



# Ingeniería del Software II

## SIS - 312

Gustavo Palabral



# **ESTIMACION PARA PROYECTOS SOFTWARE**

# Proceso de Planificación del Proyecto

- El objetivo de la planificación del proyecto es proporcionar un marco que permita hacer estimaciones razonables sobre recursos, costo y plan.
- Las estimaciones deben intentar definir el mejor y peor caso de forma que los resultados sean limitados.
- Aunque existe un grado de incertidumbre inherente, el equipo se embarca en un plan que es consecuencia de las tareas de planificación.
- Un plan debe ser adaptado y actualizado a medida que avanza el proyecto.

# Proceso de Planificación del Proyecto

- La planificación del proyecto consiste en:
  - Dividir el trabajo en partes y asignarlos a los miembros del equipo
  - Anticiparse a los problemas que puedan surgir y preparar soluciones provisionales a los problemas.
- El plan del proyecto, que se crea en el inicio de un proyecto, se utiliza para comunicar:
  - Cómo se realizará el trabajo con el equipo y los clientes del proyecto
  - Para ayudar a evaluar los avances en el proyecto.

# Planificación del Proyecto: Checklist

1. Establecer el alcance del proyecto.
2. Determinar la factibilidad.
3. Analizar los riesgos
4. Definir los recursos requeridos
  - a. Determinar los recursos humanos requeridos.
  - b. Definir recursos de software reutilizables.
  - c. Identificar recursos de medio ambiente: Hardware y Software.
5. Estimar costos y esfuerzo
  - a. Descomponer el problema
  - b. Desarrollar 2 o mas estimaciones usando: Puntos de Función, Casos de Usos, Historias de Usuario.
  - c. Reconciliar las estimaciones
6. Desarrollar un plan del proyecto
  - a. Establecer un conjunto de tareas
  - b. Definir una red de dependencias de las tareas
  - c. Desarrollar un grafico de tiempo
  - d. Definir mecanismos de seguimiento del plan

# Complementos al Plan del Proyecto

Plan	Descripción
Plan de Calidad	Describe los procedimientos y estándares de calidad que serán utilizados en el proyecto.
Plan de Validacion	Describe un enfoque y recursos que se utilizaran para la validación del sistema.
Plan de Administracion de la Configuracion	Describe los procedimientos para administrar los cambios en el software.
Plan de Mantenimiento	Describe los requerimientos de mantenimiento, costos y esfuerzo



# **ESTIMACIÓN DE PROYECTOS DE SOFTWARE**

# Estimación de proyectos de software

- La estimación de costo y esfuerzo del software nunca será una ciencia exacta.
- Muchas variables afectan el costo y esfuerzo: Humanas, técnicas, ambientales, políticas.
- Para obtener estimaciones confiables tenemos las siguientes opciones:
  - Basar las estimaciones en proyectos similares.
  - Usar técnicas de descomposición.
  - Usar modelos empíricos de estimación de costo y esfuerzo de software.
  - Esperar hasta el final del proyecto

# Estimación de proyectos de software

- Muchas veces descomponer el problema es muy complejo como para considerarlo como una sola pieza.
- La descomposición se puede hacer usando dos enfoques:
  - Descomposición del problema
  - Descomposición del proceso
- Se debe entender el alcance del proyecto y generar una estimación del “tamaño”.

# Estimación de proyectos de software

- La precisión de una estimación esta determinado por:
  - El grado de estimación del tamaño
  - La habilidad para transformar la estimación del tamaño en esfuerzo humano y tiempo.
  - La estabilidad de los requerimientos del producto y el ambiente que soporta el esfuerzo de la ingeniería del software.
- El “tamaño” se refiere a un producto cuantificable , se puede construir una medida directa (líneas de código) o una medida indirecta (puntos de función o puntos de historia)

# Estimación del Tamaño

- Existen 2 tipos de estimación:
  - Estimación del tamaño
  - Estimación de la duración
- Para un proyecto software, se tiene el siguiente proceso de la estimación de la duración:
  1. Definir funcionalidades
  2. Estimar el tamaño
  3. Derivar la duración
  4. Crear el Plan



# **ESTIMACIÓN DEL TAMAÑO CON PUNTOS DE HISTORIA**

# Estimación del Tamaño con Puntos de Historia

- Los puntos de historia son una unidad de medida para expresar el tamaño de una historia de usuario.
- Los puntos de historia son una medida relativa del tamaño.
- Son una amalgamación de la cantidad de esfuerzo involucrado en desarrollar una funcionalidad, la complejidad de desarrollarlo y el riesgo involucrado.

