

Gestión de Proyectos Software

Scrum – Roles en detalle



Contenidos

- Dueño del producto
- ScrumMaster
- Equipo de desarrollo
- Gerentes (*managers*)
- Directores de proyecto
- Estructuración de equipos Scrum
 - El modelo de Spotify



El dueño del producto



El dueño del producto

- Entiende las necesidades y prioridades de la organización que quiere que se desarrolle el software, los clientes y los usuarios
 - Se asegura de que se desarrolle la solución adecuada
- Comunica al equipo de desarrollo qué construir, y en qué orden
 - Y se asegura de que hay criterios de aceptación de cada característica, y de que estos se verifican



Responsabilidades

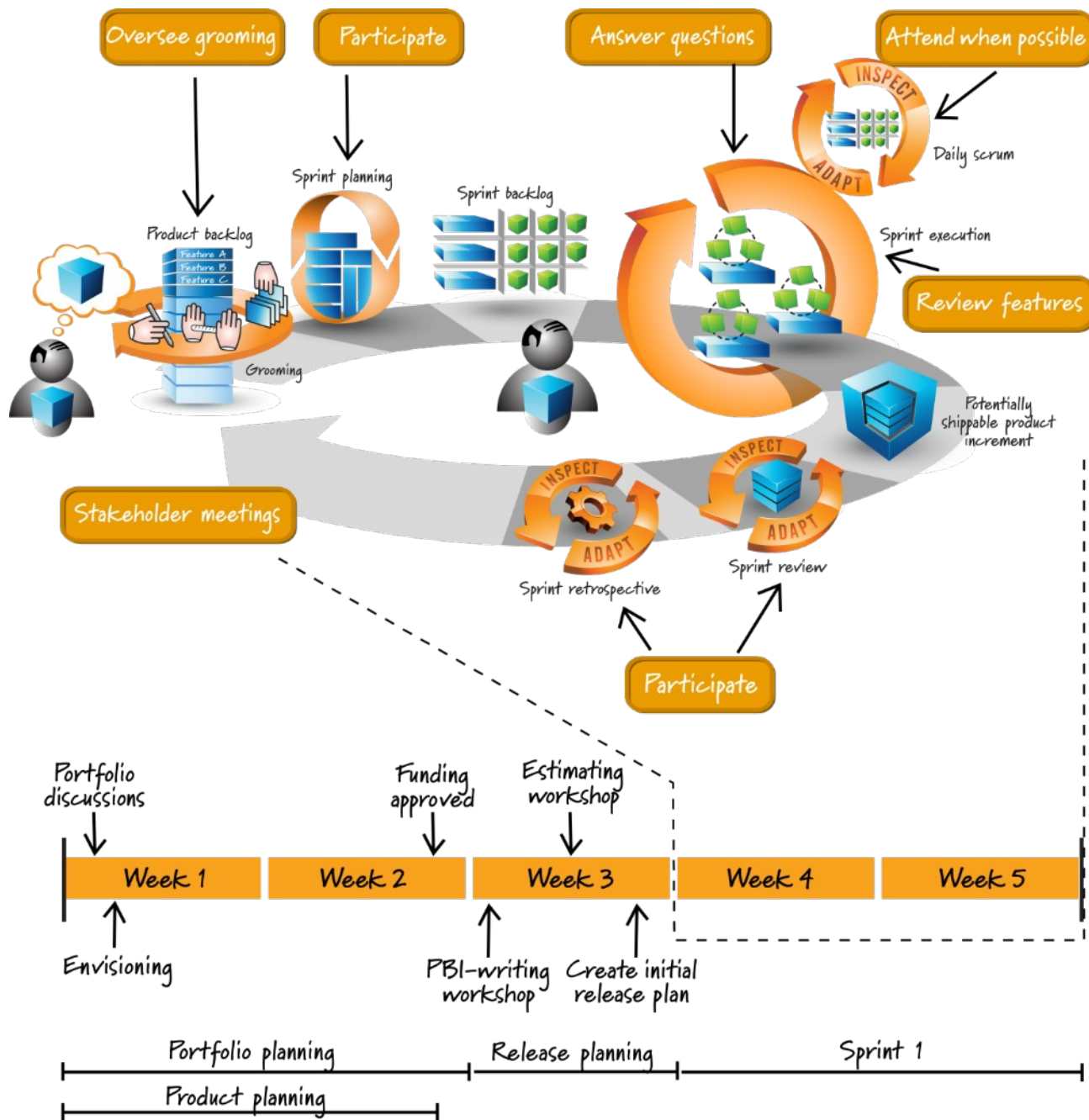
- Cierta gestión económica
 - Por ejemplo, conoce y tiene en cuenta el coste que tiene retrasar un lanzamiento para incorporar alguna característica adicional
- Participa en la planificación
- *Grooming* de la pila del producto
- Define criterios de aceptación y verifica que se cumplen
- Colabora con el equipo de desarrollo
- Colabora con los interesados (*stakeholders*)
 - Patrocinadores, clientes, usuarios...



Habilidades / Requisitos

- Conoce el dominio del problema y tiene la visión del producto que se quiere construir
 - Pero se da cuenta de que no todo se puede anticipar
- Es buen negociador y comunicador
 - Tiene que tratar con todas las partes involucradas
- Es capaz de tomar decisiones, y tiene la autoridad para hacerlo
 - Además tiene suficiente visión económica y técnica como para equilibrar ambos aspectos en sus decisiones
- Acepta la responsabilidad del éxito/fracaso del producto
- Está disponible cuando se le necesita y acepta su papel como miembro del equipo Scrum





¿Quién debería ser dueño del producto?

