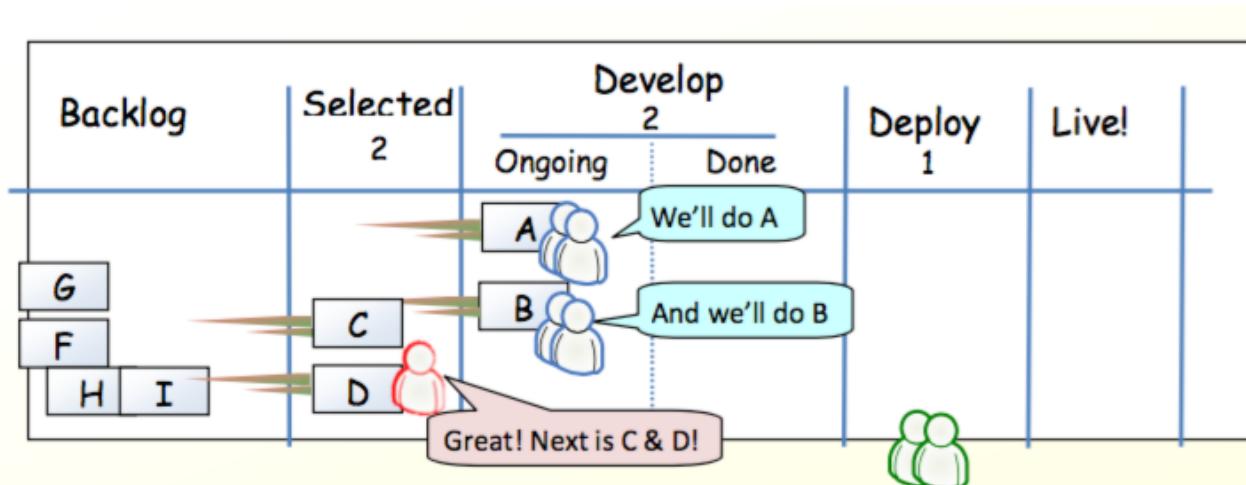
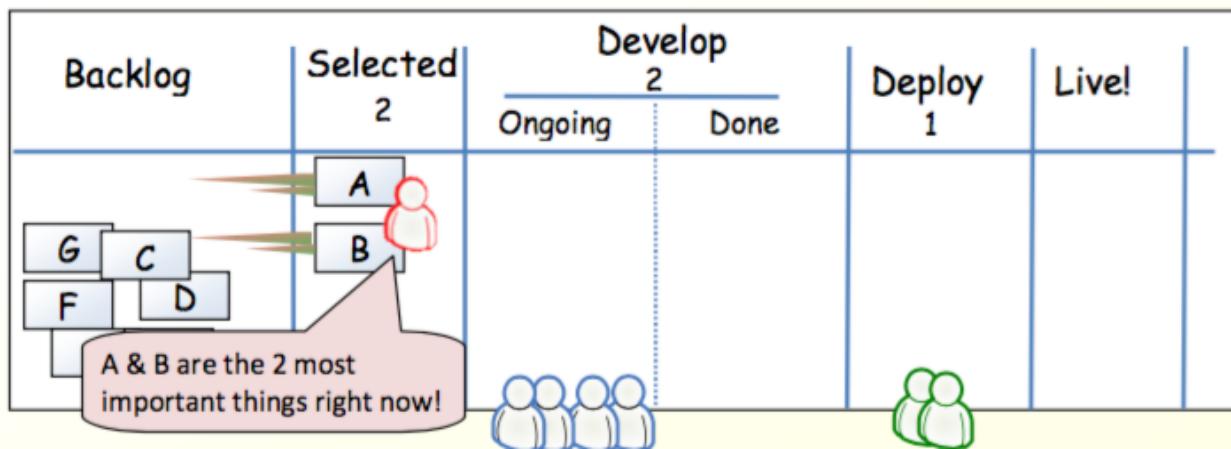
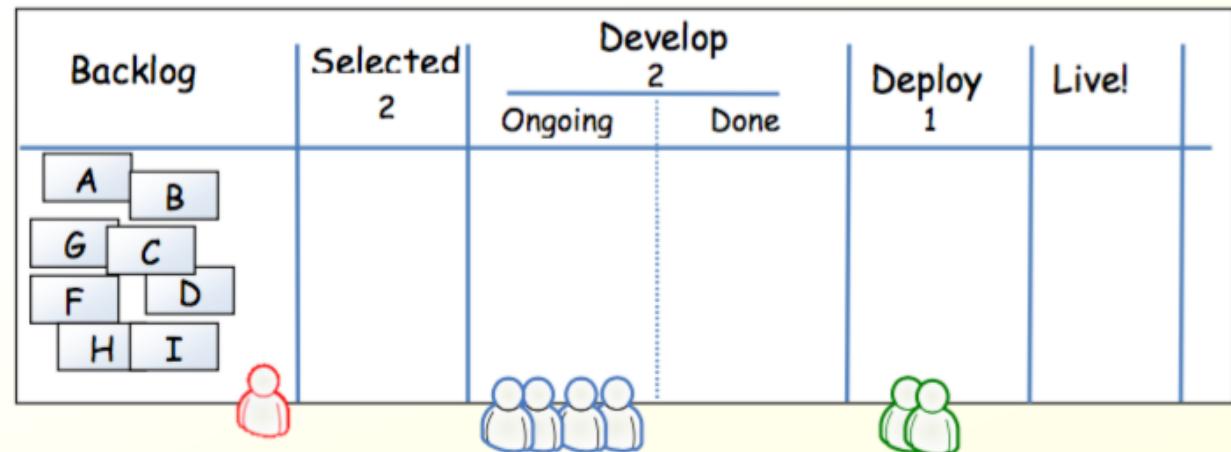
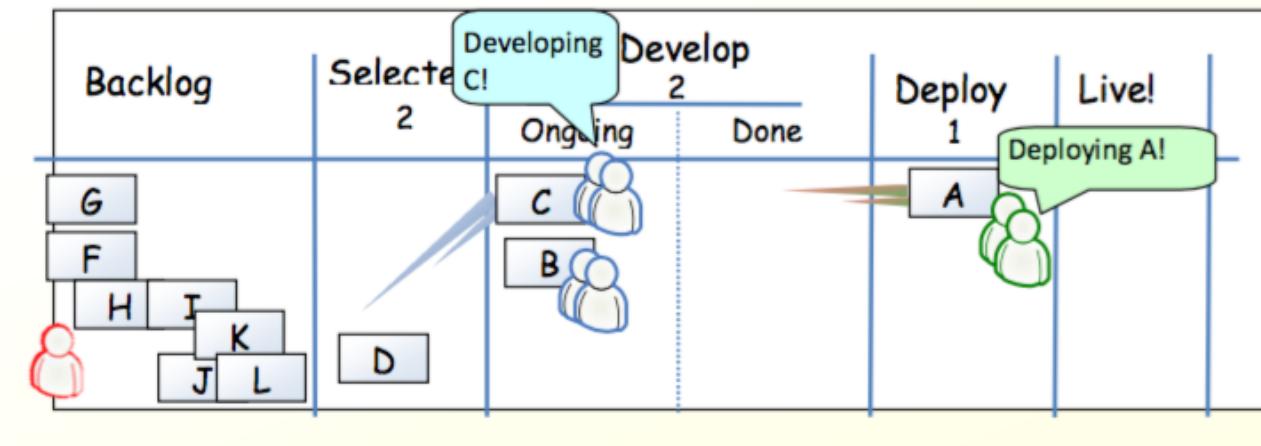