# Estimación del Tamaño con Puntos de Historia

- Existen dos enfoques para asignar el tamaño a una historia:
  - Seleccionar una historia que sea la mas pequeña y asignarle el valor de 1 punto de historia.
  - Seleccionar una historia que parezca ser de tamaño medio y asignarle una valor medio de la escala con la que se trabajara.

# Velocidad

- La velocidad es la medida del grado de progreso de un equipo de software.
- Se calcula sumando el numero de puntos de historia asignados a cada historia de usuario que el equipo puede completar o implementar en una iteración.
- Si se suma las estimaciones de los puntos de historia de todas las funcionalidades se obtendrá la estimación del tamaño total del proyecto.

# Velocidad

- Usando la velocidad del equipo y el tamaño total del proyecto, se puede derivar la duración la duración del mismo.
- El valor de estimar usando puntos de historia esta en el hecho de que separa la estimación de esfuerzo de la estimación de la duración.
- El esfuerzo y la duración están relacionados, pero separarlos permite estimarlo de forma independiente.

# Cuanto tiempo tomara pintar esta casa?



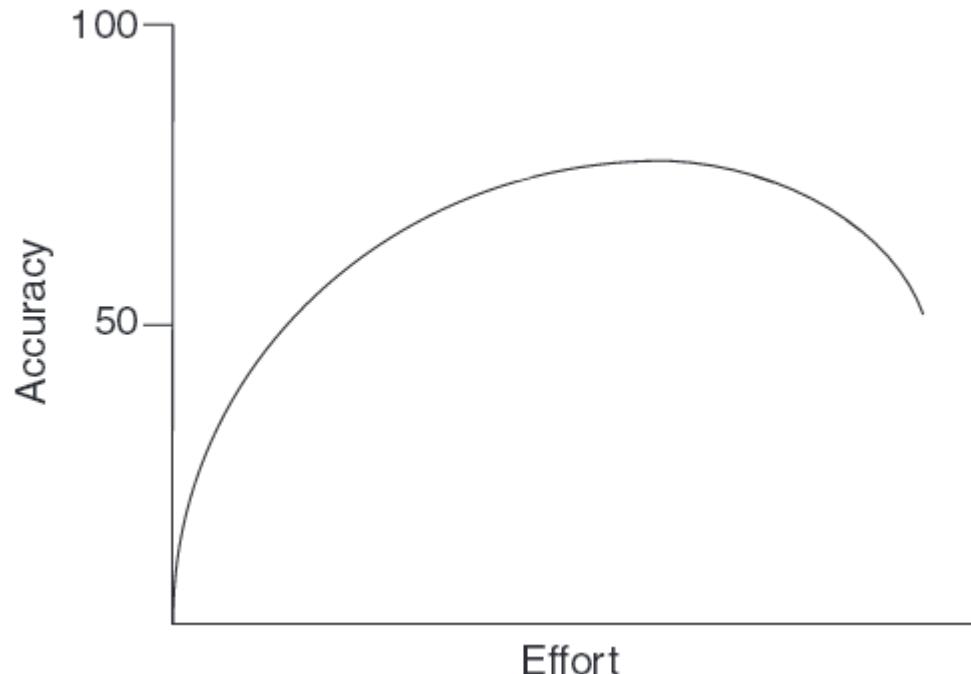
*La predicción es muy difícil, especialmente si se trata del futuro.*  
Niels Bohr



## TÉCNICAS DE ESTIMACIÓN

# Precisión de la estimación

- El esfuerzo de la estimación adicional agrega poco valor en cierto punto.



# Escala de Estimación

- Los estudios muestran que se estima mejor cuando se tiene un orden de magnitud.
- Las escalas de estimación con puntos de historia mas utilizadas son:
  - La serie de Fibonacci
  - La serie de potencias de dos.

