

Tema 5

Fases, Metodologías y Organización

5.1 Ciclos de vida

5.2 Metodologías

5.3 Organización Estructural

5.3.1 Equipo

5.3.2 Interesados

**5.4 Arquetipos de organización
empresarial**

Pablo.Bermejo@uclm.es

5.1 Ciclos de vida

- El **ciclo de vida** del proyecto se define por un conjunto de fases y las relaciones entre ellas.
- La **metodología** elegida nos indica cómo y cuándo deben realizarse las actividades (o si no hacen falta) correspondientes a cada fase.
 - Metodología de empresa: hay empresas que definen su propio modo de trabajo (o que no tienen).
 - Metodología estándar: XP, RUP,...
- Las metodologías clásicas (<1990) de proyectos software han perdido vigencia últimamente:
 - Se exige más rapidez en la respuesta y comunicaciones

5.1 Ciclos de vida

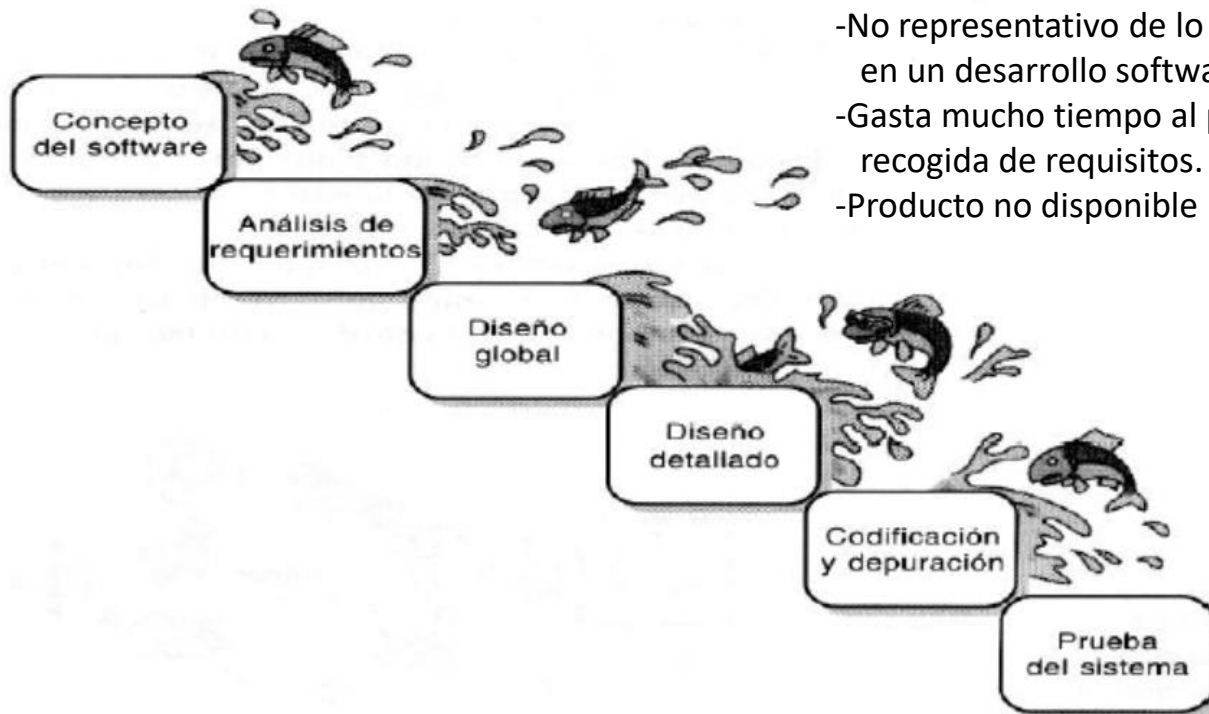
- Los interesados (positivos y negativos) intercambian información en tiempo real: más flexibilidad
- Un proyecto software dista mucho de un proyecto burocrático
- **No** hay una **metodología mejor** para todos los casos. Si te toca decidir o dar tu opinión, ten en cuenta:
 - ¿Sois un grupo grande o pequeño?
 - ¿El cliente quiere una sola entrega final o quiere participar en el desarrollo?

