IIC 3143
PLANIFICACIÓN EN DETALLE
Levantamiento, Análisis, Calendario y Estimación

Rodrigo Sandoval
Profesor Adjunto Asociado
DCC - Escuela de Ingeniería
rsandova@ing.puc.cl

Preguntas Frecuentes (Externas)

¿Cómo vas con el proyecto?

¿Cuánto te demoras en hacer _____?

¿Puedes hacer una demo hoy en la tarde?

¿Podrías incluir rápidamente _____?

¿Por qué no hacemos el siguiente cambio?

Preguntas Frecuentes (Internas)

¿Y ahora qué me toca hacer?

¿Cuánto me demoraré en hacer _____?

¿Cuál fue la última versión a la que hice *commit*?

¿Qué pasa si re-diseño toda esta parte ahora?

¿Por qué?

Preguntas externas siempre van a existir … pero se pueden anticipar y así, evitar.

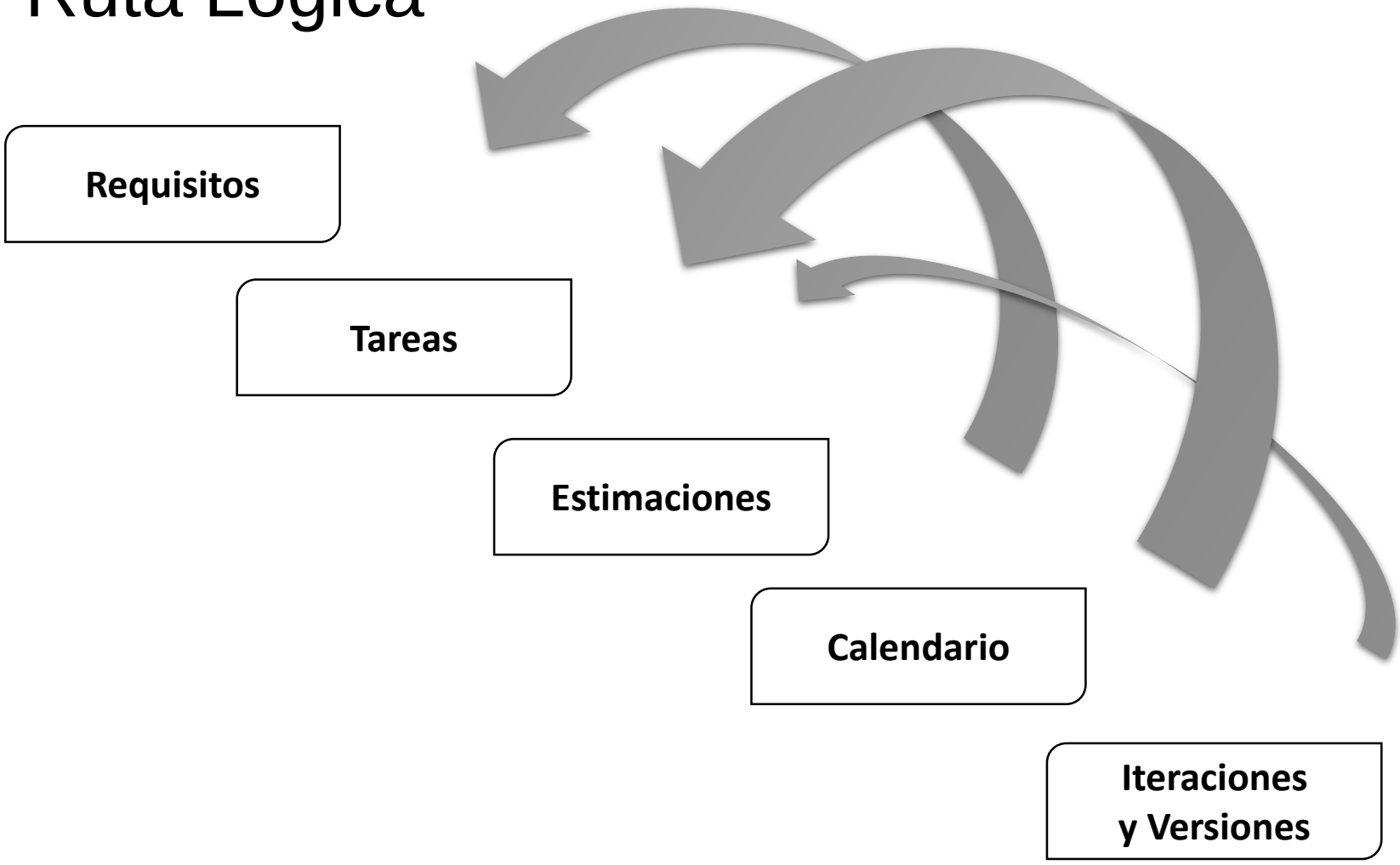
Las preguntas internas pueden también resolverse en forma automática.