# Técnicas de estimación

- Opinión de expertos
- Analogía
- Descomposición
- Planning Poker

# Cuando estimar?

- Se debe aplicar técnicas de estimación en dos momentos:
  - Antes de que inicie el proyecto o durante las primeras iteraciones, en este momento existe una gran cantidad de items (historias de usuario).
  - Durante una iteración se debe estimar cualquier nueva historia que se haya identificado, generalmente se hace al final de la iteración de forma que ayude en la planificación de la próxima iteración.



*“No tiene sentido ser preciso cuando ni siquiera se sabe sobre lo que se esta hablando”*

*John von Neuman*

## **REESTIMACIÓN**

# Reestimación

- Los puntos de historia no son una estimación de la cantidad de tiempo que tomara implementar la funcionalidad (feature), muchas veces se cae en la trampa de creer esto.
- La cantidad de tiempo que toma implementar una funcionalidad es una función de su tamaño (estimado en puntos de historia) y el grado de progreso de un equipo (su velocidad).

# Cuando no reestimar?

- Suponga que se tienen las siguientes US's:

ID	Historia de Usuario	Estimacion
1	Como supervisor, necesito ingresar nombres de los empleados de un proyecto	3
2	Como jefe de proyectos, necesito definir sesiones de revisión.	3
3	Como desarrollador, necesito ver las horas de una tarea en específico.	3
4	Como desarrollador, necesito actualizar la información de una tarea	3

# Cuando no reestimar?

- Al final de la primera iteración, se completaron las 2 primeras historias. Cual es su velocidad?
- El equipo ve que cada historia era el doble de tamaño y complejidad con respecto a la primera estimación.
- El equipo decide duplicar el numero de puntos de historia. Cual es su velocidad?
- Sin embargo, al inicio del proyecto se considero que las 4 historias eran de un tamaño y complejidad equivalente. Debido a que el equipo aun considera que son equivalentes, las estimaciones de las historias 3 y 4 también deben duplicarse.
- El equipo se dio mas puntos por las historias completadas, lo cual duplico su velocidad, pero también duplicaron la cantidad de trabajo restante entonces su situación es la misma, es decir la reestimación no fue necesaria.

# La velocidad es un ecualizador

- La estimación de una funcionalidad esta hecha en base a las estimaciones de otras funcionalidades.
- Usando medidas relativas, la consistencia tiene mayor relevancia que el valor de la estimación.
- Medir la velocidad en las primeras iteraciones permitirá perfeccionar el plan del proyecto.

# La velocidad es un ecualizador

- Supongamos que un proyecto consiste de 50 US's y cada uno se estima con 1 punto.
  - Se espera completar 10 historias por sprint.
  - Entonces se espera completar el proyecto en 5 iteraciones.
  - Después de la primera iteración se completaron solo 5 historias.
  - Si se deja que la velocidad corrija el error de estimación, el plan del proyecto tomará el doble de tiempo con respecto a lo que se había planificado.

# Cuando reestimar?

- Suponga que se tienen las siguientes US's:

ID	Historia de Usuario	Estimacion
1	Como desarrollador, necesito ver el grafico de tiempo invertido en una tarea.	3
2	Como lider de equipo, necesito ver una grafica de progreso de las tareas.	5
3	Como lider de proyecto, necesito ver un grafico de torta que muestre el tiempo que duraron las tareas.	3
4	Como supervisor, necesito ver un reporte que muestre el retraso de las tareas.	3
5	Como lider de proyecto, necesito subir archivos adjuntos al detalle de una tarea	3
6	Como lider de proyecto, necesito registrar nuevas tareas con su estimacion.	5

# Cuando reestimar?

- Suponga que el equipo planifico implementar en la primera iteración las historias 1, 2 y 6. Cual es su velocidad?
- Sin embargo, al final de la iteración solo se terminaron las historias 1 y 6.
- El equipo dice que no se termino lo comprometido porque la historia 1 fue mas difícil de lo esperado y que se hizo el doble de trabajo.
- Si se subestimo la dificultad de generar gráficos, en este caso la historia 1, se podría esperar lo mismo de las historias 2 y 3.

# Cuando reestimar?

- Escenario I: No Reestimar:
  - Se decide mantener las estimaciones.
  - El equipo tiene una velocidad de 8 y se espera que haga lo mismo en las próximas iteraciones.
  - El equipo sabe (por la experiencia del anterior sprint) que las historias 2 y 3 no se podrán completar en una sola iteración. A pesar que la velocidad es 8 y las historias suman 8, es mucho trabajo.

