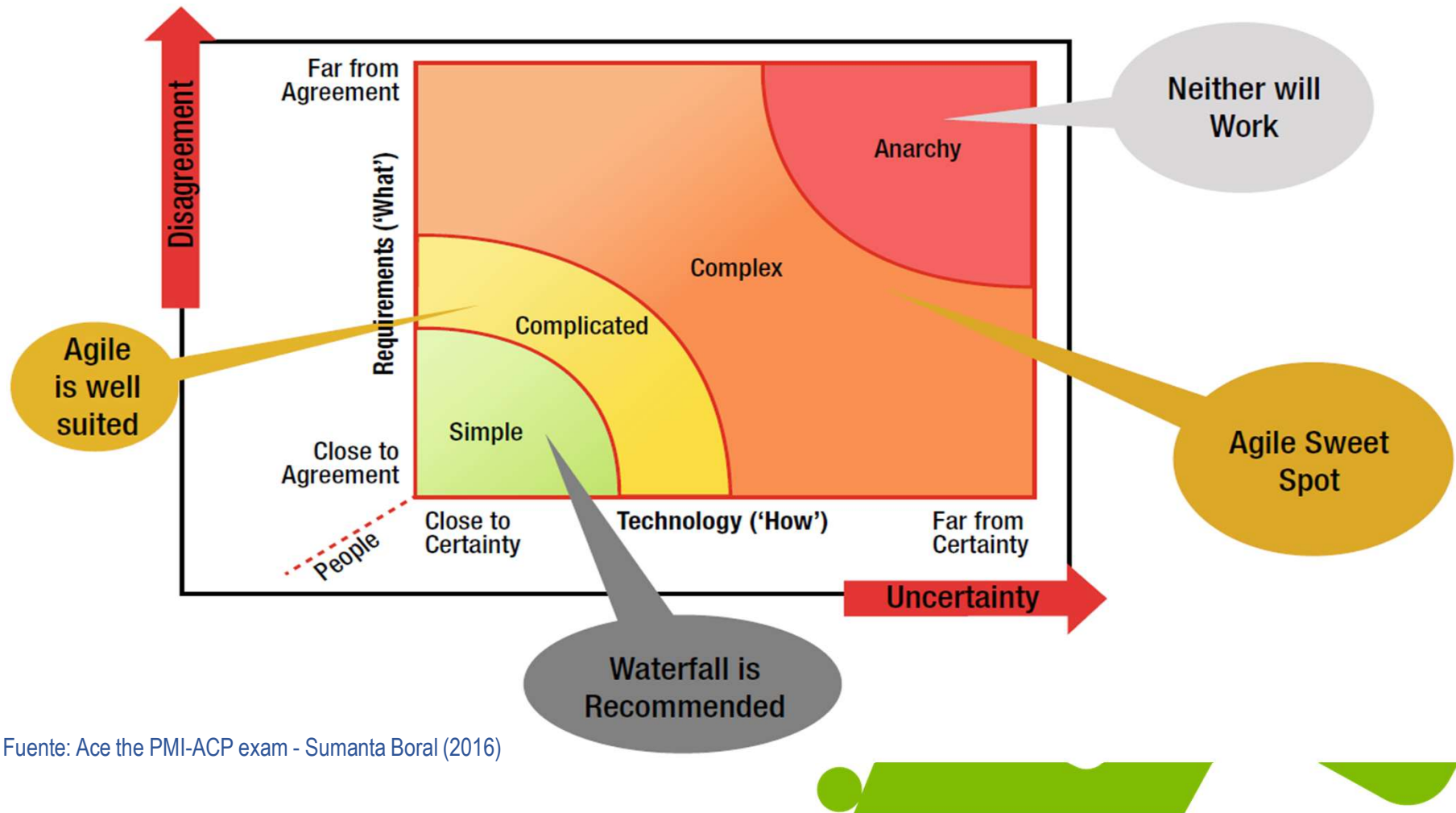
Plan Ágil



Fuente: <http://agilismoatwork.blogspot.com/2014/03/7-diferencias-entre-una-planificacion.html>



Aplicación de Enfoques



Fuente: Ace the PMI-ACP exam - Sumanta Boral (2016)

Agile y el Desarrollo de Software en TdeG.