¿Cómo?

Organizando y Planificando el Desarrollo

… de la forma más **simple** posible.

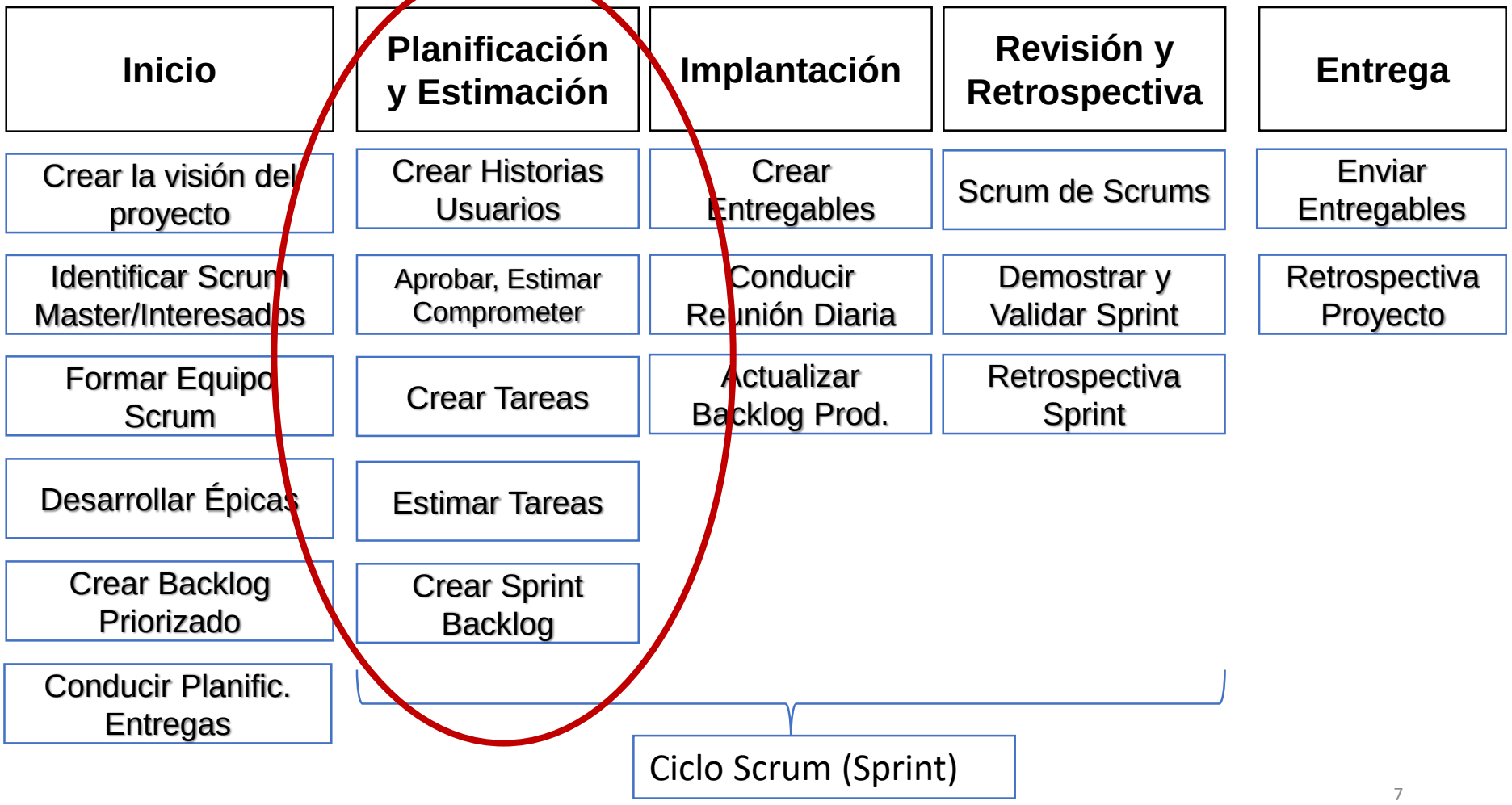
Ruta Lógica


Requisitos
Tareas
Estimaciones
Calendario
Iteraciones
y Versiones

¿Cómo se Planifica en Scrum?