- Para un desarrollo interno
 - Un representante del área de nuestra empresa que se beneficiará
- Para un desarrollo comercial (algo que luego se venderá)
 - Alguien que pueda hablar por los potenciales clientes/usuarios
 - Normalmente un responsable/director de producto o alguien de marketing
- Para un desarrollo externalizado
 - Alguien de la compañía que nos contrata
- Para el desarrollo de un componente reusable (pero no directamente un producto que se vende a clientes)
 - Alguien con buenos conocimientos técnicos (la pila va a tener entradas muy técnicas)
- El dueño del producto puede ser miembro del equipo de desarrollo
- El dueño del producto no debería ser a la vez el ScrumMaster
 - Son dos papeles ligeramente “enfrentados”



¿Quién debería ser dueño del producto?

- Una misma persona puede ser dueña de varios productos (pero no muchos)
- El dueño del producto puede nombrar a alguien como “proxy” para que actúe por él en ciertas situaciones
 - Siempre que le confiera “plenos poderes” en esas situaciones, puede ser útil
- Un producto muy grande (centenares de equipos simultáneos) puede tener un dueño del producto “jefe” pero necesitará una jerarquía de dueños de producto que le apoyen
 - Organizados por “líneas de producto”, por “características” o como sea



El ScrumMaster



El ScrumMaster

- Se encarga de ayudar a que todo el mundo entienda los valores, principios y prácticas de Scrum
- Actúa como “entrenador” para el equipo de desarrollo y para el dueño del producto
- Ayuda al equipo y a su organización o empresa a desarrollar su propia aproximación a Scrum



Responsabilidades

- Entrenar al equipo Scrum
 - Observa como el equipo usa Scrum y les ayuda a resolver los problemas que surgen
- Actuar como “líder sirviente”
 - ¿Qué puedo hacer para ayudarlos a ser más efectivos?
- Autoridad (pero solo respecto al proceso)
 - Conoce Scrum y se asegura de que se siguen sus principios y valores
- Escudo frente a interferencias externas
 - P. ej. de los directivos de la empresa o de temas que surgen desde otros equipos
- Eliminador de obstáculos
 - De todo lo que reduzca la productividad del equipo y que los miembros del equipo no puedan solucionar solos
 - P. ej., la carencia de ciertos recursos
- Agente del cambio
 - Hace lo posible para que el equipo, y su organización/empresa, entiendan la necesidad de los cambios y las ventajas de corto y largo plazo por usar Scrum

Habilidades / Requisitos

- Conoce bien Scrum
 - Mejor si también tiene algunos conocimientos técnicos y del dominio de problema
- Hace las preguntas correctas
 - Su objetivo no es dar respuestas fáciles, es conseguir que el equipo encuentre respuestas propias
 - Es más un entrenador que un jugador
 - Esto requiere que sea paciente
- Colaborativo
 - Tiene que relacionarse con el dueño del producto, el equipo de desarrollo y con otras partes (resto de la empresa, clientes...)
 - Es importante que además sea transparente respecto a la comunicación que tiene con miembros del equipo y externos
- Protector
 - Mantiene al equipo protegido de injerencias que afecten a su trabajo y a su aplicación de Scrum
 - P.ej. presiones de la empresa para recortar en calidad (definición de hecho) o para acelerar el trabajo (en lugar de buscar el ritmo sostenible)



¿Quién debería ser ScrumMaster?

- Cualquiera que reúna las habilidades indicadas antes
- Típicamente no es un papel de tiempo completo y se compagina con otras actividades en la empresa/organización
 - Puede ser conflictivo ser ScrumMaster y miembro del equipo de desarrollo, especialmente en cuanto a la priorización de actividades
 - No es recomendable ser ScrumMaster y dueño del producto
 - El ScrumMaster muchas veces debe actuar de contrapeso frente al dueño del producto
 - Se puede ser ScrumMaster de más de un equipo



¿Qué hacer si...

- ...la dirección requiere que uno de los directores de proyecto disponibles tome el papel de ScrumMaster?



Actividades de un ScrumMaster

- Como apoyo al dueño de producto, asegurarse de que:
 - La pila del producto refleja las últimas ideas del dueño de producto
 - La pila del producto captura las necesidades de todos los involucrados
 - P.ej. incluye cosas que se comentaron durante las revisiones
 - La pila del producto tiene suficientes entradas y suficientemente detalladas, todo el tiempo
 - El dueño de producto entiende el problema de la deuda técnica y lo considera importante
 - La pila del producto es bien visible siempre y para todos los participantes
 - Especialmente importante si se usa alguna herramienta electrónica
 - Los diagramas de progreso (p.ej. de burndown y de burnup) se crean, se actualizan y son visibles para todo el mundo
 - El plan de lanzamientos actual sigue siendo viable



Actividades de un ScrumMaster

- Como apoyo al equipo, asegurarse de que:
 - Todo el mundo está concentrado, tiene objetivos claros, recibe feedback pronto, tiene tareas que puede realizar pero que no son demasiado simples y siente que tiene suficiente control sobre su trabajo
 - Los miembros del equipo se apoyan mutuamente
 - No se pierden oportunidades por falta de oportunidades para dialogar
 - Las retrospectivas son productivas
 - El tablero de tareas del sprint refleja realmente lo que todo el mundo está haciendo
 - El equipo tiene la combinación de habilidades necesarias para llevar a cabo el trabajo
 - No hay tareas que nadie quiere hacer y que por tanto se posponen hasta que no queda otro remedio
 - La deuda técnica no se acumula hasta el punto de hacer complicados cambios que deberían ser sencillos
 - El equipo se responsabiliza de todos los aspectos del trabajo y nadie se considera a si mismo demasiado "importante" para hacer algunas de esas cosas
 - Se llevan a cabo buenas prácticas de ingeniería
 - Pruebas automatizadas, integración continua, revisiones de código (incluso *pair programming*) etc.