Pregunta #4

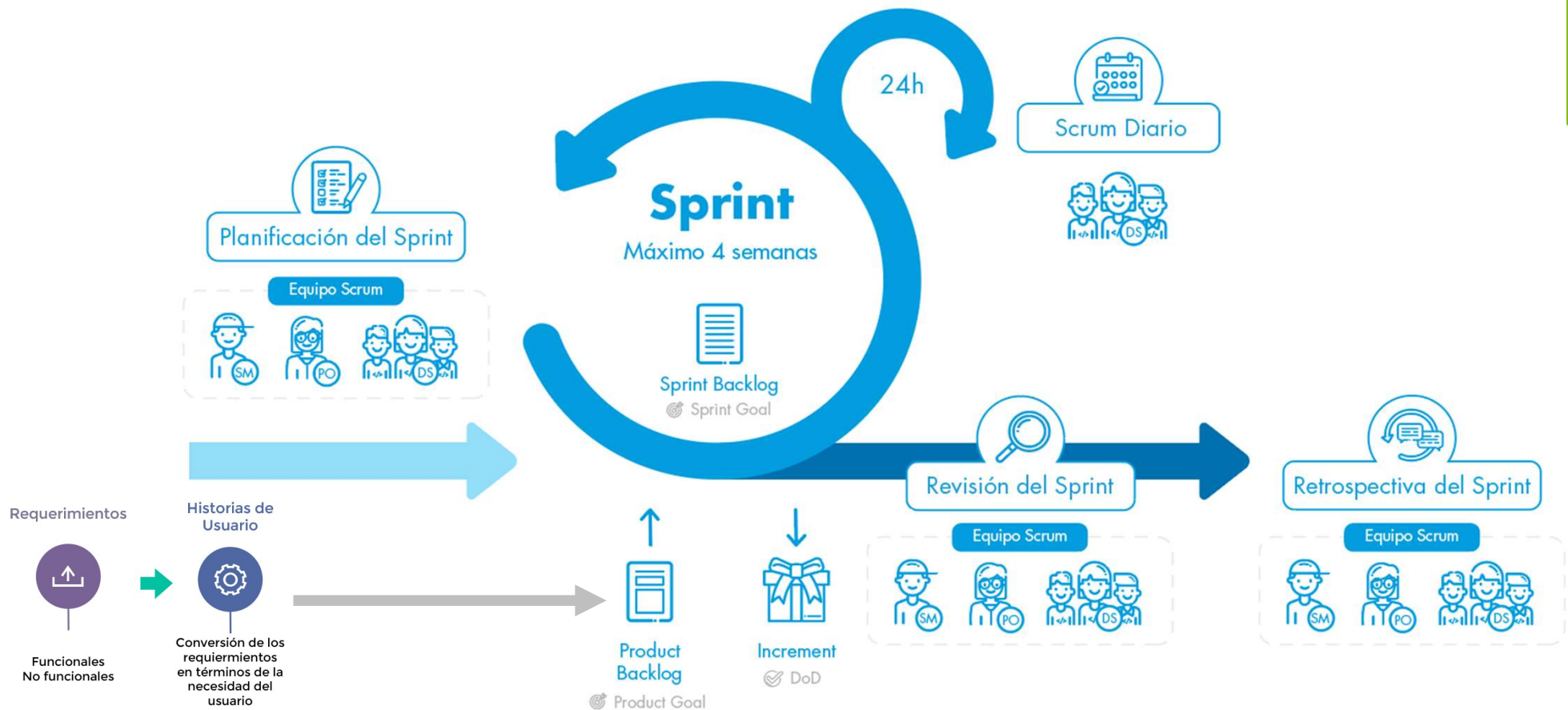


Aplicación de agile en Trabajos de Grado

¿Qué elementos mínimos debo evaluar sobre las metodologías / técnicas agile utilizadas en un TdeG?, ¿Cómo deberían los estudiantes demostrar idoneidad en su aplicación?