5.1 Ciclos de vida

- Es posible que por instinto sepas realizar tu trabajo y dirigir a tu equipo **sin metodología**; pero en cuanto el proyecto crezca estarías destinado el **fracaso**.
- Ciclo de vida \subset Metodología
Pero lo contrario no es verdad
- Ciclo de vida \cap Metodología $\neq \emptyset$
- Ciclo de vida del **proyecto** Vs. Ciclo de vida del **producto**
 - La última fase del producto es su retiro
 - Un producto puede tener asociados varios proyectos:
 - Proyecto de desarrollo
 - Proyecto para añadir funcionalidad
 - Proyecto de migración

5.1 Ciclos de vida

- Ejemplos de ciclos de vida bien conocidos (no confundir con metodologías):
 - Cascada (70s)

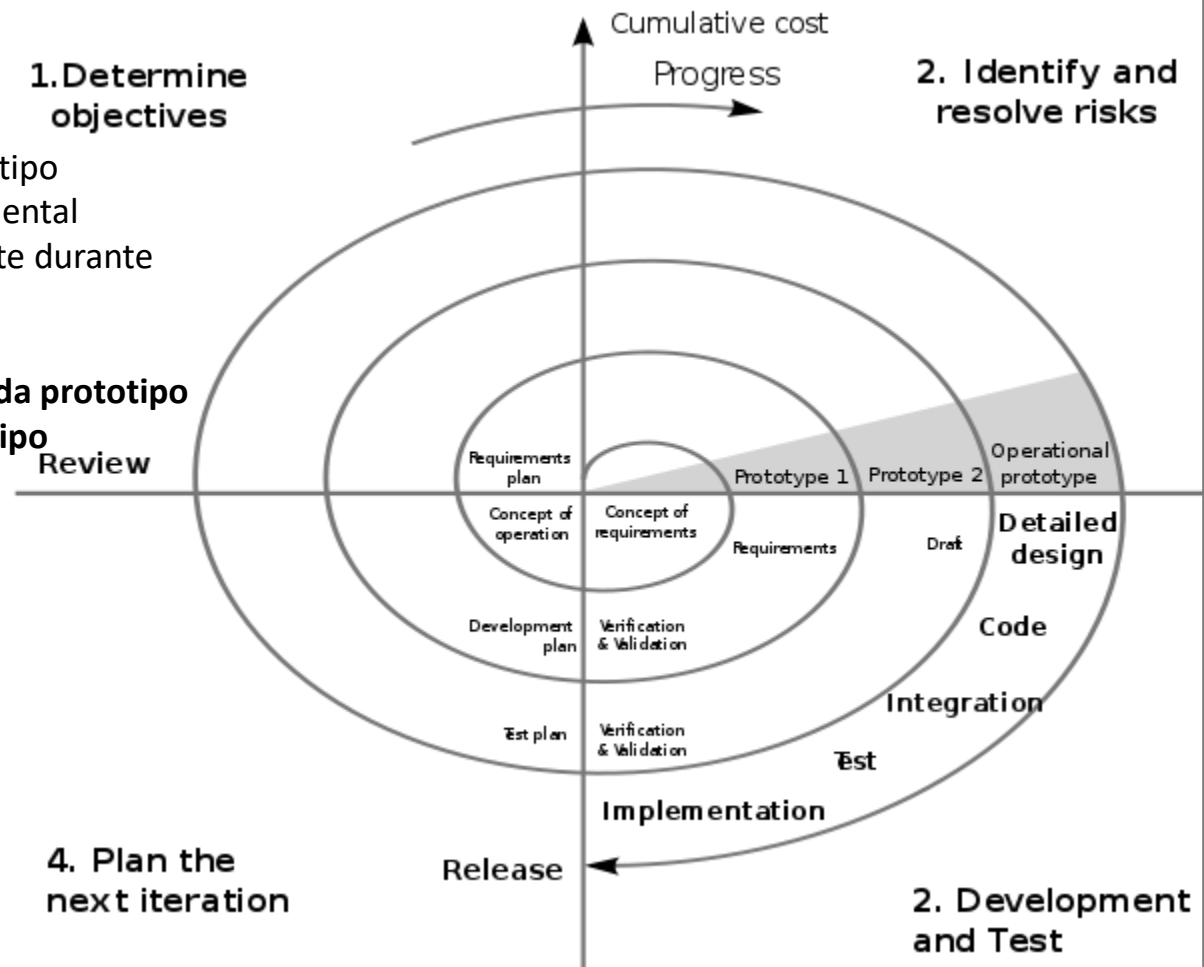


- Una fase no comienza hasta acabar la anterior
- No representativo de lo que realmente ocurre en un desarrollo software
- Gasta mucho tiempo al principio para evitar fallos en la recogida de requisitos.