Procesos de Scrum en 5 Fases

Los procesos direccionan las actividades específicas y el flujo de un proyecto Scrum



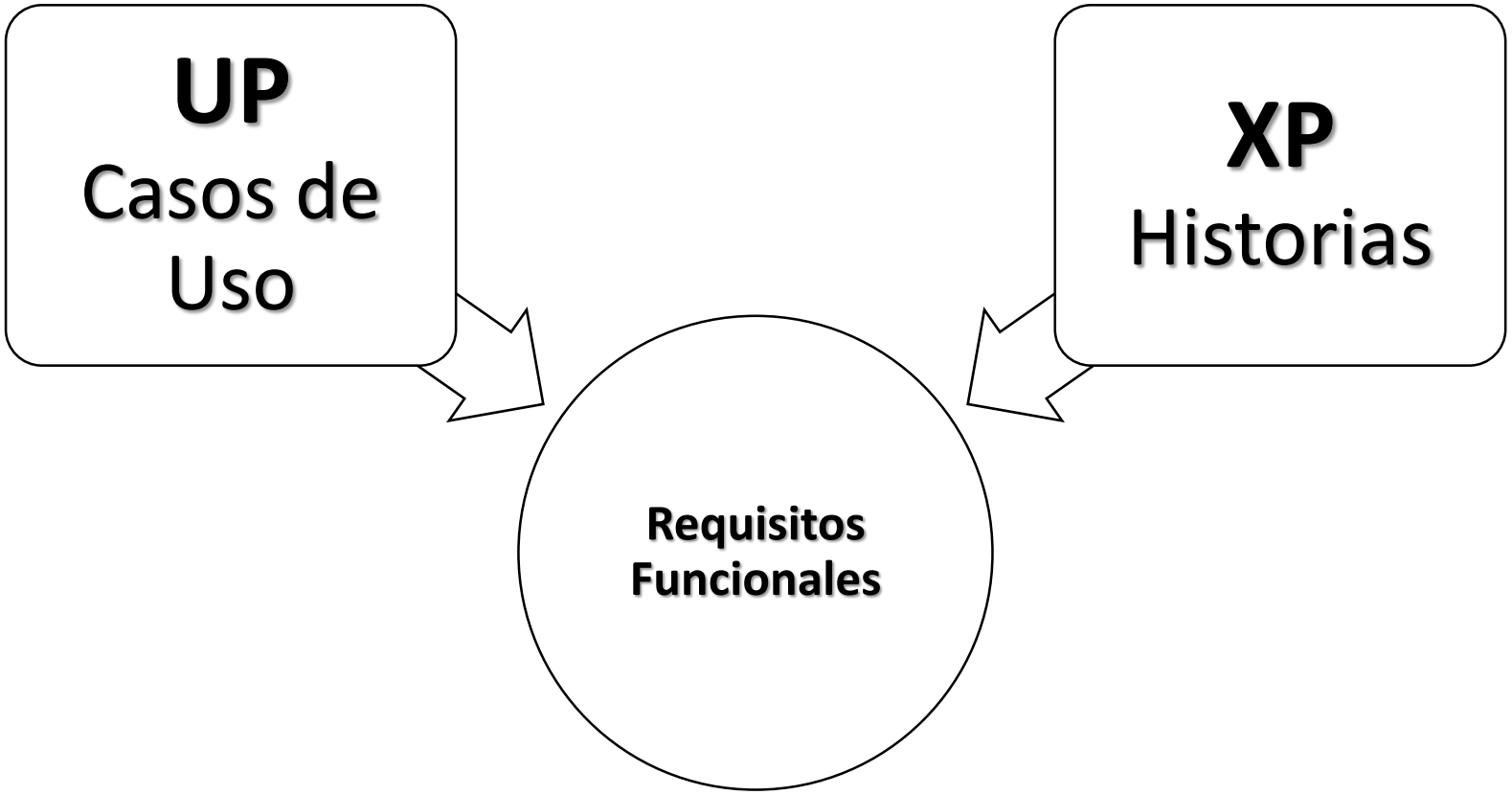
Capturar y documentar reqs.