Scrum: Esquema General de Trabajo





Extreme Programming (XP)

- Extreme Programming (XP) es una técnica ágil utilizada en el desarrollo de software.
- Se centra en la cooperación constante entre desarrolladores y clientes.
- Fomenta la integración continua y retroalimentación frecuente.
- Emplea métodos como la programación en parejas y la planificación en iteraciones (sprints).
- Tiene como objetivo mejorar la calidad del software y adaptarse rápidamente a los cambios.





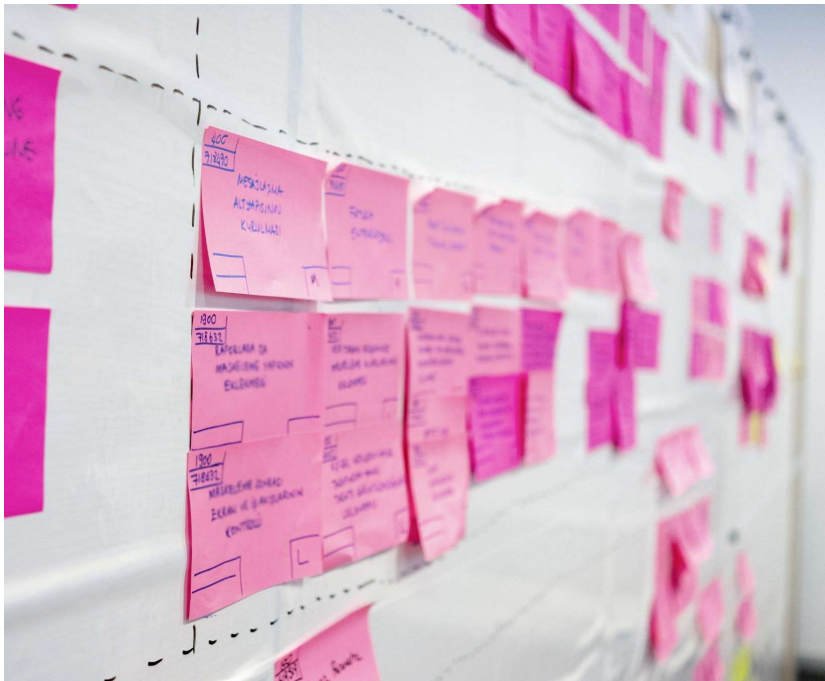
Feature Driven Development (FDD)

- 1. FDD (*Feature Driven Development*) es un enfoque ágil para desarrollo de software.
- 2. Se centra en funciones y características del cliente como base para la planificación.
- 3. Etapas de FDD:
 - Desarrollo de un modelo general
 - Construcción de una lista de características
 - Diseño y construcción por características.
- 4. FDD promueve la colaboración entre equipos multidisciplinarios.

Característica	Scrum	FDD
Ciclo Iterativo	Sprint	Iteración
Unidad de trabajo	Historia de Usuario	Función
Planificación	Sprint Planning	Planificación por Función
Backlog	Product Backlog Sprint Backlog	Lista de Funciones
Roles	Product Owner Scrum Master Development Team	Jefe de Proyecto Arquitecto Desarrollador Jefe
Artefactos	Product Backlog Sprint Backlog Burndown Chart	Modelo de Dominio Lista de Funciones Diseño Detallado
Reuniones	Daily StandUp Meeting Sprint Planning Meeting Sprint Review Retrospectiva	Inspección de Diseño Demostración de la Función
Enfoque	Gestión del trabajo Auto-organización	Entrega de funcionalidades Diseño detallado



Kanban



- Kanban (tarjeta visual - japonés) es una técnica basada en agile para gestionar el progreso del trabajo.
- Se centra en la visualización del flujo de trabajo y la limitación de tareas en curso.
- Utiliza un tablero para mostrar el estado de las tareas y su progreso.
- Se utiliza como técnica para visualización del flujo de trabajo y mejora continua dentro de una iteración.