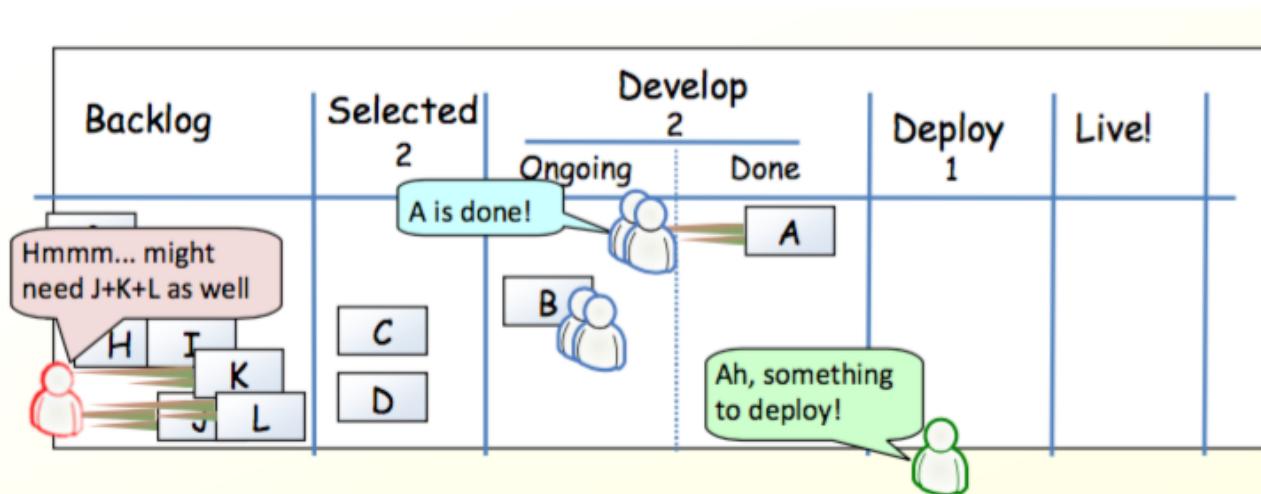
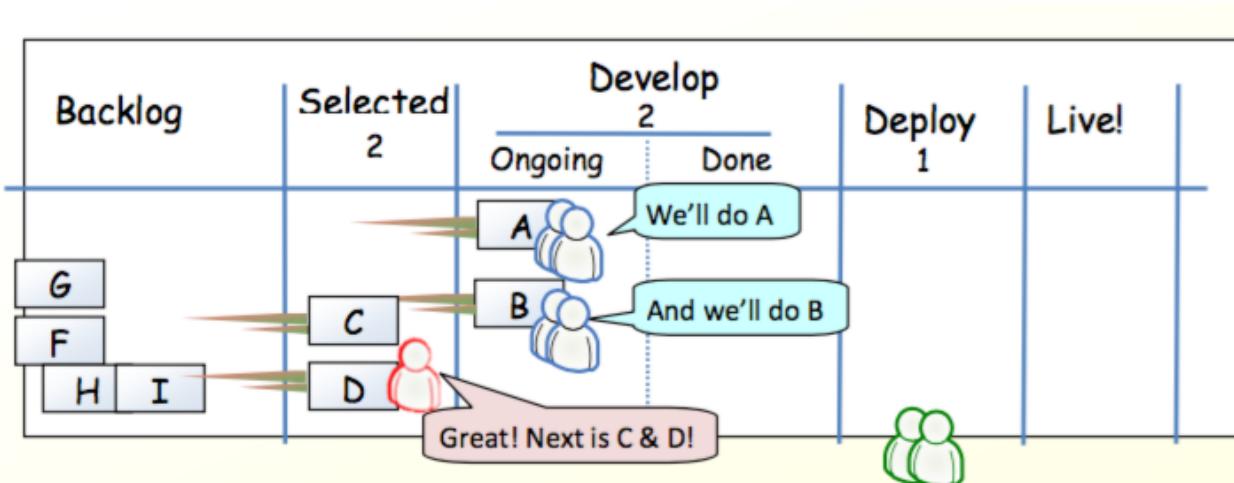
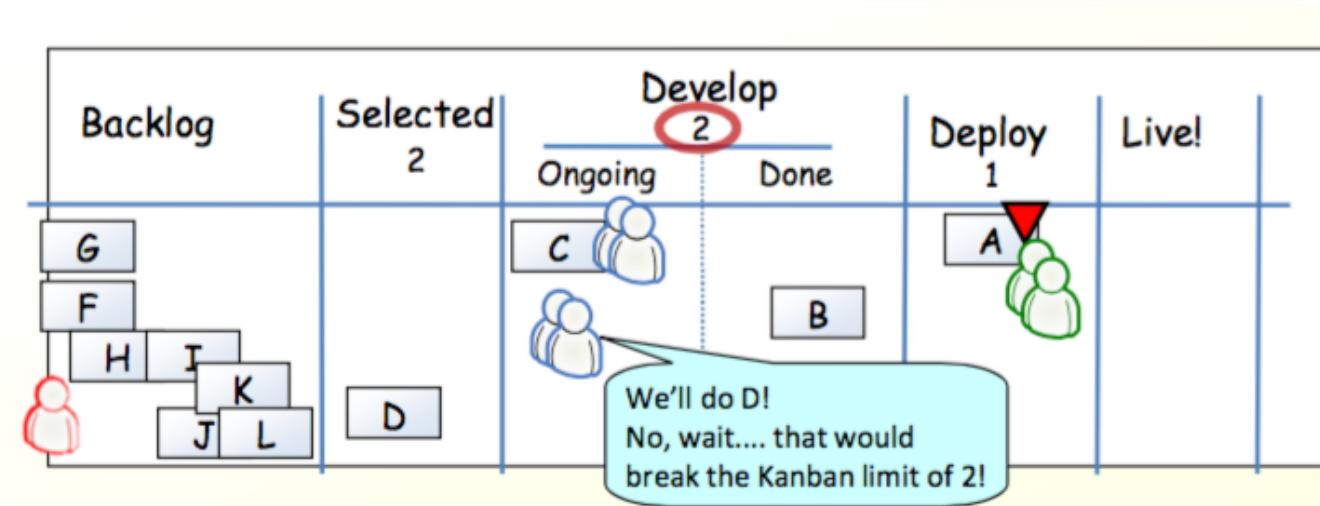
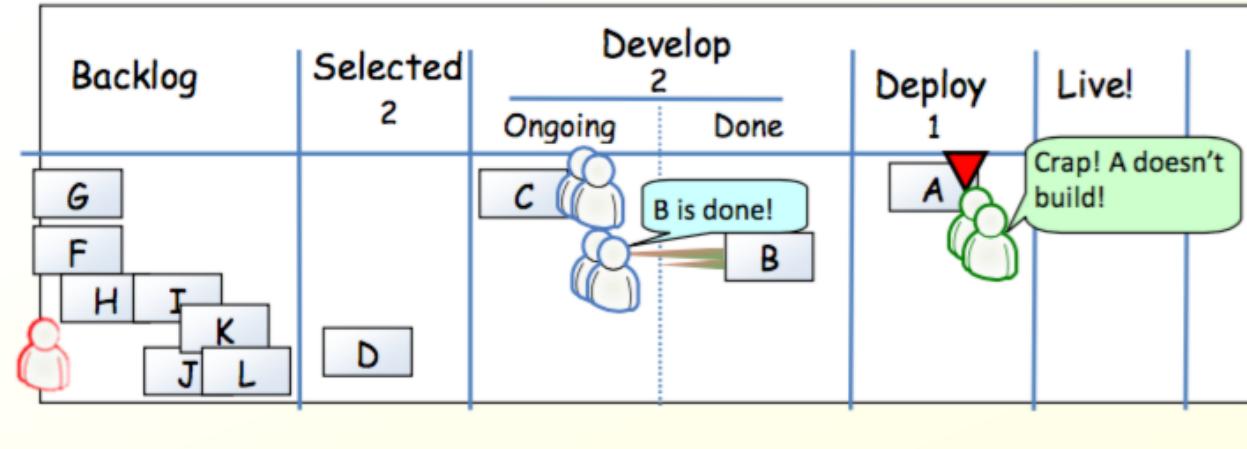