# Cuando reestimar?

- Escenario 2: Reestimar las historias terminadas:
  - Se desea ver si ajustar la estimación de la historia 1 resuelve el problema.
  - Al final de la iteración el equipo decide que se hizo el doble de trabajo en la historia 1, entonces decide duplicar la estimación. Esto significa que la velocidad es 11.
  - Debido a que las otras historias no fueron reestimadas, el equipo decide implementar las historias 2, 3 y 4. Sin embargo, se tendrá el mismo problema, las historias 2 y 3 probablemente tomen el doble de lo esperado y el equipo no lograra los 11 puntos por iteración esperados.

# Cuando reestimar?

- Escenario 3: Reestimar cuando el tamaño relativo cambia:
  - En este caso, el equipo reestima todas las historias relacionadas a generar gráficos.
  - Entonces la velocidad de la primera iteración es 11, lo mismo se espera para las próximas iteraciones.
  - En la próxima iteración solo se trabajara en la historia 2 debido a que no existe espacio para mas historias.
  - Solo se debe reestimar cuando su tamaño relativo cambia.



# **PLANIFICACIÓN**

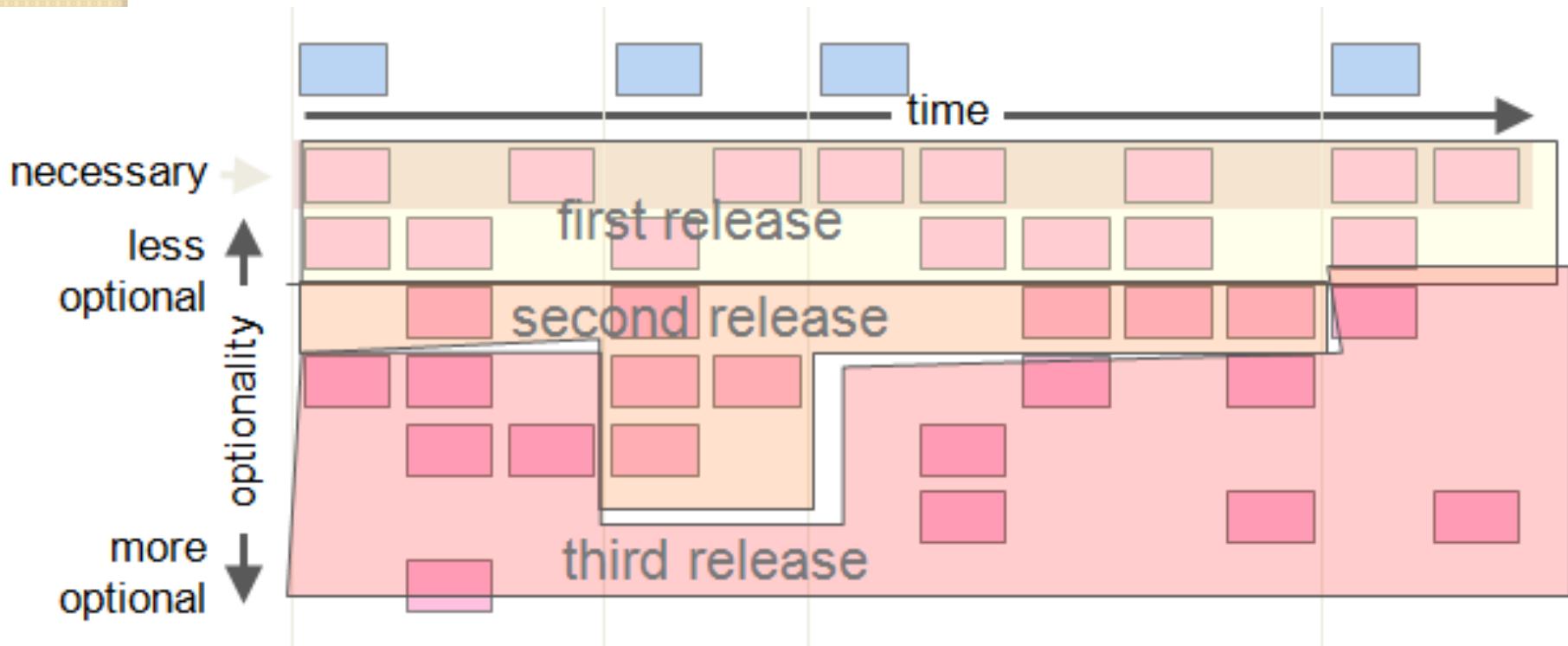
# Plan del Release

- Planificar el Release es el proceso de crear una plan a alto nivel que cubre el trabajo a realizar entre 3 a 6 meses.