¿Por dónde partir?

Capturando los Reqs.

- Siempre, *siempre*, *siempre* se busca contestar la pregunta:
  ¿cuál es el problema?
- … y no ¿qué tenemos que hacer?
- … y recordar que el cliente no siempre tiene la idea precisa de lo que se requiere como solución.
- Eso es trabajo de los analistas.

¿Cómo se documentan los requisitos?



Se puede elegir cualquiera, dependiendo del contexto

Caso de Uso - Ejemplo

- **Inscripción de Usuario (H2.1)**
- **Descripción**:
  Un nuevo usuario se inscribe en el sistema siguiendo la opción desde la página principal, para luego llenar un formulario en el cual se le solicitan varios datos personales, algunos obligatorios como e-mail, nombre completo, dirección postal, teléfono, y otros opcionales como RUT o N° Pasaporte, género, fecha de nacimiento. Se le solicita ingresar una contraseña (dos veces para asegurar que la escribió bien y se valida la correctitud del e-mail.
  La Meta es lograr un registro de inscripción de nuevo usuario y darle una credencial válida para utilizar el sistema.

- **Actores**: Usuario (nuevo)
- **Stakeholders** relevantes del caso:
  - Gerente Operaciones (“temas seguridad y almacenamiento de cuentas”)
  - Gerente Comercial (“facilidades de registro”).
- **Precondiciones**: (ninguna).
- **Triggers**: al terminar correctamente se deja un registro en el *log*.

Historia de XP

(Título descriptivo de la historia)

(Nº ID)

(Descripción breve y concisa de la historia en dos o tres párrafos describiendo propósito, entrada, salida y efectos principales)

CLIENTE

DESARROLLADOR

(Estimación Esfuerzo)

Historia y Tareas Desarrollo – Ejemplo 2

Inscripción de Nuevo Usuario

H 04

El usuario interesado accede a un formulario donde se le piden sus datos personales, como nombre, apellidos, e-mail (además utilizado para validación de la inscripción), ciudad de residencia, fecha nacimiento.

Todos los datos, excepto fecha de nacimiento son obligatorios.

La fecha de nacimiento debe validarse en rango [110 años atrás, hoy]

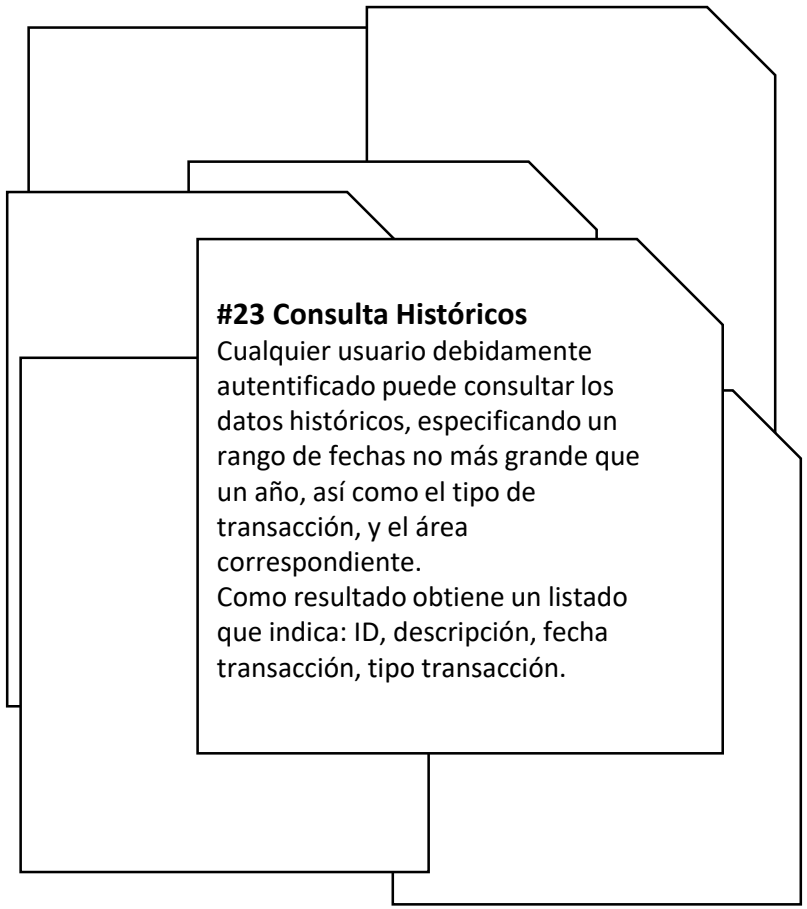
Consideración 1: validación de la inscripción es una funcionalidad separada. Se puede comenzar con una inscripción simple y directa, sin validación.