- Producto no disponible hasta el final

5.1 Ciclos de vida

– Espiral (80s)

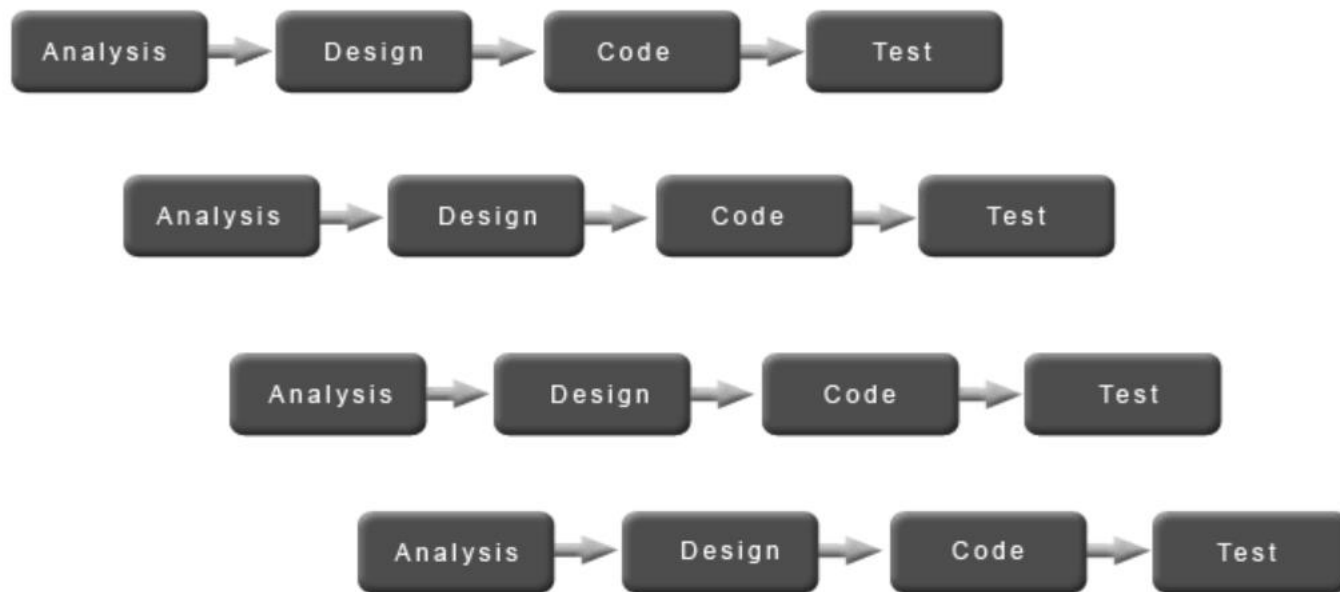
- De cada iteración resulta un prototipo
- Refinamiento del producto incremental
- Requiere la participación del cliente durante todo el ciclo de vida
- Más flexible a cambios
- Realiza **gestión de riesgos para cada prototipo**
- Solo es funcional el último prototipo**



5.1 Ciclos de vida

– Iterativo e Incremental (80s)

- El producto se construye por entregas
- Parecido al modelo de espiral, pero
 - sin gestión de riesgos
 - cada entrega es *algo* más funcional



Alistair Cockburn:

'Incrementar' significa añadir funciones al sistema; **iterar** significa repetir trabajo para mejorar el sistema'

- 5.1 Ciclos de vida**
- 5.2 Metodologías**
- 5.3 Organización Estructural**
 - 5.3.1 Equipo
 - 5.3.2 Interesados
- 5.4 Arquetipos de organización empresarial**