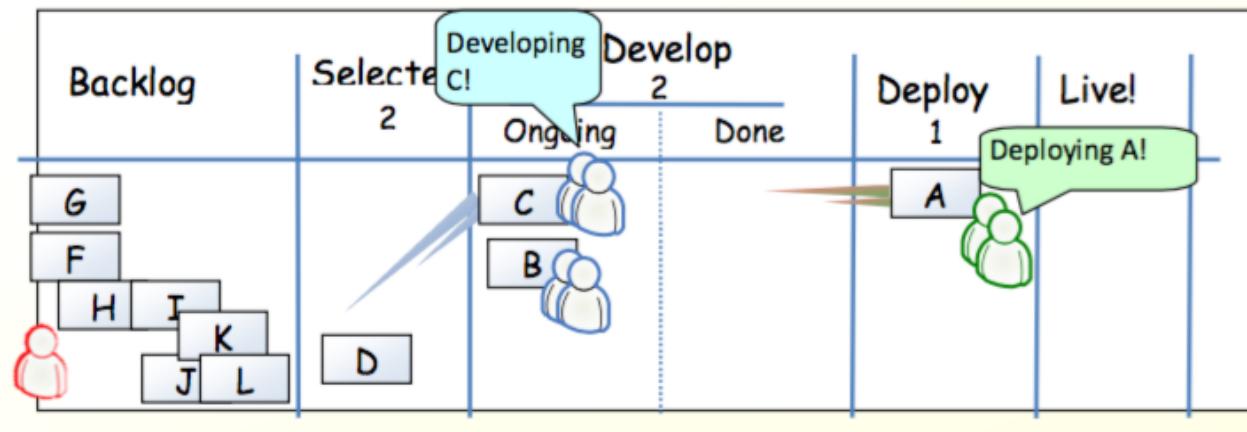
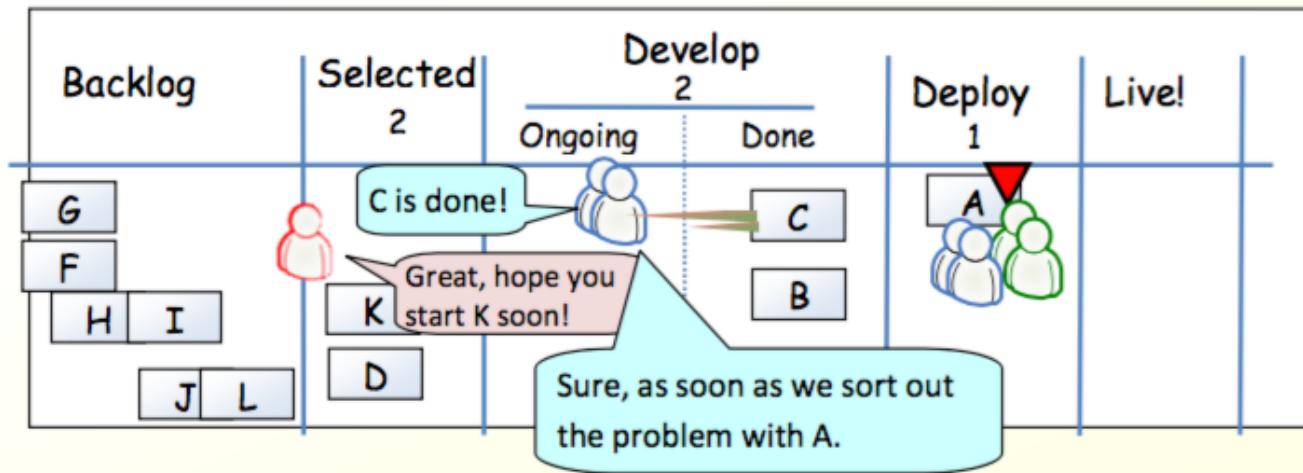
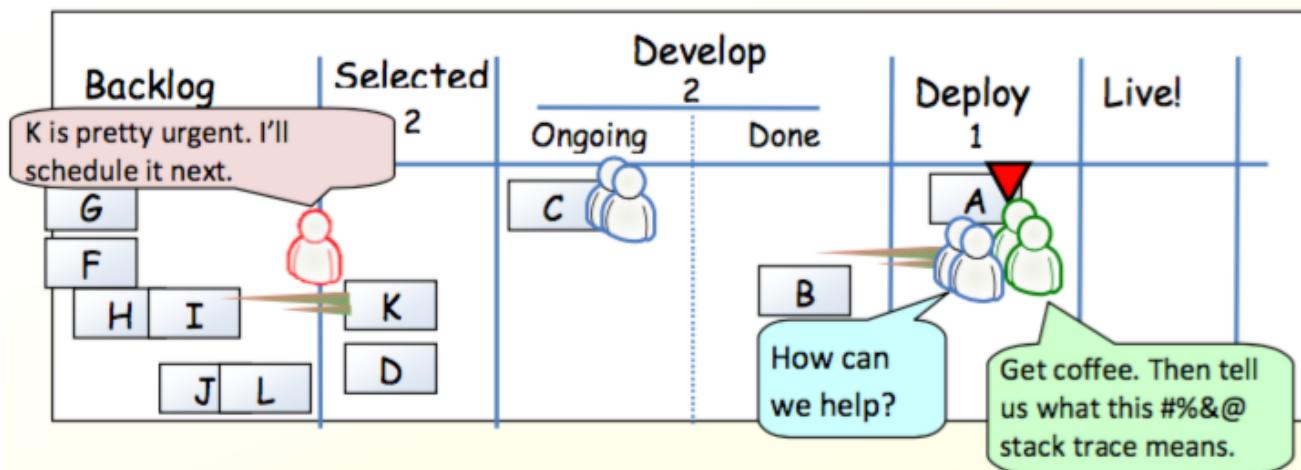
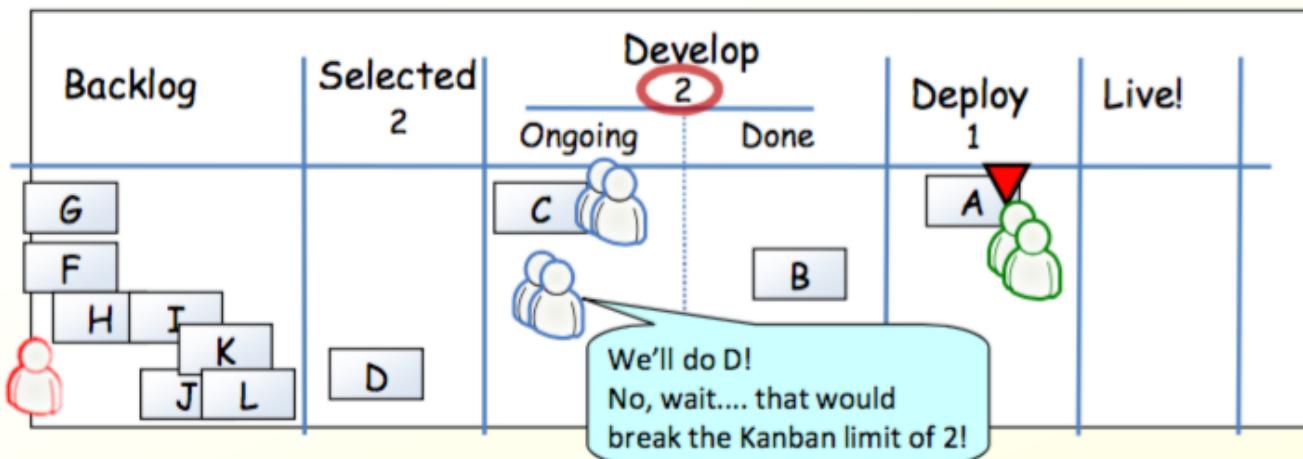