Fuentes: Gemini, Copilot



Elementos de Scrum / XP aplicables al Trabajo de Grado

Planeación Agile

- Justificación (o caso) del Negocio
- Declaración de la Visión del Proyecto
- Definición de Roles (PO, SM, DT)
- Matriz de requerimientos
- Épicas / Historias de Usuario
- Product Backlog priorizado

Planeación del Sprint (1..n)

- Definición de Mínimo Producto Viable
- Sprint Goal
- Definition of Done
- Estimación de esfuerzo y creación de tareas para H.U.
- Sprint Backlog

Ejecución del Sprint (1..n)

- Daily Standup Meeting
- Seguimiento y control del sprint (burndown chart, **Tablero Kanban**, etc.).
- Programación por parejas (XP)
- Integración continua (XP)

Revisión del Sprint (1..n)

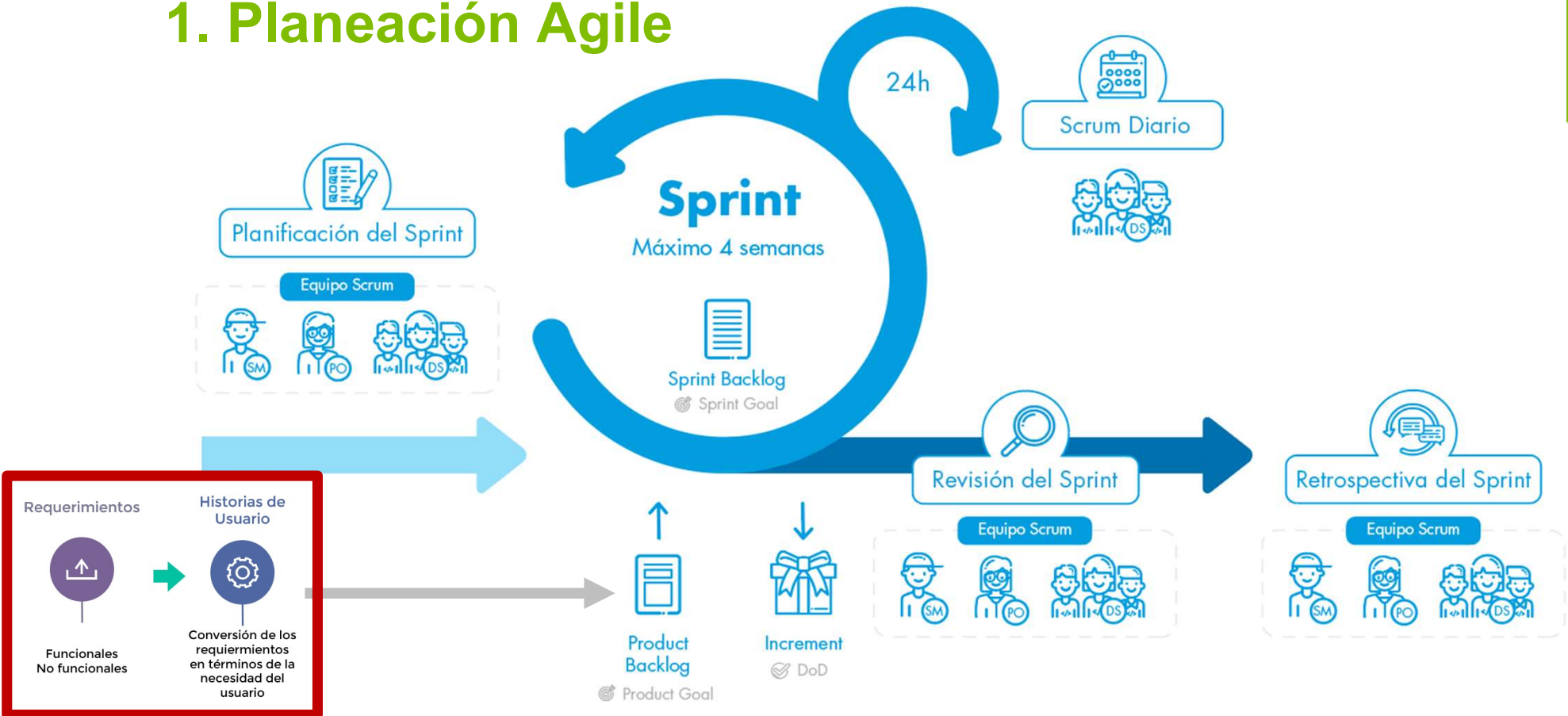
- Reunión de revisión del sprint.
- Retrospectiva