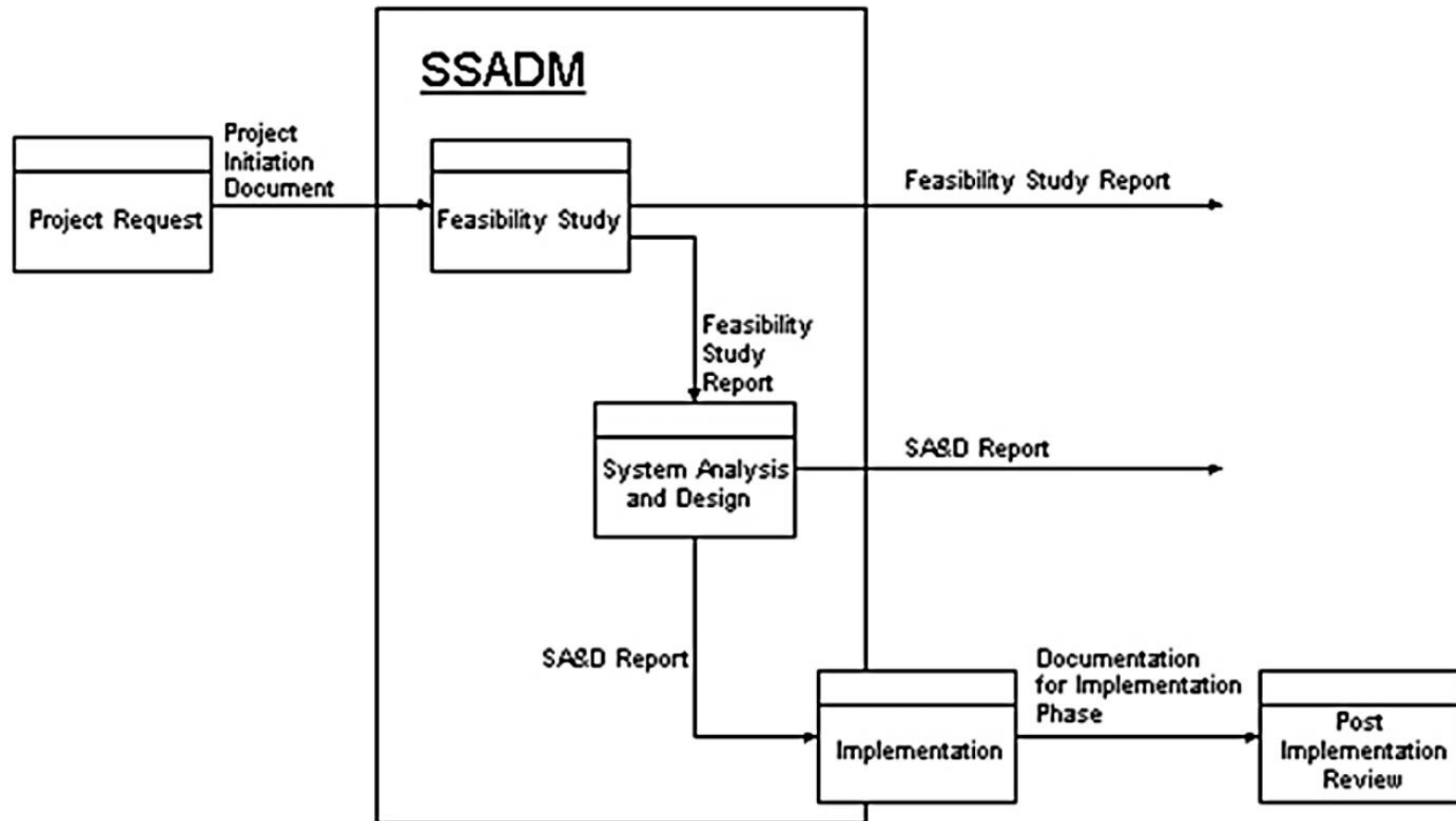
5.2 Metodologías

- La taxonomía a más alto nivel que puede hacerse entre metodologías es:
 - **Plan-driven o clásicas:** metodologías en las que se planifica todo antes de realizar el trabajo. Pueden incluir procesos para gestionar cambios, pero pueden resultar muy costosos.
 - **Change-driven o ágiles:** aquellas que planifican el trabajo a corto plazo, teniendo en cuenta que puede haber cambios en cada iteración.