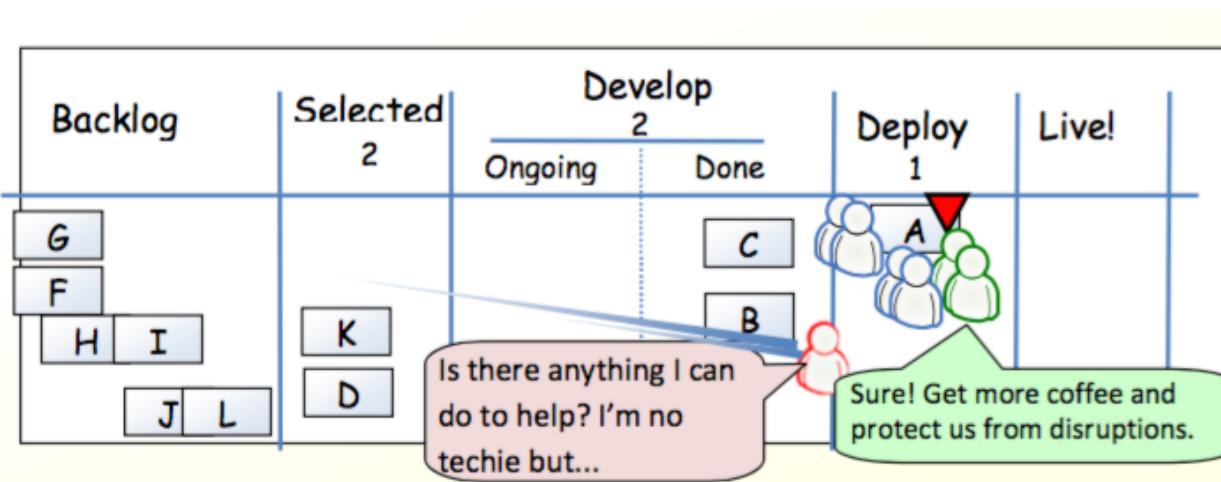
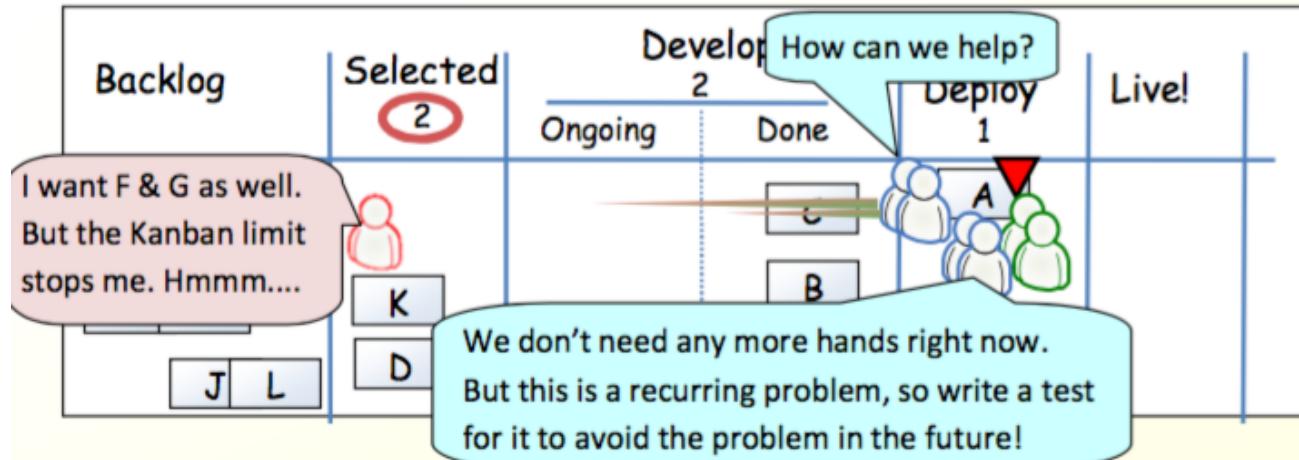
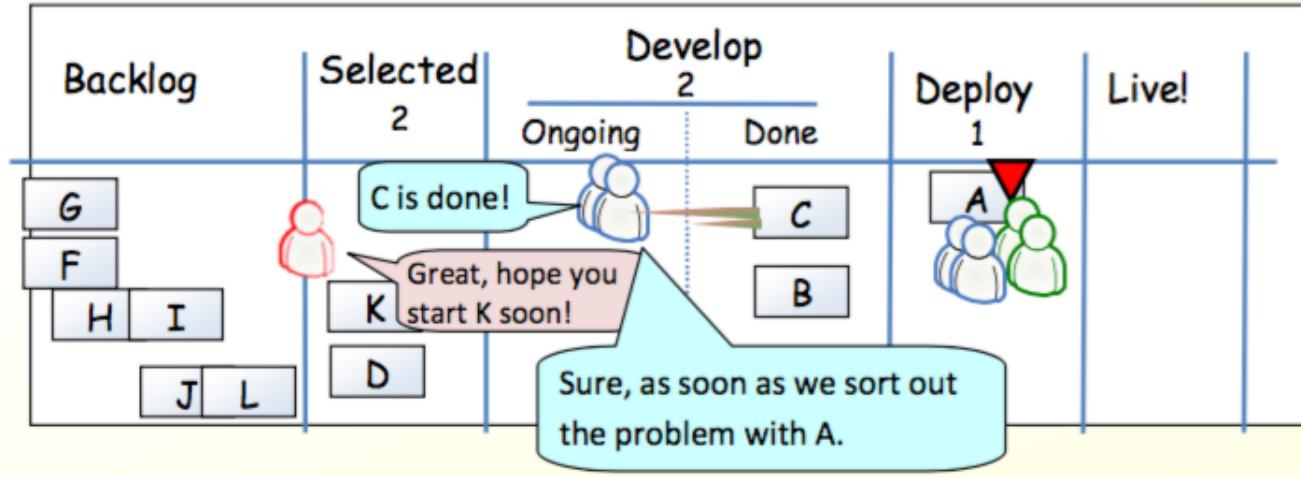
# Un ejemplo ....