Esquema General de Trabajo

1. Planeación Agile





Propuesta de Elementos Mínimos de Ágil / Scrum en el TdeG (1/4)

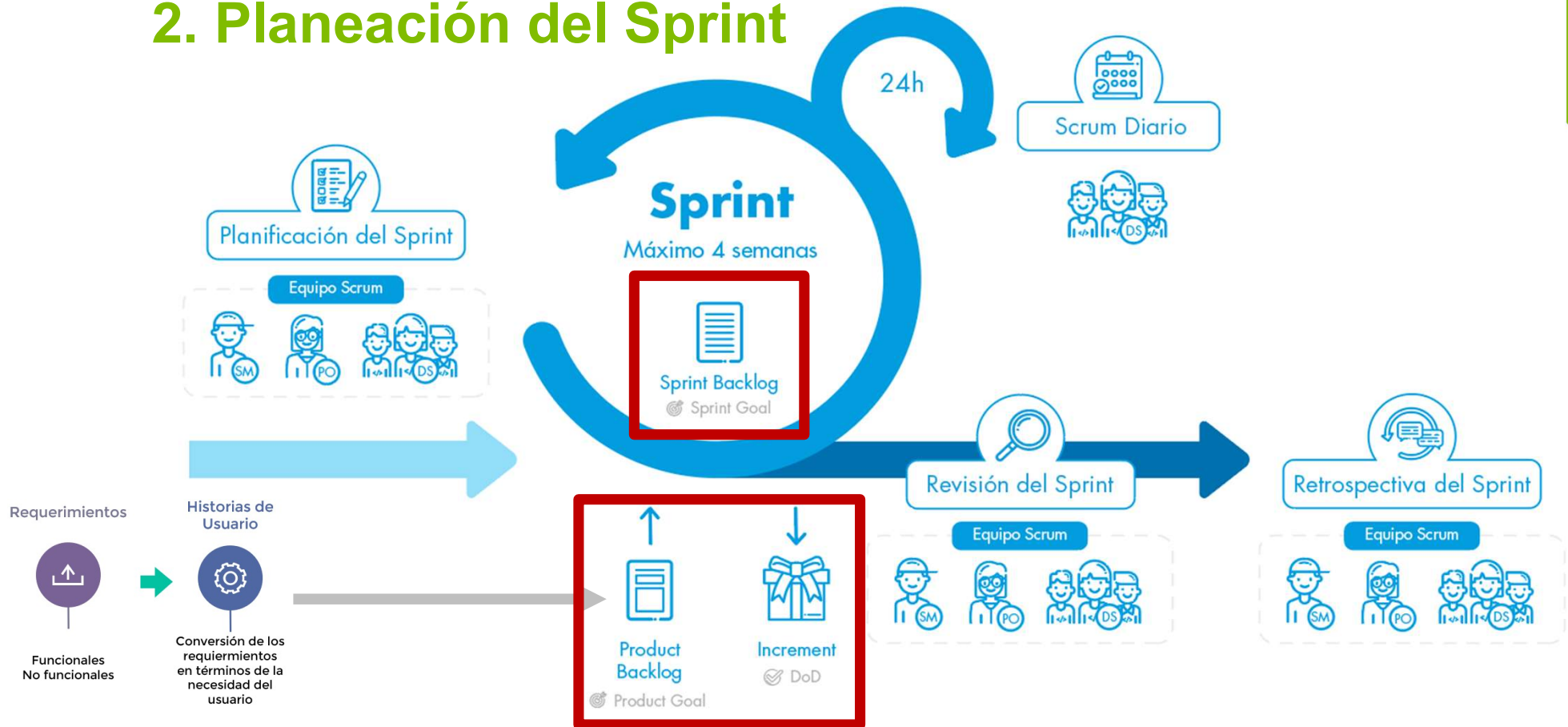
Sección del Documento de TdeG	Elemento	Tipo de Elemento
Descripción Contexto y Justificación del Problema Desde el Modelo Biopsicosocial y Cultural	Justificación o caso de negocio	General Scrum