PLAN-DRIVEN

- **Structured System Analysis and Design Method (SSADM) (80s)**
 - Ciclo de vida de cascada con fases un poco superpuestas
 - Metodología **para las fases de análisis de requisitos y diseño del sistema.**
 - No contiene especificaciones para desarrollo ni pruebas.
 - Estándar abierto y gratuito. Disponibles herramientas CASE.
 - Utilizado en departamentos de estado de Reino Unido

5.2 Metodologías: SSDAM

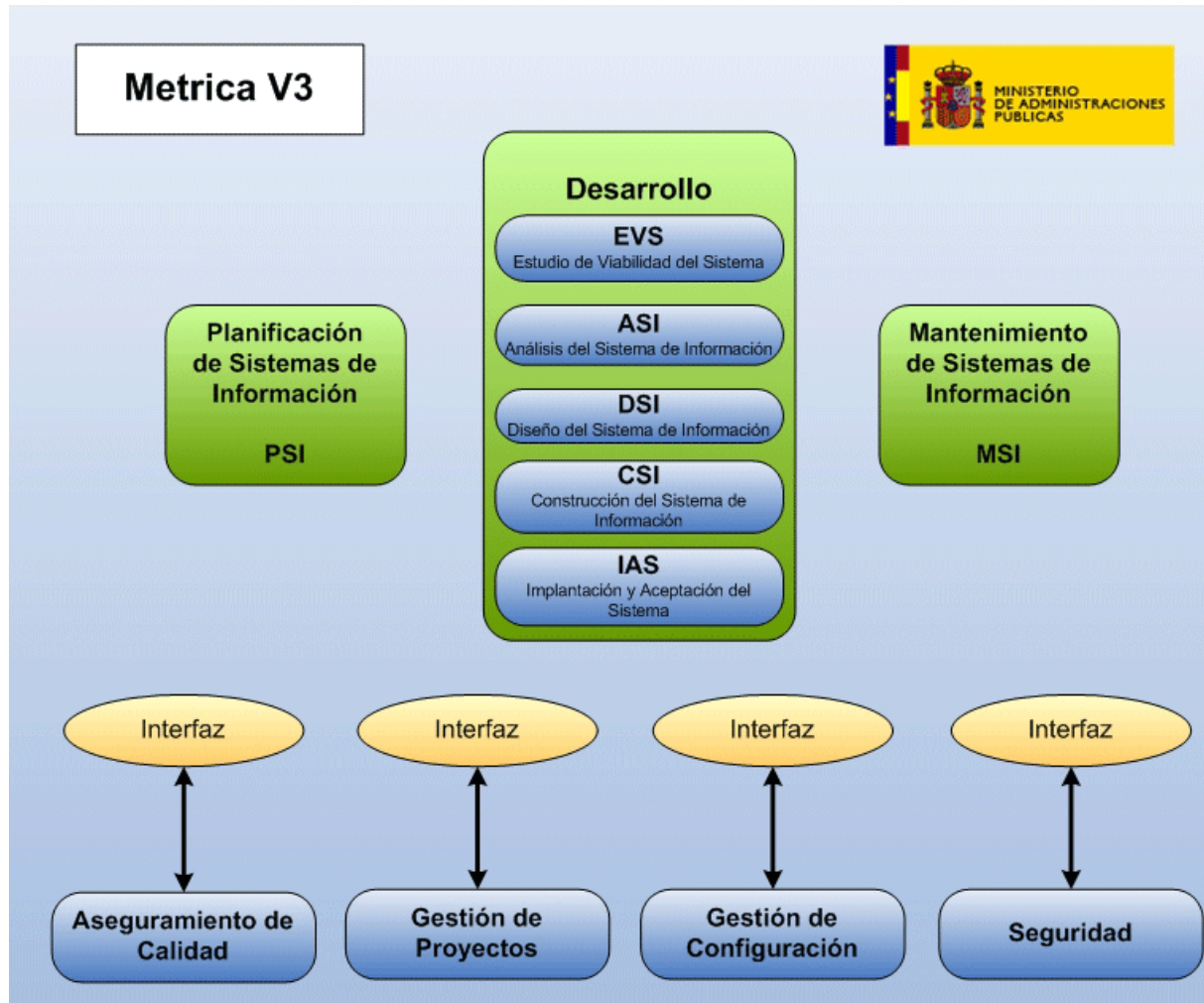


5.2 Metodologías: Métrica

– METRICA v3 (00s)