El equipo de desarrollo



El equipo de desarrollo

- Grupo multidisciplinar
 - El grupo debe tener todas las habilidades necesarias para el desarrollo de software: arquitectura, programación, tests, gestión de base de datos, diseño de GUI...
- Tener equipos especializados suele ser un obstáculo para llevar a cabo Scrum de la mejor manera
 - Aunque en productos grandes con muchos equipos pueden encajar razonablemente



Responsabilidades

- Ejecución de los sprints
 - El equipo se auto-organiza para hacer el trabajo
- Inspección y adaptación diaria (Scrum diario)
- Mantenimiento de la pila del producto
 - Parte del trabajo de un sprint incluye ir preparando los próximos
- Planificar los sprints
- Inspección y adaptación de producto y proceso (revisiones y retrospectivas)



Habilidades / Requisitos

- Capaz de auto-organizarse
 - El equipo no necesita que se señale un líder que le diga a cada uno qué hacer
- Lo bastante multidisciplinar
 - Colectivamente el equipo debe tener todas las habilidades necesarias
 - Distintas habilidades, niveles de experiencia etc. aportan más ideas y facilitan la colaboración
 - Lo ideal es que todos los miembros sepan hacer varias cosas, aunque algunos estén más especializados en unas que en otras
 - Algunas veces habrá personas muy especializadas y decidiremos que repartan su tiempo entre distintos productos en desarrollo
 - Hay que evitar que se conviertan en cuellos de botella



Habilidades / Requisitos

- Actitud de mosqueteros
 - La responsabilidad de hacer el trabajo es colectiva
 - El equipo tiene éxito o no lo tiene
 - Nadie dice “ese no es mi trabajo” si hay un trabajo que hacer y tiene las habilidades necesarias para hacerlo
 - Si alguien ve un problema con algo, aunque sea en algo que no es su especialidad, lo mencionará para que se pueda discutir
- Comunicación de banda ancha
 - La información pierde valor con el tiempo, por eso acelerar el ritmo al que se comparte es importante
 - La comunicación cara a cara es lo más eficiente, pero la tecnología puede ayudar
 - La ceremonia es un obstáculo
 - Tres niveles de indirección para hablar con el cliente, tener que crear documentos sin valor para comunicar algo o tener que conseguir una firma son obstáculos
 - Un equipo de N miembros tiene $N*(N-1)/2$ canales de comunicación entre individuos
 - Si es grande, la comunicación es más complicada



Habilidades / Requisitos

- Comunicación transparente
 - Comunicar con claridad ayuda a la comprensión y fomenta la confianza dentro del equipo
- Tamaño correcto
 - Regla general: de 5 a 9 personas
 - En equipos pequeños, es más fácil que haya interacciones constructivas, se pierde menos tiempo en coordinación, nadie puede “escaquearse” sin que se note y la gente normalmente se encuentra más cómoda
 - Para productos grandes es preferible tener varios equipos más pequeños que uno muy grande



Habilidades / Requisitos

- Concentrado y comprometido
 - Más fácil si se trabaja solo en un producto
 - Hay datos que apoyan la idea intuitiva de que trabajar en más de dos proyectos al mismo tiempo disminuye notablemente la productividad
- Trabaja a un ritmo sostenible
 - No se trabaja con más intensidad a final del proyecto que al principio
- Vida larga
 - Equipo, no grupo
 - Mantener al equipo junto tanto tiempo como sea económicamente sensato
 - Hay datos que confirman la idea intuitiva de que los equipos que llevan más tiempo juntos son más eficientes y productivos
 - Crear un equipo que funciona tiene un coste, pero es una inversión



Gerentes (*managers*)



Gerentes (*Managers*)

- Scrum no menciona explícitamente a las gerentes (a veces se llaman directoras)
 - De área, de proyecto, de cuenta, ejecutivas...
 - En una empresa típica, las gerentes/directoras de proyectos y de cuentas responden ante las de área (ingeniería, marketing, ventas, QA...) que a su vez responden ante las ejecutivas
 - Puede haber otras jerarquías más o menos ortogonales a esta, por ejemplo por regiones geográficas o mercados
- Cada empresas y organización que use Scrum tendrá que encontrar como compatibilizarlo con su estructura organizativa y qué papeles pueden jugar sus gerentes

Tareas de las gerentes: definición de límites

- Los equipos Scrum se auto-organizan en un entorno dado
 - Las gerentes establecen este entorno
- Seleccionar productos o proyectos que se abordarán
- Repartir tareas entre equipos
 - P.ej., se puede decidir que el despliegue de una aplicación es trabajo de cada equipo Scrum, o que se tienen equipos de operaciones para esto
- Modificar (o establecer inicialmente) la composición de equipos
 - Lógicamente deberían consultar a los miembros de un equipo antes de incluir o retirar a alguien, pero la decisión es de la gerencia
- Delegar ciertas tareas de gestión a los equipos Scrum
 - Entre comunicarles una decisión ya tomada y delegar en ellos toda la responsabilidad hay muchos grados intermedios (convencerles, consultarles, acordarlo con ellos...)
 - Esto aumenta la autonomía y confianza de los equipos