Esquema General de Trabajo

2. Planeación del Sprint





Propuesta de Elementos Mínimos de Ágil / Scrum en el TdeG (2/4)

Sección del Documento de TdeG	Elemento	Tipo de Elemento
Diseño Metodológico	Justificación de la elección de la metodología, ágil, y/o híbrida	General
	Definición de Roles (PO, SM, DT)	Scrum
	Matriz de Requerimientos	General
	Diagramas de Casos de Uso	General
	Épicas / Historias de Usuario	Scrum
	Product Backlog Priorizado	Scrum
	Diagramas de Diseño (Arquitectura, BD, procesos, etc.)	General
	Cronograma con la descripción de los Sprints a realizar (*)	General
	Estructura de Desglose de Trabajo (EDT) con la descripción de los Sprints a realizar	General

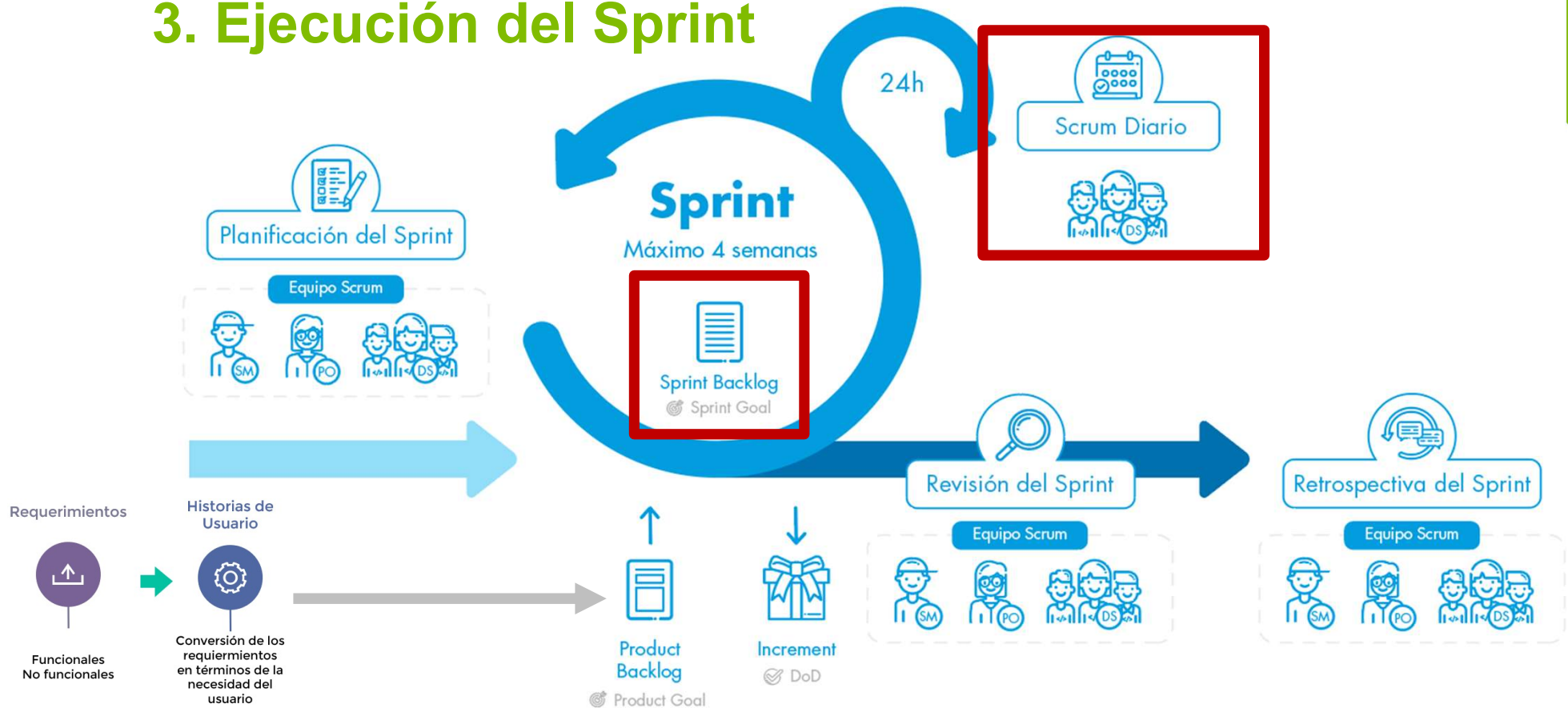
PROPUESTA DE DOCUMENTACION

1. Diagrama de casos de uso
2. Diagrama de procesos (BPM)
3. Diagrama de arquitectura (clases, capas, módulos, etc.)
4. Diagrama de secuencia (OOP)
5. Mockups / pantallazos / UI
6. Modelo Relacional



Esquema General de Trabajo

3. Ejecución del Sprint





Propuesta de Elementos Mínimos de Ágil / Scrum en el TdeG (3/4)

Sección del Documento de TdeG	Elemento	Tipo de Elemento
Resultados y Discusión (Para cada sprint dentro del documento)	Definición de Producto Mínimo Viable	Scrum