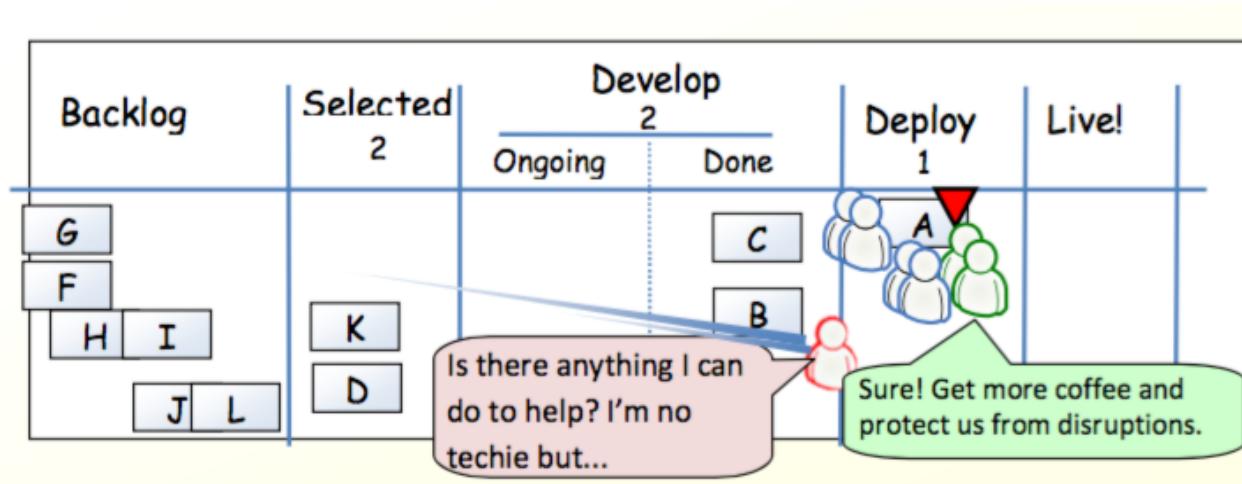




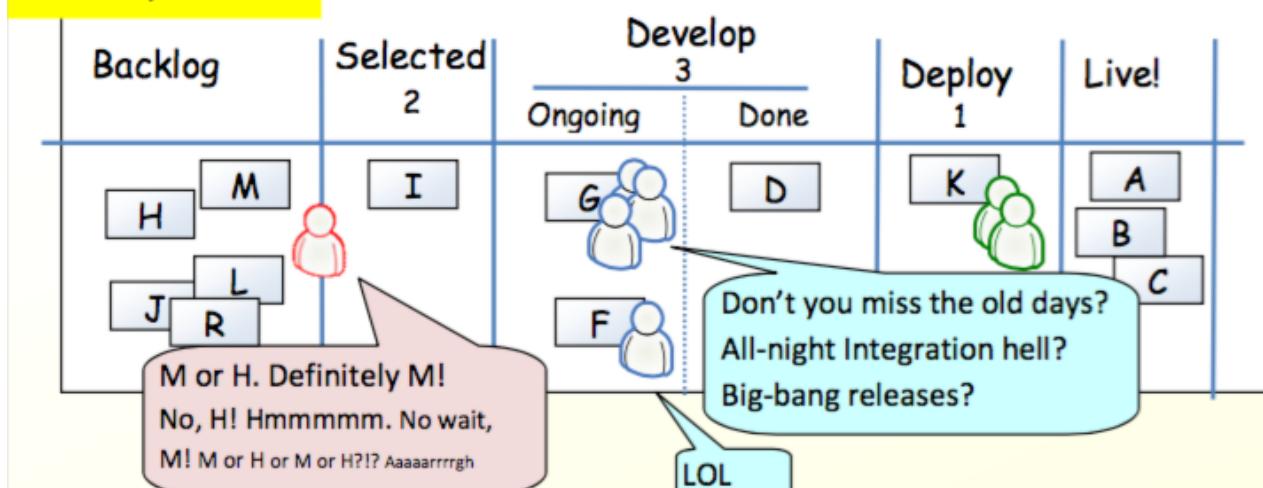






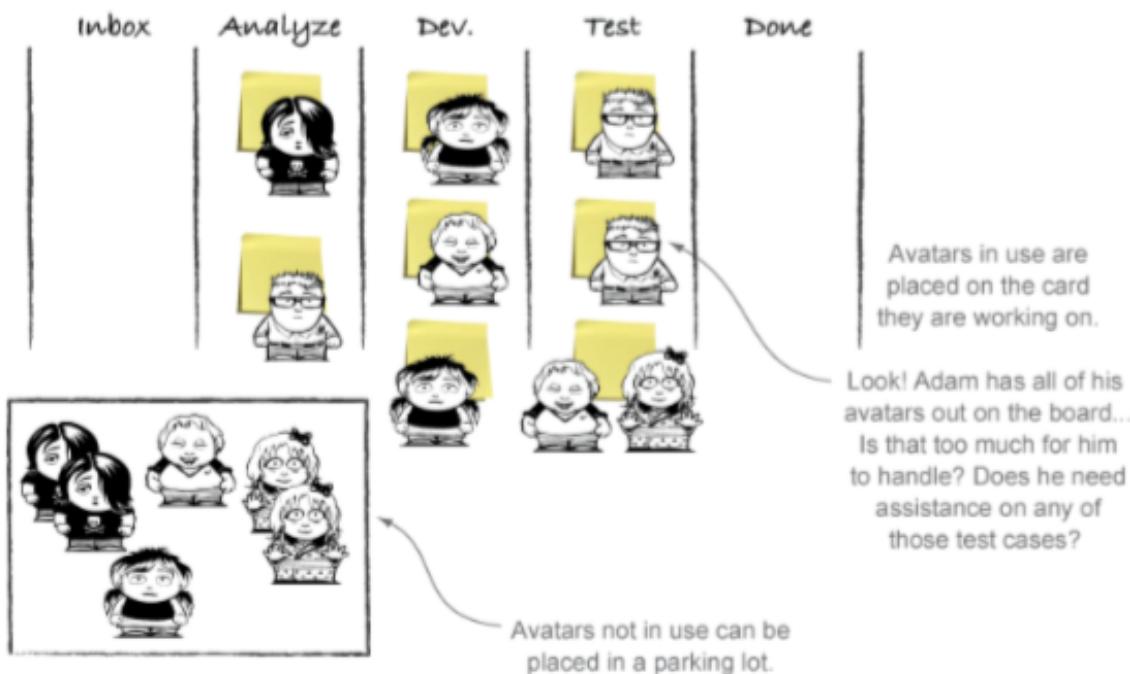


A few days later...



# Controlando Límites de Personas

- Pueden usarse avatars con las tareas
- Máximo número de un avatar en el tablero



# Otra Forma

- Asignando filas específicas del tablero a las personas



# En resumen

- Tres principios
  - Visualización colectiva del trabajo
  - Limitación del trabajo en progreso (WIP)
  - Gestionar el flujo

# Como Optimizar el Flujo

- Ajustar el WIP
- Reducir las esperas (tareas pequeñas y de igual tamaño)
- Eliminar bloqueos (ayuda desde otros)
- Equipos multifuncionales

Ivar Jacobson



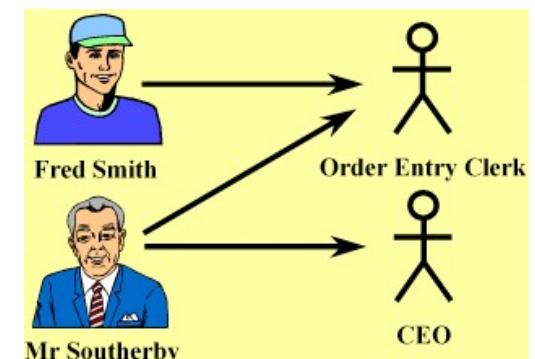
- Metodología Desarrollada por Ivar Jacobson (1992)
- Actualizada en 2011 (mejor soporte de metodologías ágiles)
- Resultado de muchos años de experiencia en Ericsson
- Integrado como parte del lenguaje de modelación unificado (UML)
  - Aspectos Estáticos (Ej. Diagrama de Clases)
  - Aspectos de Comportamiento y Dinámicos (Casos de Uso)
- Parte importante del proceso unificado de desarrollo