Tareas de las gerentes: apoyo a los equipos

- No dirigirles, sino apoyarles
- Motivar y poner objetivos globales elevados
- Facilitar el aprendizaje continuo
 - P.ej. dar tiempo para ir a cursos y conferencias
 - Dar *feedback* sobre el desempeño individual
- Proporcionar liderazgo en su área
 - P.ej., la directora del área de recursos gráficos apoyará y coordinará a los diseñadores gráficos en los distintos equipos Scrum
 - También establecer estándares e iniciativas para mejorar el trabajo en sus áreas
 - P.ej., la directora del área de QA puede establecer el uso de ciertas herramientas de automatización de tests en la organización



Tareas de las gerentes: adaptación del entorno

- Entender y llevar los valores de Scrum y de las metodologías ágiles al resto de la empresa/organización
- Eliminar obstáculos en la organización
 - Esta es tarea de los ScrumMasters, pero las decisiones finales las tendrá la gerencia
- Coordinar grupos internos
 - Si unos equipos usan Scrum y otros no, es posible que los beneficios cruciales de Scrum (poner valor en manos de los clientes lo antes posible) se pierdan
- Coordinar con socios externos
 - Se pueden promover tipos de subcontratos y alianzas que faciliten el trabajo de modo ágil
 - Por ejemplo buscando alternativas a los contratos “llave en mano” con las subcontratas



Tareas de las gerentes: gestionar el flujo de creación de valor

- Cierta gestión económica (p.ej., beneficios y pérdidas en su área)
 - Una gerente es generalmente responsable de cómo los recursos de que dispone generan beneficios
- Monitorizar medidas e informes (de aquellos que aportan valor) y asegurarse de que esa información se usa para mejorar lo antes posible
 - Mantener medidas que informen sobre obstáculos para un buen flujo de trabajo
 - P.ej., analizar cómo evoluciona el tiempo medio desde que algo se empieza hasta que termina
 - Mantener medidas sobre el valor entregado
 - Qué se ha entregado, lo que ha costado en tiempo y dinero, o problemas de calidad (p.ej. errores que han surgido en producción)



Directores de proyecto



Directores de proyecto

- Ni el ScrumMaster ni el dueño de producto son el “director de proyectos versión Scrum”
 - El ScrumMaster no “manda”, y el dueño de producto tiene autoridad solo con respecto al producto
- Los directores de proyecto tradicionales tienen unas responsabilidades que en Scrum están repartidas

Actividad de gestión	Dueño del producto	ScrumMaster	Equipo de desarrollo	Otro director o gerente
Coordinar la gestión	X			X
Alcance	Nivel macro		Nivel de sprint	
Tiempo	Nivel macro	Ayuda al equipo Scrum a usar el tiempo eficazmente	Nivel de sprint	
Coste	X		Estimación de historias y tareas	
Calidad	X	X	X	X
Gestión equipo			X	Formación
Comunicaciones	X	X	X	X
Riesgo	X	X	X	X
Obtención	X			X

Directores de proyecto

- Como norma, la responsabilidad de coordinarse con otros equipos Scrum es de cada equipo
- Si hay muchos (decenas, cientos) de equipos esto es complicado
 - Se puede tener un director de proyecto cuya principal responsabilidad sea facilitar esta coordinación
 - Pero primero hay que ver si esta coordinación es tan complicada
 - Si la mayor parte de estos equipos forman clusters de comunicación más intensa, es posible que puedan organizarse sin ayuda externa
 - Con aproximaciones tipo Scrum de Scrums
 - Tener un responsable externo a los equipos Scrum no es ideal
 - Los equipos adquieren la sensación de que la coordinación no es su problema
- También se puede tener un director de proyecto para coordinar un proyecto grande en el que no todos los equipos son Scrum
 - Porque habrá que gestionar subcontratas, proveedores etc.