- El Plan del Release es importante por:
  - Ayuda al Producto Owner y al equipo a decidir cuanto se desarrollara.
  - Ayuda a definir las expectativas sobre lo que se desarrollara en un tiempo definido.

# Plan del Release

- Los pasos generales para planificar un release son:
  - a. Determinar las condiciones de satisfacción.
    - Definidas como una combinación de tiempo, alcance y recursos.
  - b. Estimar las historias de usuario.
  - c. Seleccionar el tiempo de una iteración.
  - d. Estimar la velocidad
  - e. Priorizar las historias de usuario
  - f. Seleccionar las historias y definir la fecha de entrega del Release.

# Plan del Release



# Plan de la Iteración

- El Plan de la Iteración proporciona una vista detallada a corto plazo que los equipo usan para guiar el trabajo durante una iteración.
- El equipo debe detallar las tareas necesarias para implementar completamente las historias seleccionadas para la iteración.

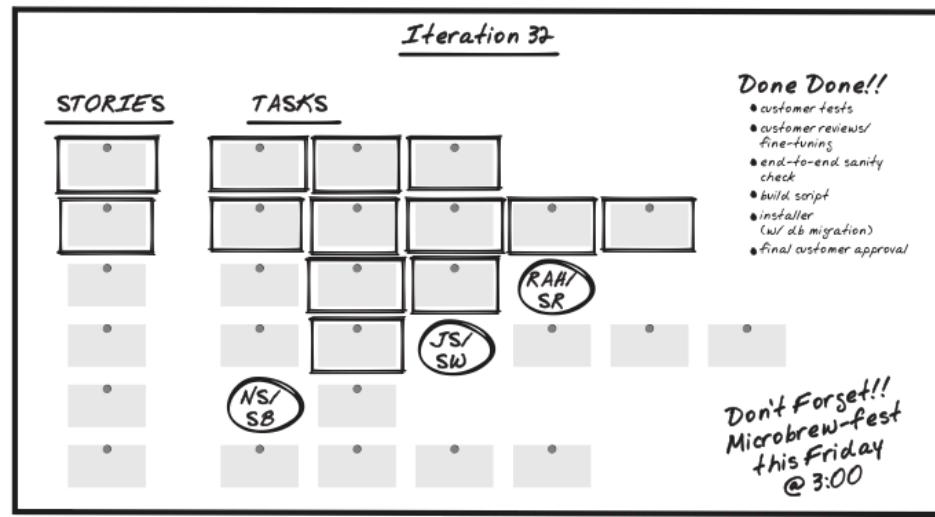
# Plan de la Iteración

- Los pasos generales para planificar una iteración son:
  - Ajustar las prioridades
  - Determinar la velocidad de la iteracion
  - Identificar el objetivo / meta de la iteracion
  - Seleccionar las historias de usuario
  - Dividir las historias en tareas
  - Estimar las tareas en horas