**ERICSSON** 



# El “qué” es la suma de metas de usuario

- Usuarios tienen metas, queremos que software les ayude a lograrlas
- Una vez que se han capturado las necesidades de todos los actores hemos completado los requisitos funcionales
- Un actor puede representar a muchos usuarios
- Un usuario puede representar a más de un actor



# EL Caso de Uso

- Relato de como un actor usa el sistema
- Describe requisitos funcionales
- Descripción es en forma de texto (se usan plantillas)
- Conjunto de CU representa una suerte de contrato de lo que el sistema será capaz de hacer

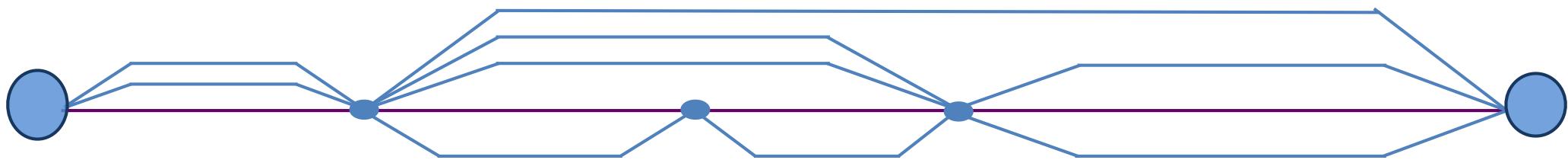
# Actores

- actores principales
  - inician el CU
  - casi siempre personas
  - tienen asociada una o más metas de usuario
- actores secundarios
  - participan pero no lo inician
  - generalmente subsistemas
  - pueden ser funciones organizacionales
- actores fuerza de escena
  - no participan en interacción pero impactan en la secuencia
  - permiten asegurar que todos los intereses sean considerados

# Un CU representa muchos escenarios

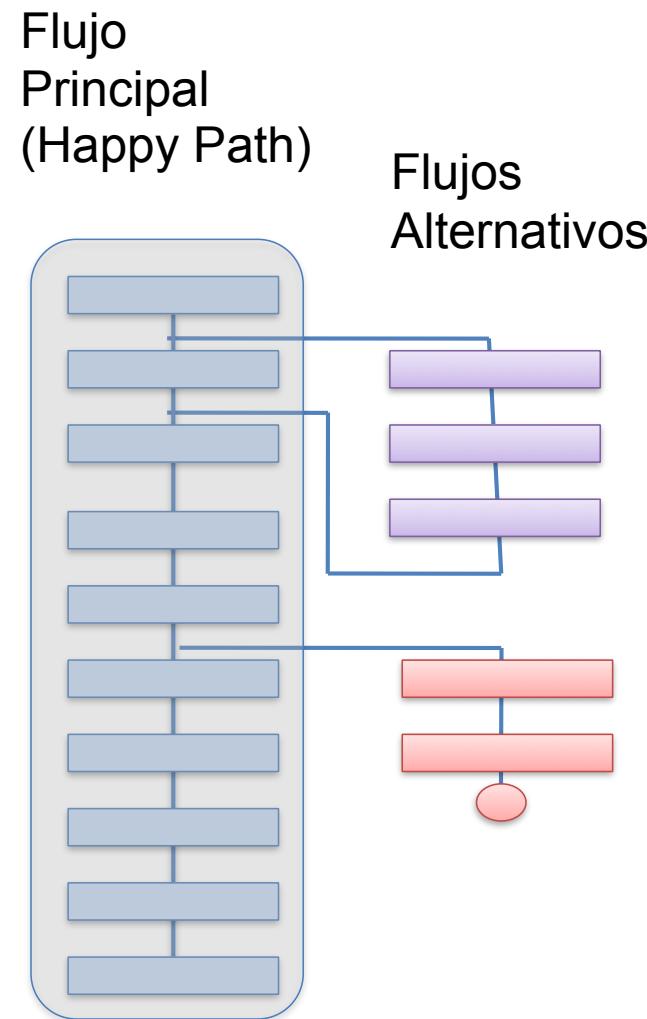
- escenario - secuencia o flujo específico de interacciones entre actores y el sistema
- algunos flujos logran objetivo o meta del usuario (exitosos) y otros no (fallidos)
- Ejemplos
  - reserva de asiento en un vuelo que termina con la confirmación por email
  - compra de entradas en ticketek falla por rechazo al pago con tarjeta