Estructuración de equipos Scrum



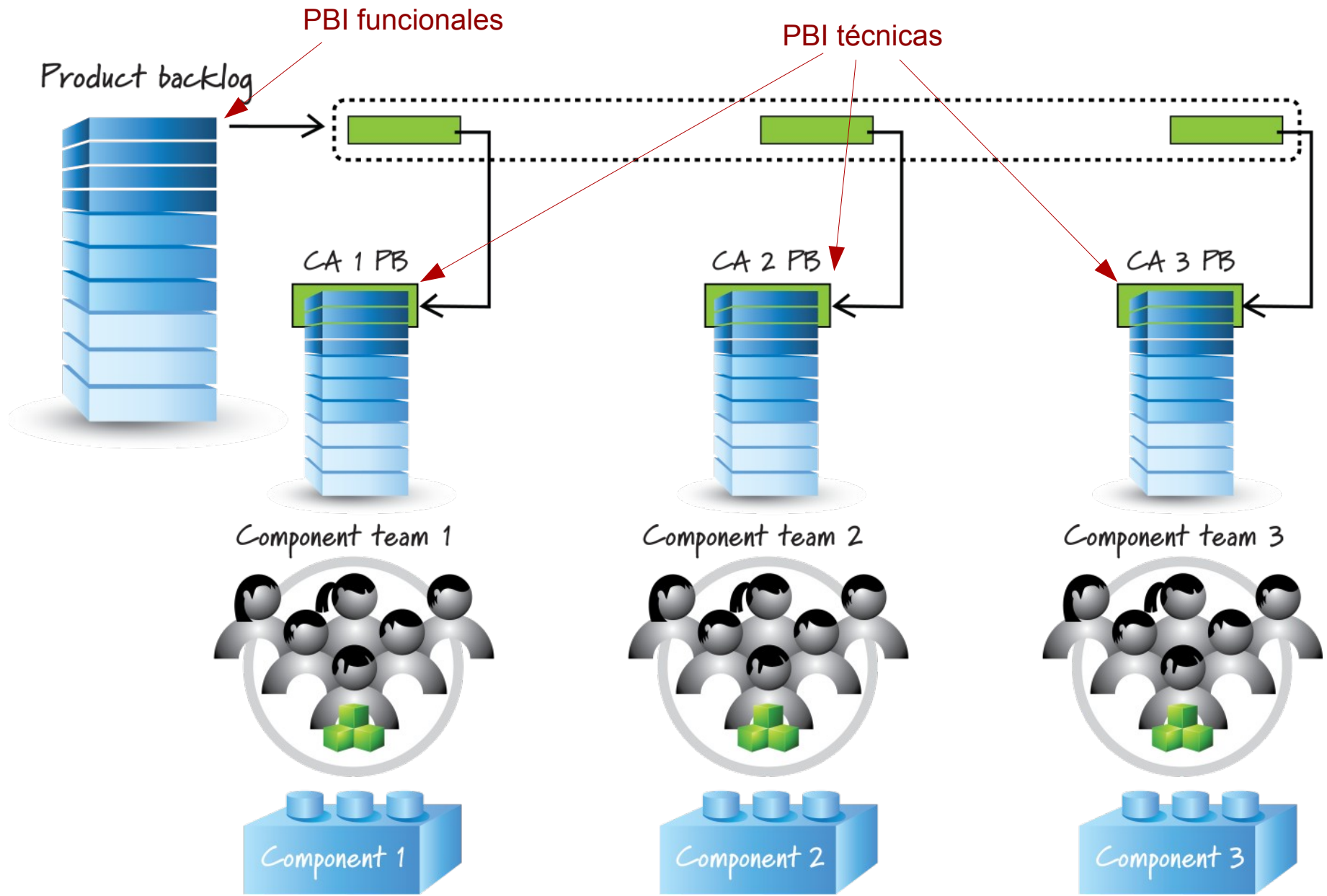
Estructuración de equipos Scrum

- Para un producto pequeño, un equipo Scrum es suficiente
- Para productos más grandes, va a ser necesario coordinar varios equipos

¿Equipos de características o de componentes?

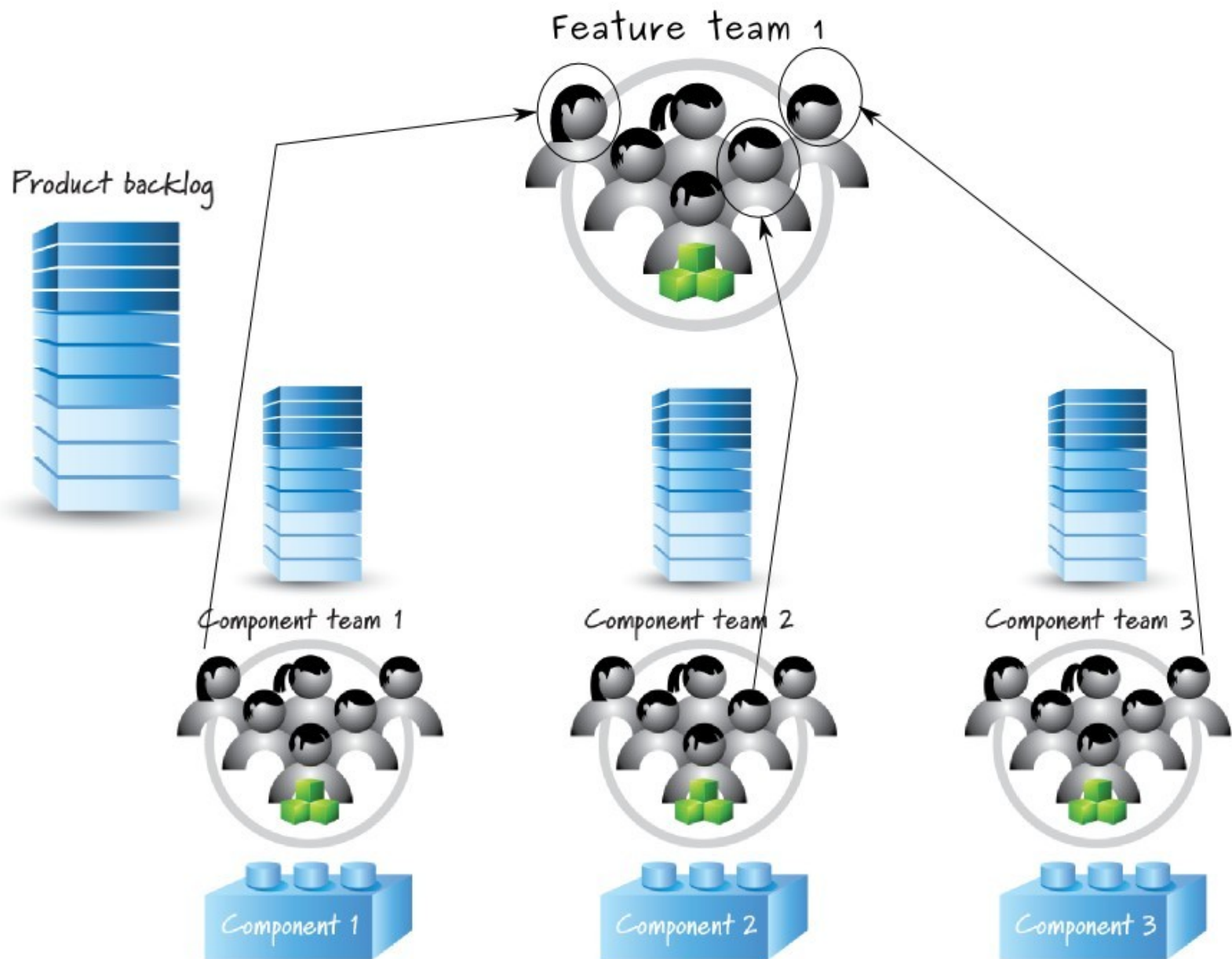
- Un equipo de características es un equipo multidisciplinar y multicomponente que toma características (historias de usuario) de la pila y las completa
- Un equipo de componentes se focaliza en desarrollar un componente o subsistema que se usa para implementar partes de diversas características
- Scrum favorece tener equipos de características, pero muchas organizaciones prefieren los equipos de componentes
 - Que haya equipos responsables de algunas partes del código y que si otros necesitan cambios se los pidan
 - Generalmente para componentes que se usarán en distintos productos/proyectos
- Los equipos de componentes pueden funcionar si hay un solo producto en desarrollo
 - Sería una forma de distribuir el trabajo, aunque quizás no la óptima
- Cogemos entradas de la pila del producto, las dividimos en sus piezas de nivel de componente, y cada pieza va a la pila del equipo de componente correspondiente
 - La división la hacen entre los equipos involucrados, o alguien con un rol de *chief architect*
- Después de que cada equipo hace su pieza, se integran para completar las características de la pila del producto
 - Para repartirse el trabajo de esta integración, los equipos se pueden coordinar con una aproximación tipo Scrum de scrums (la vemos después)