Consideración 2: La validación de obligatoriedad se puede definir como historia separada, dependiendo de su prioridad.

Consideración 3: A su vez, la validación de fecha puede ser otra historia separada

5 días

“Documento de Reqs”



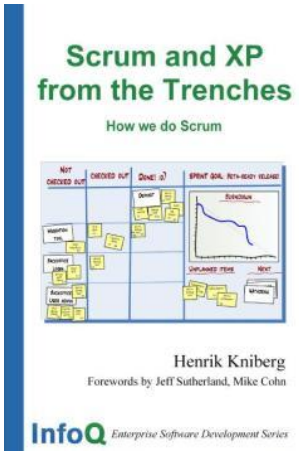
#1 – xxxxxxxxx xxxxxxx
#2 – xxxxxxxxxxxxxx xxxxxxx
#3 – xxxxx xxxxx xxxxxxxxx
#4 – xxxxxxxxx xxxxxxxxxx
#5 – xxxxx xxxxxxxxxxxxx
#6 – xxxxxxx xxxxxx xxx xxxx
…

Atributos de un Req

- ID: facilita su referencia en conversaciones técnicas y diarias.
- Nombre o título: la declaración.
- El detalle o descripción.
- Importancia: Alta, Media, Baja (u otra escala)

Adicionales (según Henrik Kniberg *)

- "How to demo"
  → una aprox. a la demo y test. func.
- Requestor. (Identificar quién pide)
- Estimación.



Product Backlog de Scrum

- El Cliente (PO) prepara y entrega esta lista de requerimientos PRIORIZADOS (en orden).
- Si no tiene conocimiento o experiencia para este nivel de detalle, el analista prepara el detalle, pero la prioridad la pone el cliente.
- A este nivel puede ser prioridad: alta/media/baja

Descripción	**Prioridad**
#3 - Llenar formulario de datos de cotización	Alta
#4 - Modificar datos de cotización	Baja
#5 - Consulta/Búsqueda de cotizaciones por cliente y por fecha	Media
. . .	

¿Cómo secuenciar actividades?

De los Reqs a un Plan Calendarizado

Planificación vs. Calendarización

Planificación y Calendarización son dos cosas diferentes.

Planificación General

- Enfoque de proyecto.
- Estructura del equipo.
- Aspectos relevantes de la arquitectura y la solución.
- Hitos relevantes.
- PLAN GENERAL
- (Primer Plan)

Calendarización

- Plan detallado: **Ordenamiento y Estimación de Actividades**.
- Asignación de tareas por persona.
- Activación de protocolo de control de cambios.

Plan Calendarizado

- Tomar la lista de Casos de Uso y descomponer en tareas técnicas o sub-CdU.
- … y/o dimensionar en forma gruesa.
- Establecer *timeboxes* (iteraciones) cortas y determinar lo que se alcanza a hacer en cada *timebox*.
- Establecer un orden – preliminar – de desarrollo de las partes → Es fundamental para tener disciplina de trabajo y evitar los clásicos: *¿En qué estaba? ¿y ahora qué hago?*

Descomposición en tareas

#12 Registro Nueva Transacción
Cualquier usuario debidamente autentificado puede ingresar una nueva solicitud, que se traduce en una transacción. Debe indicar el tipo de solicitud, utilizando la fecha actual como fecha de solicitud.

#23 Consulta Históricos
Cualquier usuario debidamente autentificado puede consultar los datos históricos, especificando un rango de fechas no más grande que un año, así como el tipo de transacción, y el área correspondiente.
Como resultado obtiene un listado que indica: ID, descripción, fecha transacción, tipo transacción.