# Caso de uso: Comprar Pasaje

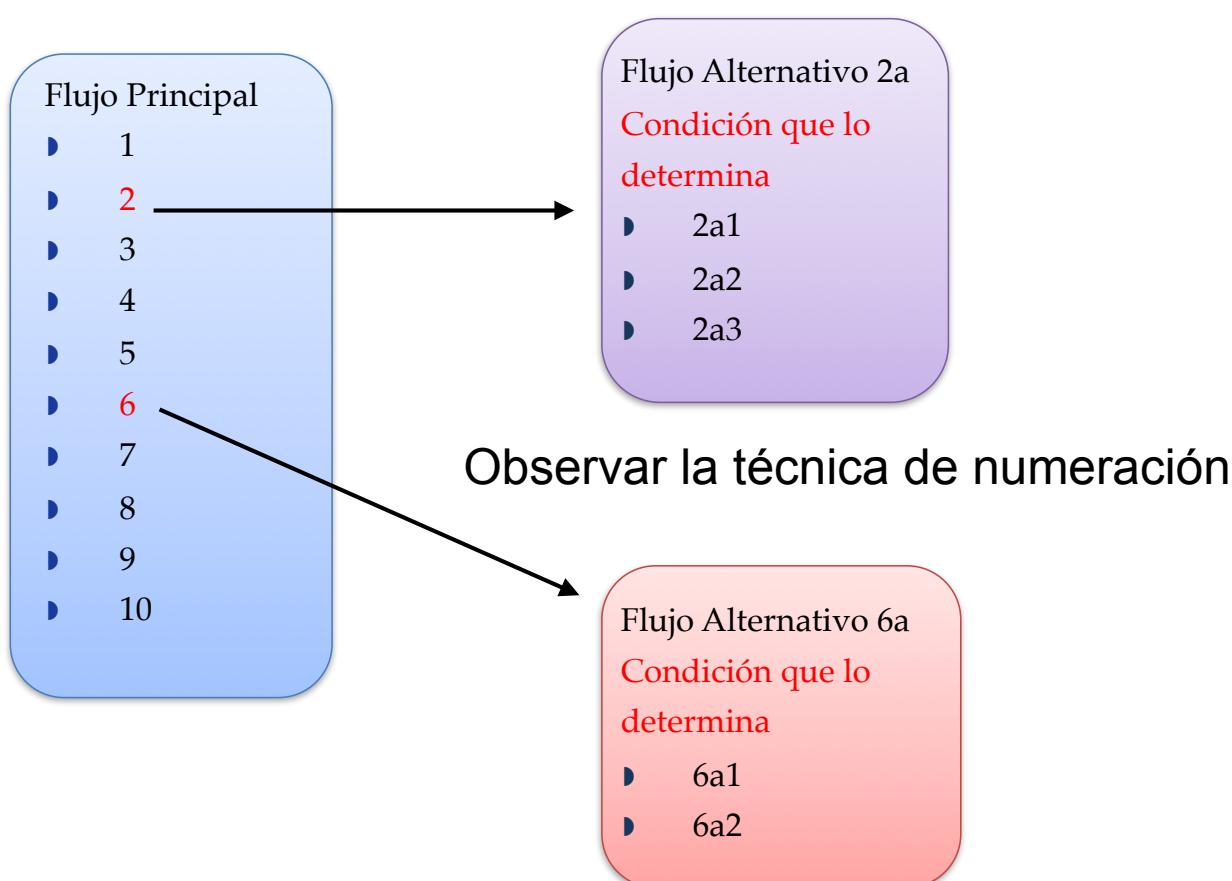


- Escenario principal
    - El usuario indica que quiere comprar pasajes para un vuelo; ... sistema envía ticket electrónico
  - Escenarios alternativos
    - al especificar la forma de pago el usuario olvida especificar la marca de la tarjeta de crédito, ...
    - al pagar con tarjeta de crédito el usuario ingresa una fecha de expiración errónea, ...

# CU: Flujo principal y Flujos alternativos



# Flujo principal se destaca separándolo del resto



# Separación entre flujo principal y flujos alternativos es muy importante

- La secuencia de pasos hasta la obtención de una meta puede tener numerosas variantes pero ...
- Normalmente hay una secuencia que se da la mayor parte de las veces (cuando todo sale segun los planes – happy path)
- Este flujo principal se separa y se pone máxima atención en él
- Una vez que entendemos el flujo principal nos preguntamos sobre las situaciones alternativas o de excepción

# Ejemplo: Planta de Revisión Técnica

UC1: INGRESAR NUEVO AUTOMOVIL

Proyecto REVTEC - Revisión Técnica

Contexto:

Actor Primario: Funcionario Recepción

Actores Secundarios: Subsistema Contable, BD

Actores Fuera de Escena: Administrador de la Planta, Organismos fiscalizadores

Alcance: Sistema Revisión Técnica

Precondiciones: Funcionario Recepción tiene papeles del auto en su poder.

Postcondiciones: Registro del auto es ingresado al sistema



# Flujo Principal

1. Funcionario de Recepción recibe pago e ingresa datos del vehículo
2. Sistema verifica que no haya sido ingresado anteriormente y solicita confirmar la información ingresada
3. Funcionario de Recepción confirma datos correctos
4. Sistema genera registro del auto y recibo del dinero con ayuda de sistema contable y BD de vehículos
5. Funcionario de Recepción entrega comprobante e indica al auto avanzar a revisión

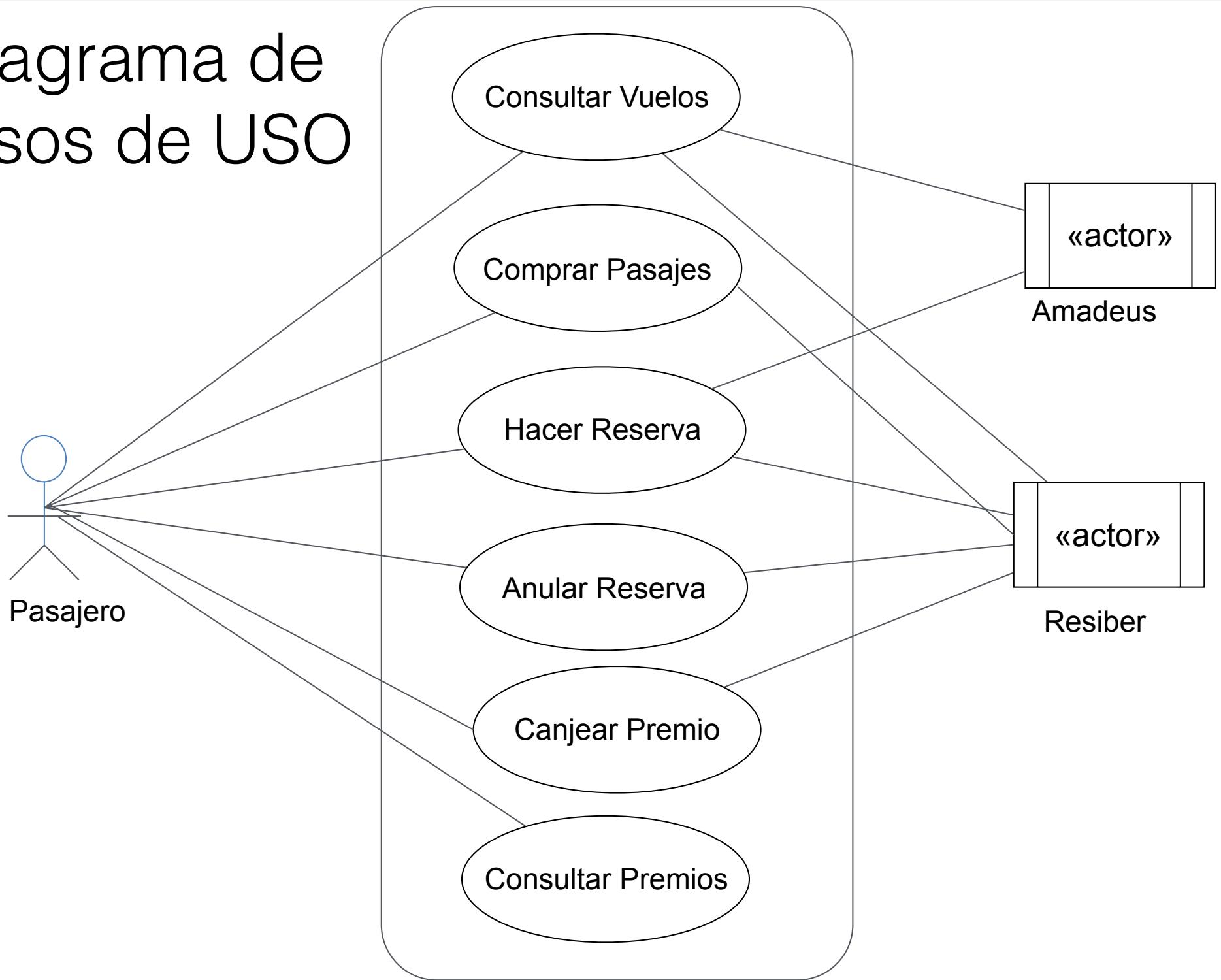
# Flujos Alternativos

- 1a. Alguno de los papeles no es el correcto (año pasado, etc)
  - 1a1 Automovilista reemplaza por papeles correctos o CU termina
- 1b Automovilista no tiene el dinero necesario
  - 1b1. Termina caso de uso
- 2a Auto ya había ingresado
  - 2a1 Se ejecuta caso de uso UC2:REVISAR AUTO POR SEGUNDA VEZ
- 3a. Información es incorrecta
  - 3a1. Funcionario corrige datos erróneos y vuelve a intentar
- 4a. Subsistema BD no puede grabar el registro
  - 4a1. Se graba record en forma local en espera de solución del problema
  - 4a2. Se intenta nuevamente la transacción cada 3 minutos en tres ocasiones más. Si no es posible se le informa al automovilista que “el sistema está caido” y se termina el caso de uso
- 4b. Subsistema contable no puede ingresar la transacción
  - 4b1. Se graba record en forma local en un archivo para ser procesado mas tarde
  - 4b2. Se genera igual el recibo para el automovilista

# El diagrama de casos de uso del UML muestra el quién y el qué

- Ilustra los nombres de los casos de uso y actores, y las relaciones entre ellos.
- Es un diagrama de contexto visual sucinto para el sistema:
  - muestra la frontera del sistema, los casos de uso que están dentro, y los actores que están fuera.
- Sugerencias:
  - mostrar los actores que son sistemas con una notación diferente de la de los actores humanos;
  - mostrar los actores principales a la izquierda y los otros a la derecha.