¿Equipos de características o de componentes?

- Si hay varios productos que usan los mismos componentes, seguramente por eso los creamos, alguno de los equipos de componentes posiblemente acabará siendo un cuello de botella para el desarrollo de algunos de estos productos
 - ¿Por qué no hemos terminado el producto X? Porque falta algo del componente 3, y el equipo de ese componente está implementando peticiones para el producto Y y el producto Z porque el dueño de producto del equipo de ese componente así lo ha decidido
- Incrementamos las posibilidades de no terminar de implementar cada característica
 - Porque puede fallar cualquiera de los equipos de los componentes que necesita esta característica
- La alternativa es tener equipos de características, que puedan modificar todos los componentes que necesiten
- Compartir la responsabilidad del mantenimiento de componentes reusables entre los equipos de características es más complejo que con un equipo dedicado al componente, pero es factible y debería ser el objetivo a largo plazo
 - Aunque normalmente se acaba por llegar a algún modelo que mezcle equipos de características y de componentes
 - Esta mezcla (ver siguiente transparencia) puede ser un punto intermedio hacia tener equipos de características “puros”



Coordinación de múltiples equipos: Scrum de Scrums

- Miembros de varios equipos se juntan para hacer algo parecido al Scrum diario
- Normalmente no todos los días, aunque típicamente más de una vez a la semana
- 15 minutos es referencia, pero puede decidirse hacerlo más largo
- ¿Qué ha hecho mi equipo desde el último Scrum de Scrums que pueda afectar a otros equipos?
¿Qué planeamos hacer? ¿Qué problemas tenemos que otros equipos podrían ayudar a resolver?



Coordinación de múltiples equipos: Tren de lanzamientos

- Sincronización entre equipos basada en un ritmo de trabajo común
- La metáfora del tren es porque hay un horario público de “salida de la estación” de cada lanzamiento
 - Todos los equipos tienen que subir su carga al tren antes de la hora de salida
 - El tren sale aunque no esté toda la carga
 - La carga que no va en un tren, puede ir en el siguiente



Coordinación de múltiples equipos: Tren de lanzamientos

- Hay fechas fijas para los frecuentes lanzamientos
 - Fechas fijas (y calidad fija) => alcance variable
- Los equipos tienen sprints de la misma duración
 - Empiezan y terminan en las mismas fechas. Esto permite sincronizar a todos los equipos del producto
- Los equipos se suelen agrupar en áreas de características
 - Cada área tiene su pila
 - Si hay varios equipos trabajando en un área de características, cada uno trabajará normalmente con una "vista" de esta pila
- Además de estas pilas de las áreas de características (con historias implementables) suele haber:
 - Una pila de portafolio con épicas
 - Una pila de producto
 - Con características aún grandes, pero seguramente todavía no implementables