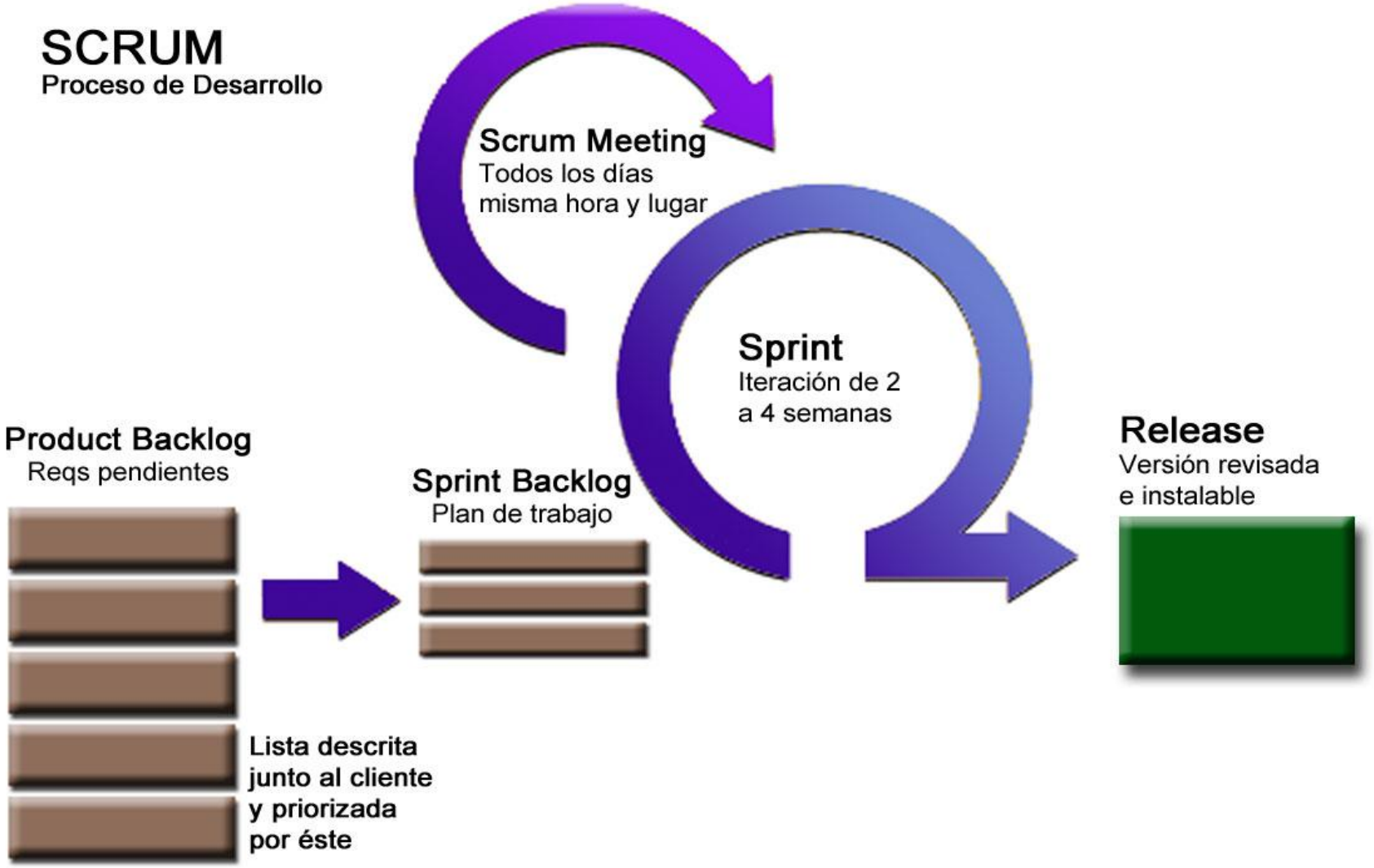
- Al descomponer en tareas, se pueden identificar posibles intersecciones con otros reqs, pero más importante, se puede estimar con mejor precisión.

Estimación “automática”