# Diagrama de Casos de USO



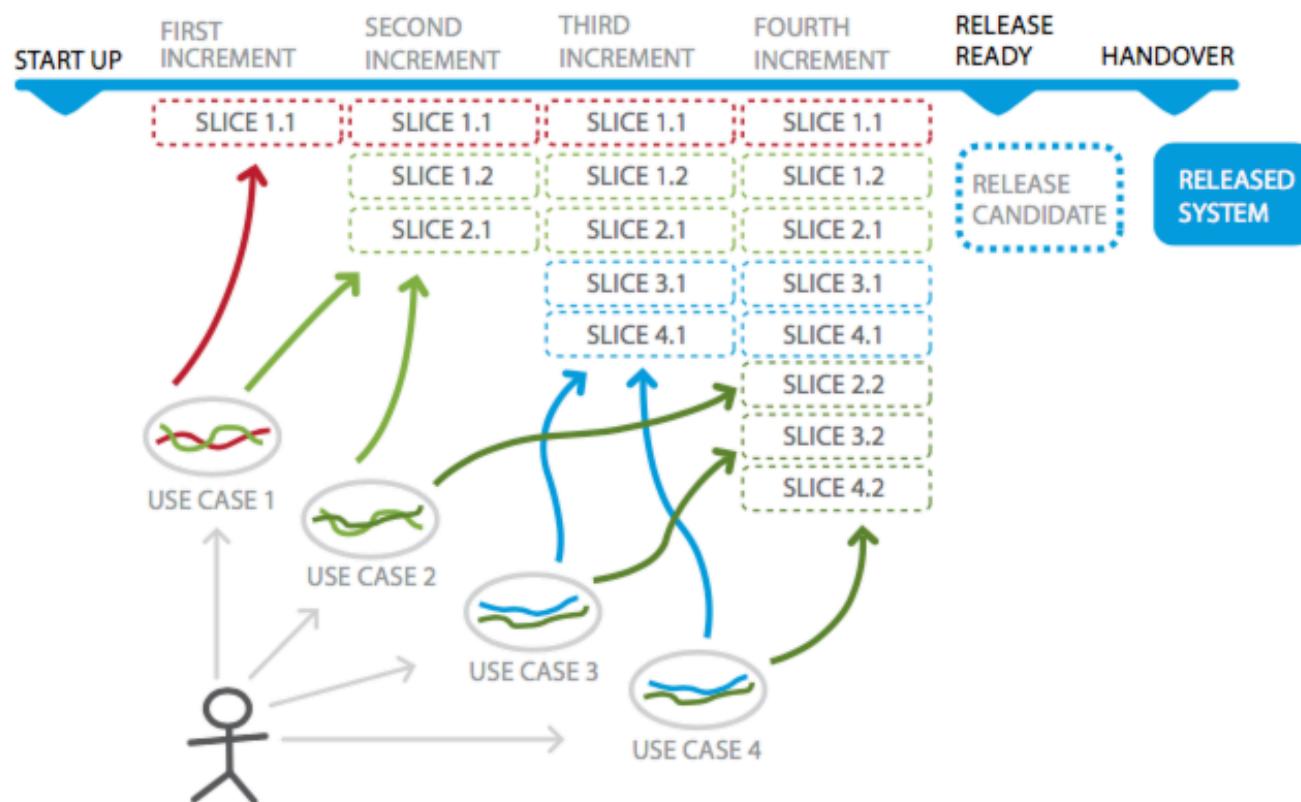
# Use Cases vs User Stories

- ambos representan un trozo de funcionalidad de valor para el usuario
- US son mas pequeños (no más de unos días de tiempo de desarrollo) porque es solo una promesa de conversación
- Un US se parece más a un solo escenario de un UC
- Los US se desechan después de la iteración, los UC son mas permanentes
- El UC es el resultado del análisis, el US suele ser el comienzo de la conversación con el usuario

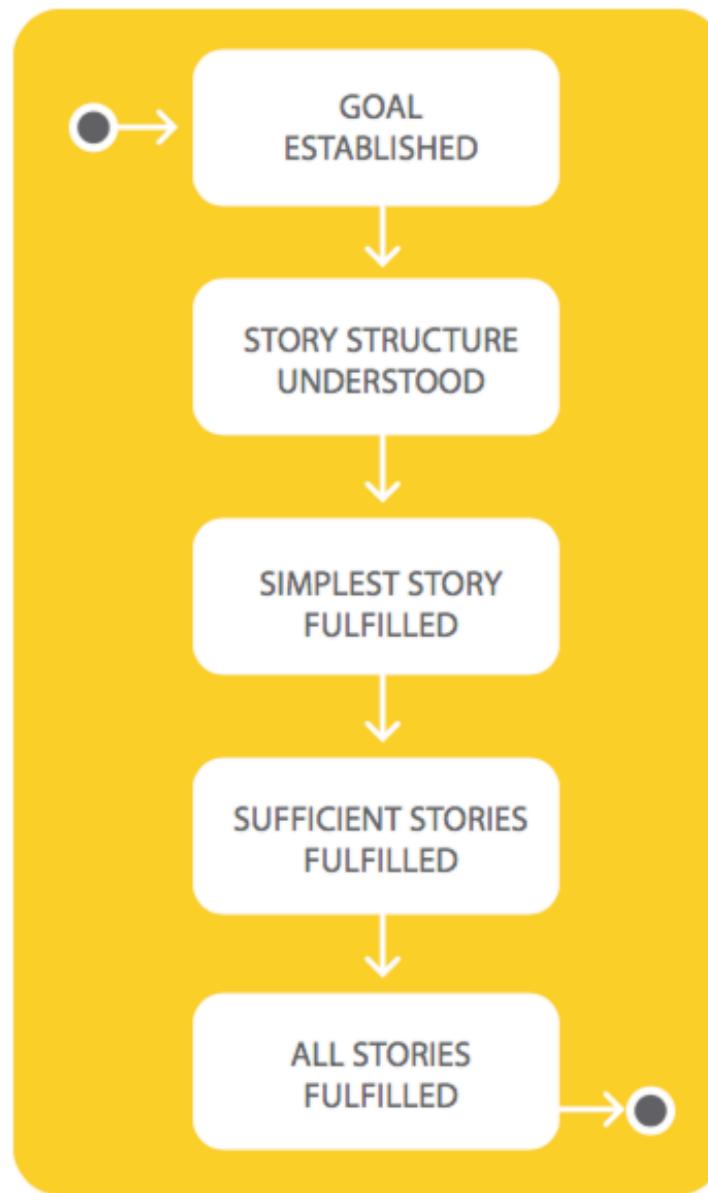
# Casos de Uso 2.0: slices

- Principios
  - hacerlo simple con relatos
  - entender el "big picture" (diagrama de CU)
  - foco en lo que tiene mayor valor (happy path)
  - construir en rebanadas (slices)
  - entregar en incrementos
  - adaptar a necesidades del equipo

# Slices and Increments



# Elaboración de un Caso de Uso



# Use Case, Stories, Slices