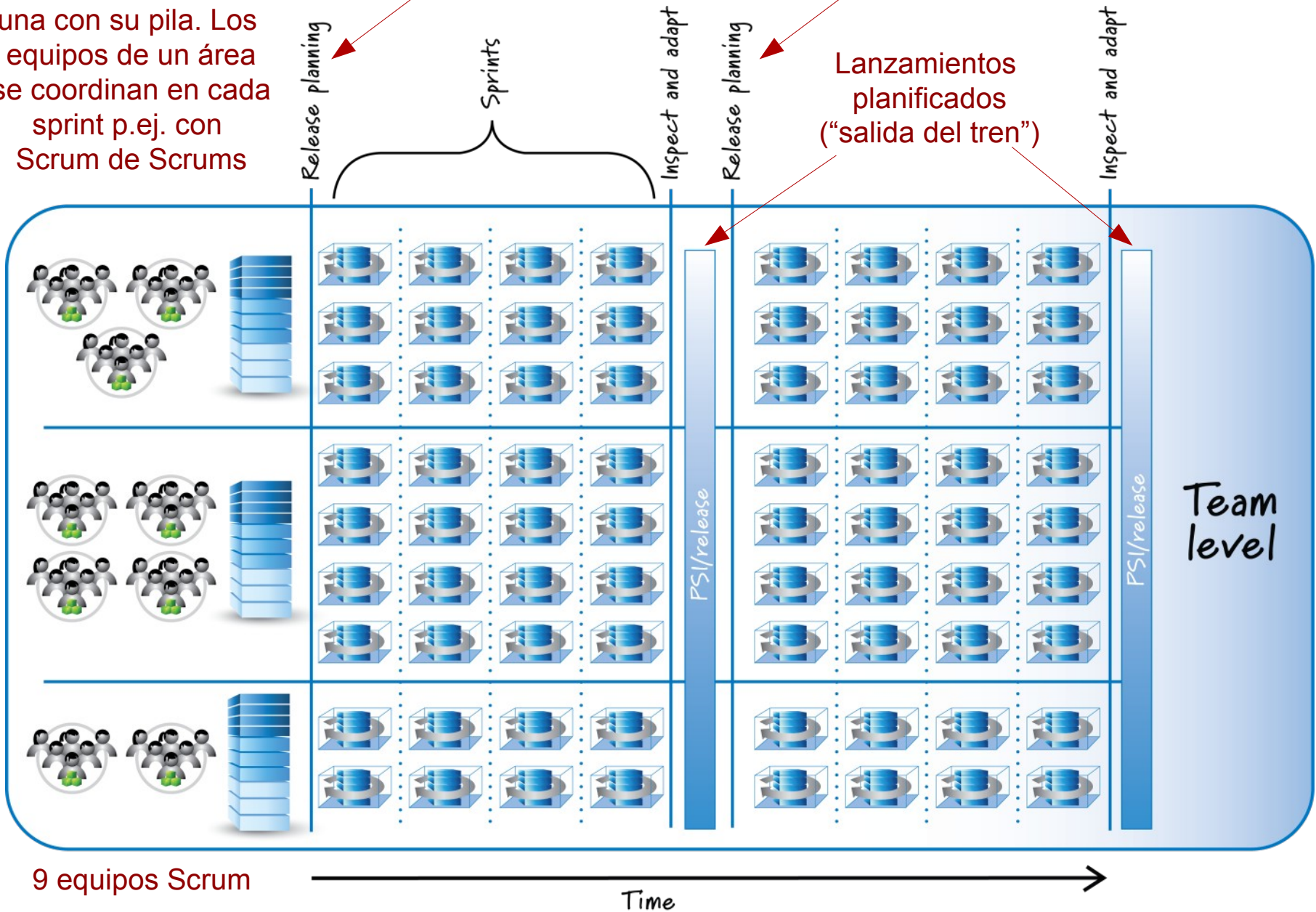
Coordinación de múltiples equipos: Tren de lanzamientos

- Se hace integración continua tanto a nivel global como al de característica y/o componente
 - Sin esto, cuando llegara el lanzamiento habría que ponerse a integrar, y si esto no se ha hecho de manera continua suele costar mucho
- Se pueden hacer sprints de “endurecimiento” a nivel global para reducir la deuda técnica y para validación y pruebas de nivel de lanzamiento
 - Por ejemplo, el último sprint antes de cada lanzamiento
- Cierta infraestructura común (aplicaciones de desarrollo, infraestructura para integración continua, infraestructura para despliegues, datos de pruebas...) debe ir estableciéndose por adelantado

3 áreas de características cada una con su pila. Los equipos de un área se coordinan en cada sprint p.ej. con Scrum de Scrums

Complejas; participan todos los equipos que tienen que hacer sprint mapping coordinado

Lanzamientos planificados ("salida del tren")



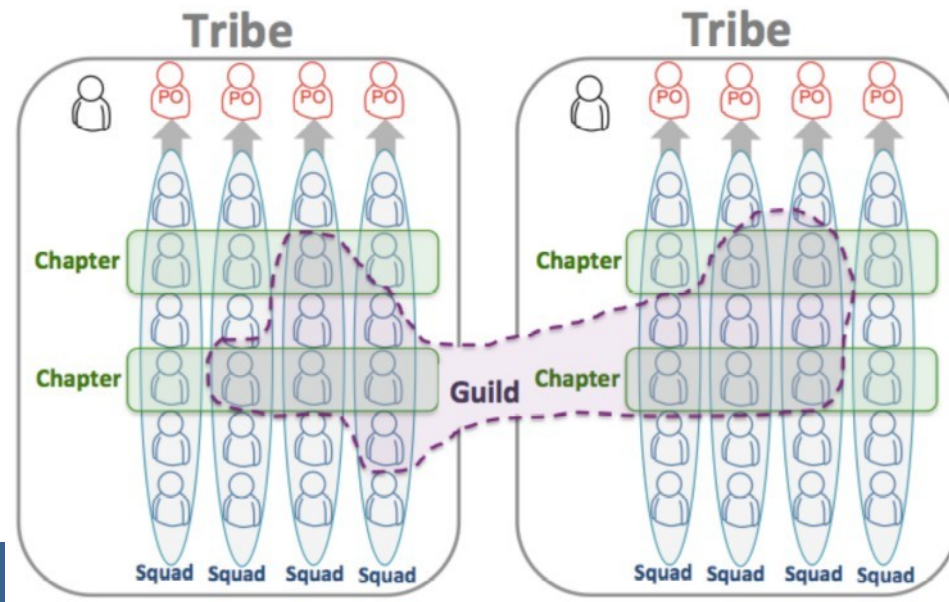
9 equipos Scrum

El “modelo de Spotify”