- Ciclo de vida cierta similitud a cascada
- Proyectos públicos en España para desarrollar Sistemas de Información.
- Estructura de Procesos: Planificación, Desarrollo y Mantenimiento. Las salidas de un proceso se utilizan como entradas de otros procesos.
- Gran importancia a la captura de requisitos.
- Gestión de **planificación, coste y calidad** a lo largo de todo el ciclo de vida.
- Múltiples herramientas CASE.

5.2 Metodologías: Métrica



Ejemplo de entregables PSI

- Modelo de sistemas de información
- Arquitectura tecnológica
- Plan de mantenimiento

DESARROLLO

- Valoración de Riesgos (EVS)
- Coste/beneficio (EVS)
- Requisitos (ASI)
- Glosario de términos (ASI)
- Resultado de Pruebas (CSI)
- Producto software(CSI)
- Plan de Implantación (IAS)

MSI

- Catálogo peticiones de cambio
- Estudio de la petición
- Plan de cambio

5.2 Metodologías: RUP

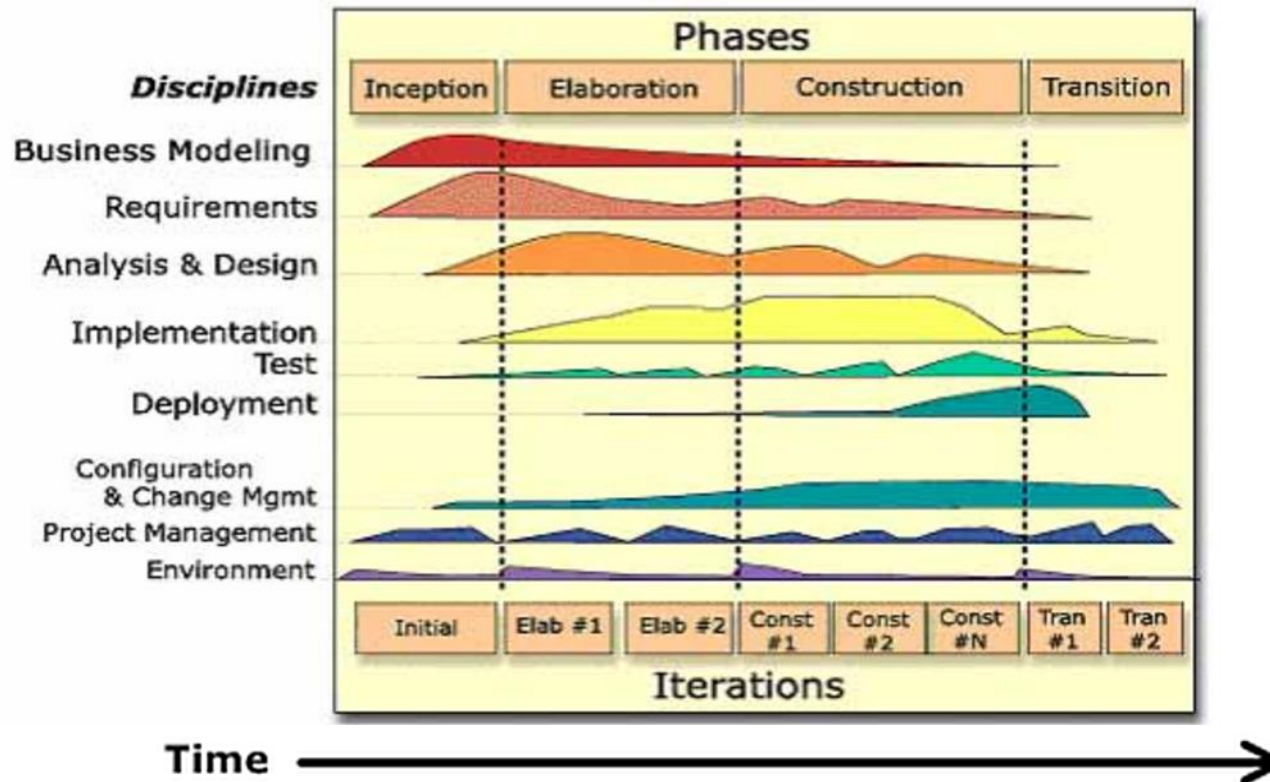
– Rational Unified Process, RUP (00s)