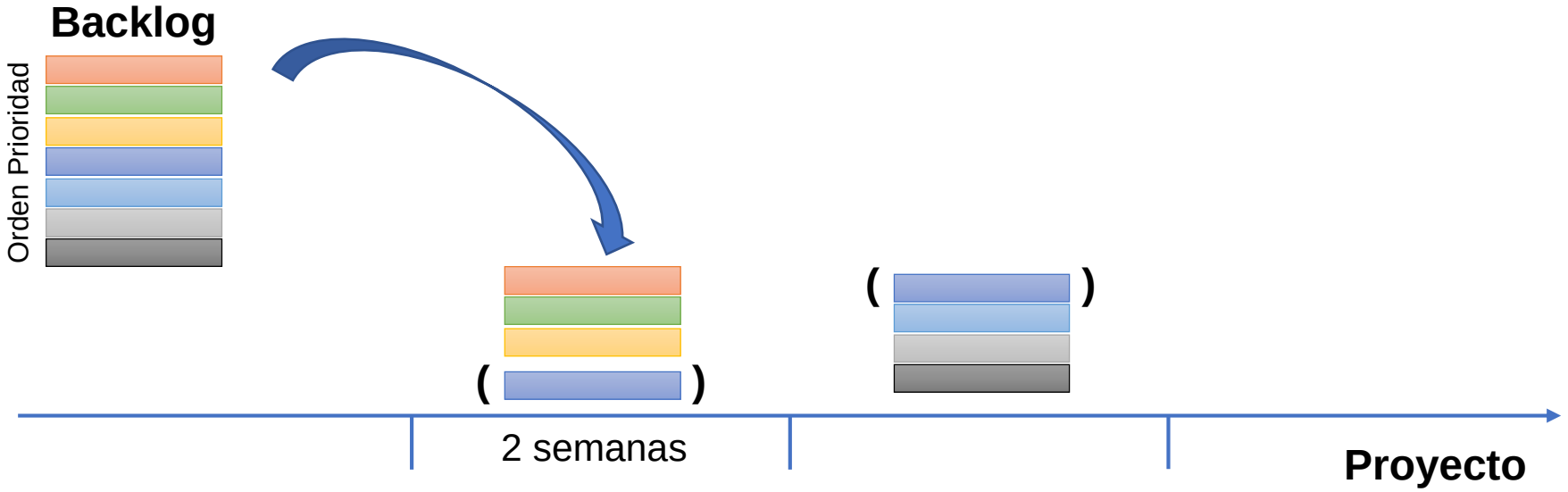
Req	Complejidad
#2 – Página de inicio	Baja
#3 – Ingreso solicitud	Alta
#4 – Consulta históricos	Media