	Sprint Goal, y Definition of Done	Scrum
	Selección de H.U. para el Sprint Backlog	Scrum
	Estimación de esfuerzo y creación de tareas para cada H.U.	Scrum
	Evidencia de programación por pares	XP
	Evidencia de integración del código en repositorio	XP
	Evidencia del seguimiento (Burndown chart, Kanban)	Scrum
	Actualización de Cronograma y documentación del TdeG	General
	Proceso de construcción y resultados del software (arquitectura, Mockups, etc.)	General
	Pruebas de Software	General
PROPUESTA DE PRUEBAS 1. Pruebas de caja blanca (unitarias) 2. Pruebas de Integración (módulos del presente sprint y anteriores) 3. Pruebas de carga y stress	Evidencia de los Resultados de la revisión de cada sprint	Scrum



Esquema General de Trabajo

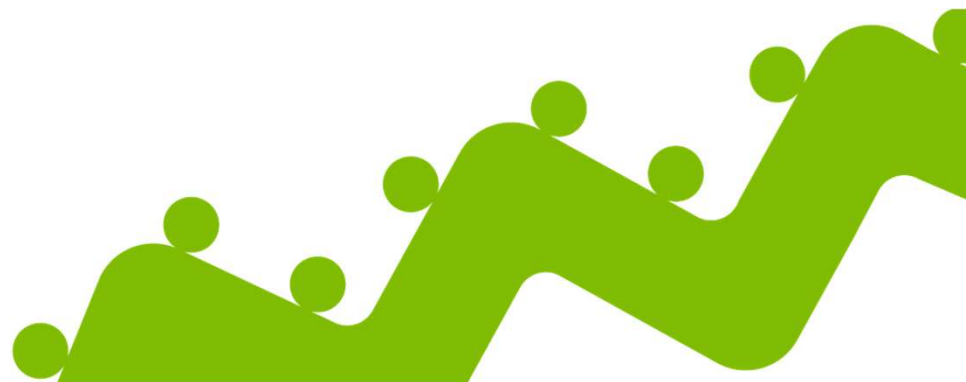
4. Revisión del Sprint





Propuesta de Elementos Mínimos de Ágil / Scrum en el TdeG (4/4)

Sección del Documento de TdeG	Elemento	Tipo de Elemento
Lecciones Aprendidas y Trabajo Futuro (por cada sprint)	Evidencias de la reunión de retrospectiva	Scrum
	Elementos de cada retrospectiva	Scrum
	Síntesis de todas las retrospectivas (lecciones aprendidas)	General





Gracias





Perguntas



Dudas e
Inquietudes