El modelo de Spotify

- En 2012 se publicó el modelo ágil para organizaciones grandes con muchos equipos que se usaba en Spotify (Scaling Agile @ Spotify)
 - En ese momento tenían ya 30 equipos en lo que esencialmente es su único producto
- Tuvo bastante repercusión y se intentó copiar en distintos sitios



El modelo de Spotify

- Los equipos eran denominados escuadras
- Las escuadras se agrupaban en tribus
 - Dentro de una tribu se trabaja en varias áreas pero relacionadas
 - P.ej., puede haber una tribu para todo lo relacionado con el reproductor de música en desktop
- Cada escuadra debía ser multidisciplinar y autónoma, con su propio dueño de producto para el área de características del producto en la que se centran
 - P.ej., podía haber una escuadra para la gestión de las listas de reproducción en el reproductor desktop
- Los dueños de producto de cada equipo informaban al gerente de cada tribu
 - Los diseñadores gráficos igual
- Los ingenieros sin embargo se organizaban en capítulos
 - Los que trabajaban en API de backend tenían un capítulo con un gerente, los que trabajaban en frontend Android otro etc.
 - Se pretendía poder mover ingenieros entre equipos sin que tuvieran que reportar a un gerente distinto tras el cambio
- También tenían gremios como una forma sencilla de que la gente con intereses, conocimientos o herramientas comunes pudieran compartirlos



El modelo de Spotify

- Este modelo realmente **nunca se implementó realmente**, era más un objetivo y unas ideas que la realidad de Spotify
 - También se pretendía ir ampliando y completando este modelo, pero nunca se llegó a hacer
 - Esto no estaba nada claro en el documento que se publicó
- Dirigir a los ingenieros de manera independiente a los equipos donde están no funcionó bien
 - Quienes les dirigen, los gerentes de cada capítulo, no se preocupan de los objetivos de estos equipos
 - Tampoco pueden ayudar con cosas que surjan con los otros miembros de los equipos porque no son “su problema”
 - No hay un contrapeso técnico a los dueños de producto en cada equipo. No hay una persona concreta a la que pedir responsabilidades técnicas por lo que hace un equipo
 - En un equipo podía haber 6 o 7 ingenieros, que podían pertenecer a 3 o 4 capítulos y por tanto tener 3 o 4 gerentes. Si había desacuerdos, el dueño de producto tenía que discutir con los ingenieros de su equipo y, si no se solucionaba en el equipo, escalar el problema a todos los gerentes de estos ingenieros

El modelo de Spotify

- Había demasiado empeño en que los equipos fueran autónomos
 - Pero conforme hay más equipos, es más probable que haya solapes entre los equipos, y es más interesante ir creando equipos especializados
 - Y esto resta autonomía porque cada responsabilidad que un equipo cede a otro más especializado es algo sobre lo que dejan de tomar todas las decisiones
 - Además, no había prioridades definidas claramente a nivel de la empresa
 - Un equipo puede ser autónomo, pero la dirección hacia donde se quiere ir debe establecerse globalmente desde la empresa
- No había definidas formas de colaboración entre equipos
 - Este problema es difícil, y no se puede dejar simplemente que los equipos lo resuelvan como puedan
- Tampoco la colaboración dentro de los equipos es automática. Requiere conocimiento y práctica
 - Muchos trabajadores de Spotify no sabían suficiente sobre prácticas ágiles para que esto funcionara bien automáticamente

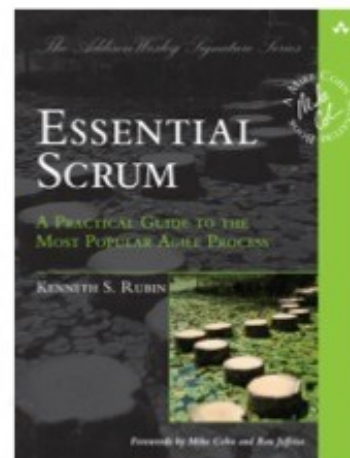
Bibliografía

- Kenneth S. Rubin. *Essential Scrum. A practical guide to the most popular agile process*
 - Capítulo 9 (*Product owner*), 10 (*ScrumMaster*), 11 (*Development team*), 12 (*Scrum team structures*), 13 (*Managers*)
- Failed #SquadGoals. Spotify doesn't use "the Spotify model" and neither should you (2020)
- Y un par de cositas de este libro (que en general me gusta poco): Andrew Pham and Phuong-Van Pham. *Scrum in Action. Agile Software Project Management and Development*
 - Capítulo 12 (*How to adapt Scrum (without destroying its agile foundations or doing negative ScrumButs)*)



Visual AGILExicon®

- ✱ Slides in this presentation contain items from the Visual AGILExicon®, which is a trademark of Innolution, LLC and Kenneth S. Rubin.
- ✱ The Visual AGILExicon is used and described in the book: "**Essential Scrum: A Practical Guide to the Most Popular Agile Process**"
- ✱ You can learn more about the Visual AGILExicon and permitted uses at: <http://innolution.com/resources/val-home-page>



Connect with Innolution:

Facebook.com/InnolutionLLC

Twitter.com/krubinagile

Visual AGILExicon®