Complejidad	Esfuerzo
Baja	1
Alta	2
Media	4

Scrum Workflow

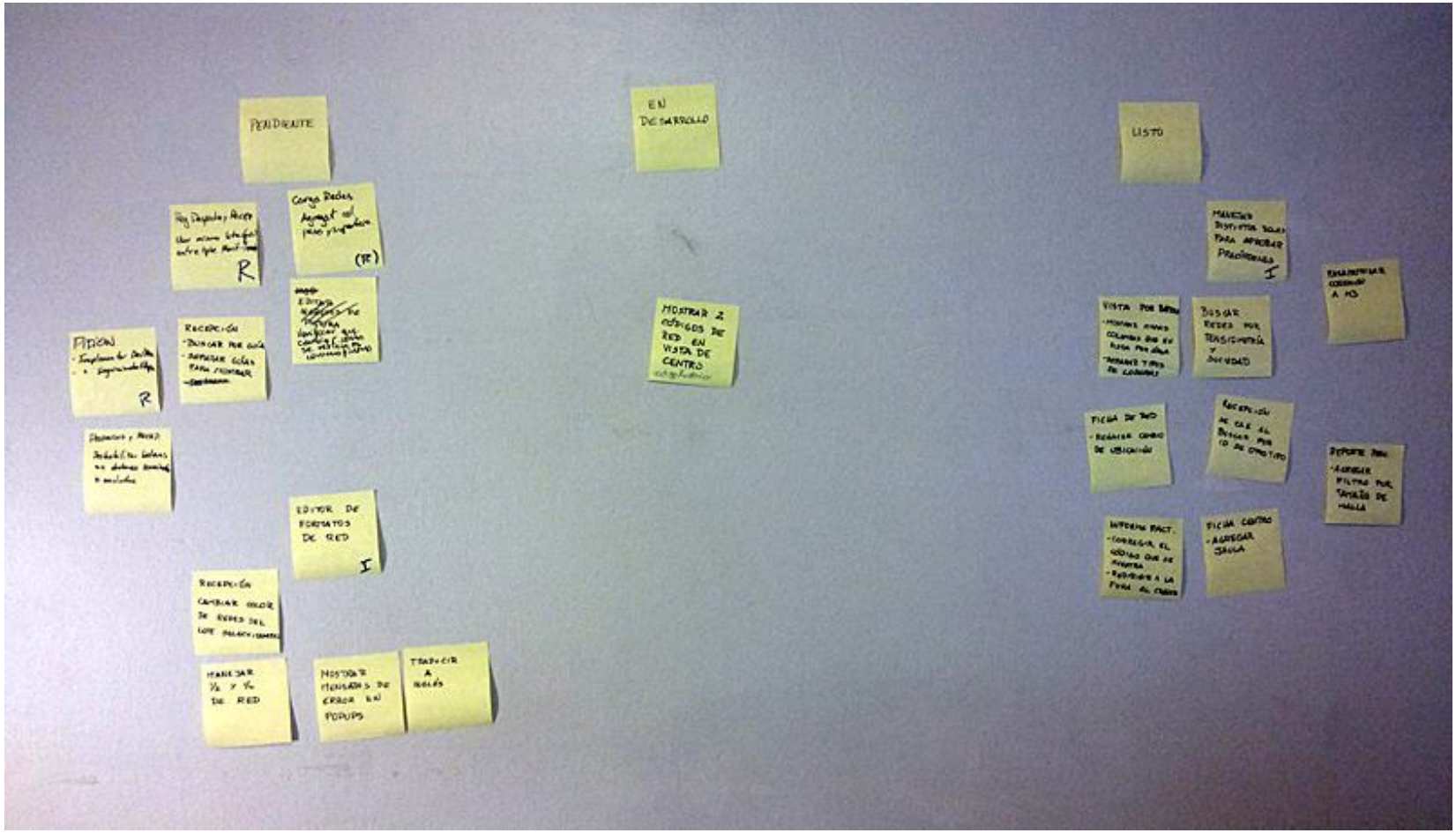


Timebox



- Preguntas más fáciles de contestar
- ¿Alcanzo a hacer A+B+C en dos semanas? → “De todas maneras”
- ¿Alcanzo además a hacer D? → “Posiblemente”
- Entonces D queda como opcional, pero programado para la sig.

Organización con Kanban



Ordenando el Calendario

- *Scrum – Sprint Plan*

Descripción	**Origen**	**Respon-sable**	**Estado**	**Iter 1**	**Iter 2**	**Iter 3**
Crear BD	Pedro	Pedro	Terminada	1	0	0
Crear Tablas Modelo 1	Pedro	Jorge	Terminada	2	0	0
Instalar servidor Web y configurar seguridad	José	José	En progreso	1	0	0
Implementar Stored Proc para Cotizaciones	Pedro	Jorge	No iniciada	4	4	0
. . .						

Asignación de tareas

- Let's plan ***Pair Programming***

Descripción	Esfuerzo	Designado
CdU #3	3	Pedro+1
CdU #4	3	Jorge+1
CdU #5	4	José+1
CdU #6	6	Jorge+1
..		

El esfuerzo se dimensiona con el esfuerzo pensado en una única persona, pero se asignan dos.

TOTAL ESFUERZO:
~ 2 x $\sum$ Esfuerzo

Realmente se asignan dos personas, pero se establece un asignado responsable principal. El "acompañante" se podrá ir definiendo en el día a día

(Otro paréntesis)

Pair-Programming

Pair Programming

- Todo código destinado a producción es creado por dos personas trabajando juntas en un mismo computador.
- PP mejora la calidad del software sin impactar la fecha de entrega.
- Es contra-intuitivo, pero dos personas trabajando juntas agregarán tanta funcionalidad como dos trabajando por separado, excepto que la calidad será mucho mayor.

Pair Programming

- La mejor manera es sentarse uno al lado del otro frente al monitor.
- Uno teclea y piensa tácticamente en el método que se está escribiendo (*driver*), mientras el otro piensa estratégicamente cómo ese método calza en la clase (*navigator*).
- Se van rotando en los roles continuamente.
- Dentro del equipo, los pares cambian continuamente.

Pair Programming

- Ventajas:
  - Mejora la calidad del software desarrollado, tanto a nivel de diseño y limpieza (*simple design & refactoring*), como de funcionalidad correcta (TDD).
  - Conocimiento compartido: respaldo si un desarrollador se ausenta.
  - Refuerzo de buenas prácticas de programación y uso de los estándares definidos por el equipo.
  - Facilita el conocimiento colectivo del código (otra práctica XP).

Pair Programming

- Referencia por excelencia: www.pairprogramming.com
- Otros:
  - www.extremeprogramming.org/rules/pair.html
  - Un ejercicio: PairDraw industriallogic.com/games/pairdraw.html