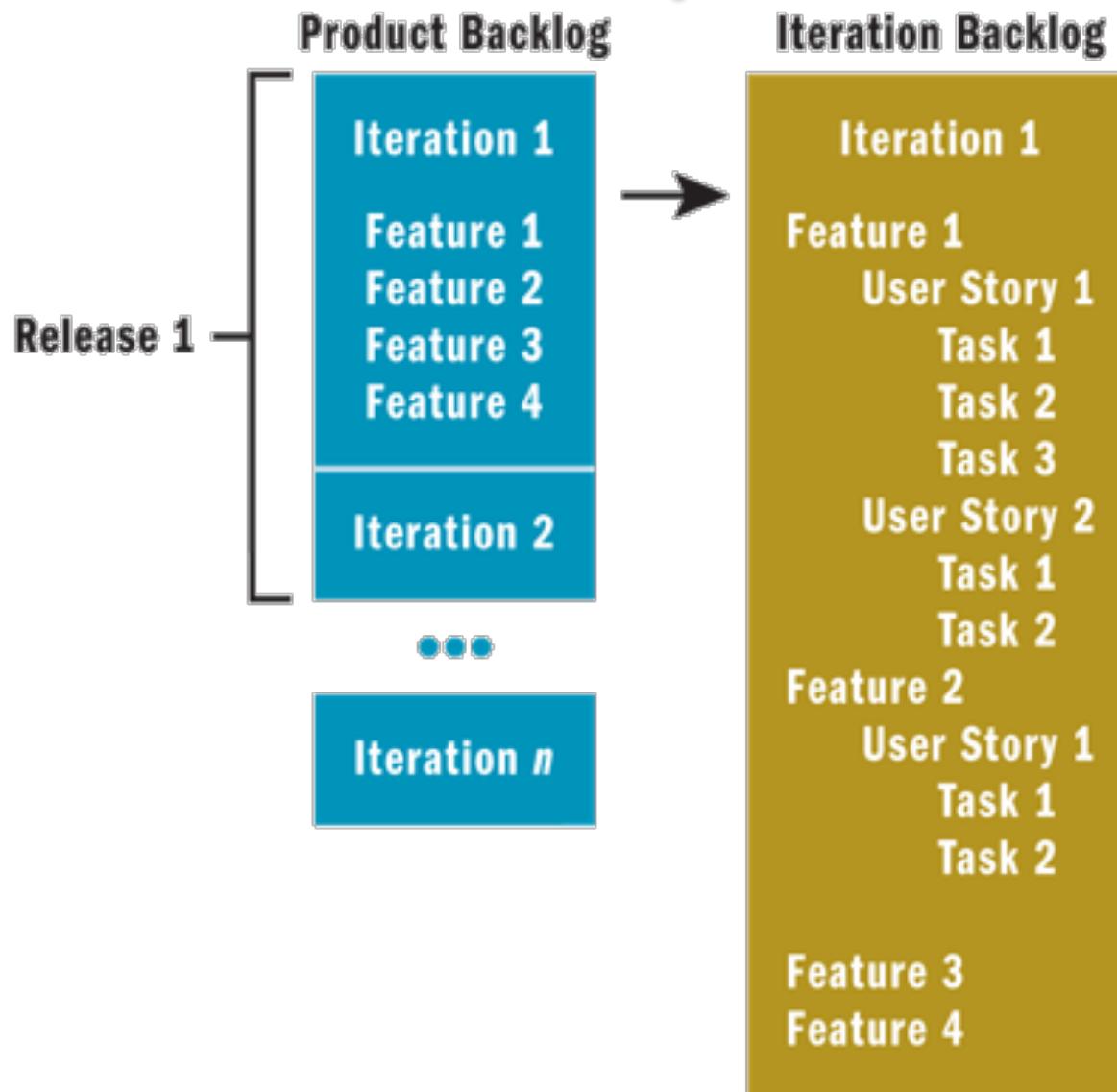
# Plan de la Iteración

ID	Historia de Usuario / Tarea	Horas
I	Como desarrollador, necesito ver el grafico de tiempo invertido en una tarea.	
	Disenar la Interfaz de Usuario	4
	Agregar los filtros del grafico	8
	Crear la estructura de datos para almacenar los datos del grafico	3
6	Crear los servicios para recuperar los datos del grafico	3
	Como líder de proyecto, necesito registrar nuevas tareas con su estimación	
	Crear el formulario de registro	3
	Mostrar el listado de tareas	2
	Crear el efecto de drag & drop en el listado para mover las tareas	5

# Plan de la Iteración



# Plan del Release y la Iteración



# Resumen

- Poner mayor esfuerzo en una estimación no incrementa su exactitud.
- La reestimación se debe realizar solo cuando hubo un cambio en el tamaño relativo.
- Un Plan de Release es una vista a alto nivel del trabajo a realizar en un periodo de tiempo.
- Un Plan de Iteración es una vista detallada del trabajo a realizar en una iteración para implementar las historias de usuario.