- Ciclo de vida iterativo con funcionalidad incremental.
- Proyectos largos
- Describe tanto los entregables a realizar como las habilidades necesarias
- Define papeles: jefe de proyecto, gestor de calidad, gestor de configuración,...
- Cada fase finaliza con un hito o entrega
- Fases superpuestas:
 - Inicio: descripción y evaluación del proyecto, riesgos, calidad e interesados.
 - Elaboración: arquitectura del sistema
 - Construcción: implementación
 - Transición: pruebas, entrenamiento y chequeo de calidad.

5.2 Metodologías: RUP

Ejemplos de entregables:

- Requisitos del producto (Inicio)
- Casos de uso (Elaboración)
- Código (Construcción)
- Manual de instrucciones(Transición)



5.2 Metodologías - ágil

METODOLOGÍAS CHANGE-DRIVEN o ÁGILES

- **Manifiesto Ágil:** En 2001, 17 de las personas más influyentes en el desarrollo software se reunieron para intentar dar unos principios formales que alejan el desarrollo tradicional de cualquier ingeniería del desarrollo software, el cual es influenciado principalmente por el **factor humano** del equipo y el carácter cambiante y lleno de **incertidumbre** del cliente.

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.*

Cuidado: se suele **malinterpretar** como que la parte derecha no es importante.

5.2 Metodologías - ágil

Las metodologías ágiles ya habían sido propuestas antes del manifiesto, con ciclos de vida iterativos e incrementales. Pero hacía falta un pistoletazo de salida. A partir de 2001 el uso de ágil y publicaciones relacionadas ha crecido exponencialmente. Ojo, **no se trata de ágil vs clásicas**.



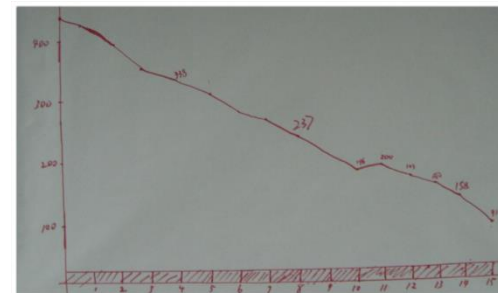
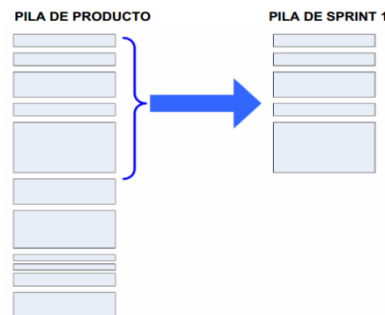
© netmind, We Know IT, KnowIT y OfficeMind son marcas registradas por netmind, sl



**Conferencia sobre la certificación Agile
Certified Practitioner de PMI**

5.2 Metodologías: Scrum

- Para equipos pequeños
- Si el equipo es grande o distribuido, meetings por video conferencia.
- Backlog + *sprints* incrementales
- **Sprint**: fase de 1 a 4 semanas de programación y reuniones diarias (15 minutos de pie) para añadir valor a la última versión del producto.
- Antes de cada sprint, Product Owner se reúne y reordena las tareas del backlog según el valor que aportan al cliente.
- El equipo se auto-gestiona para lograr el objetivo planteado para cada sprint.
- Durante el sprint, **no se permiten cambios**.
- La **fecha de cierre** de un sprint es inalterable.
- **No gestiona la calidad ni riesgos**
- **Artefactos**: Backlog del producto, backlog del sprint, Burndown chart.



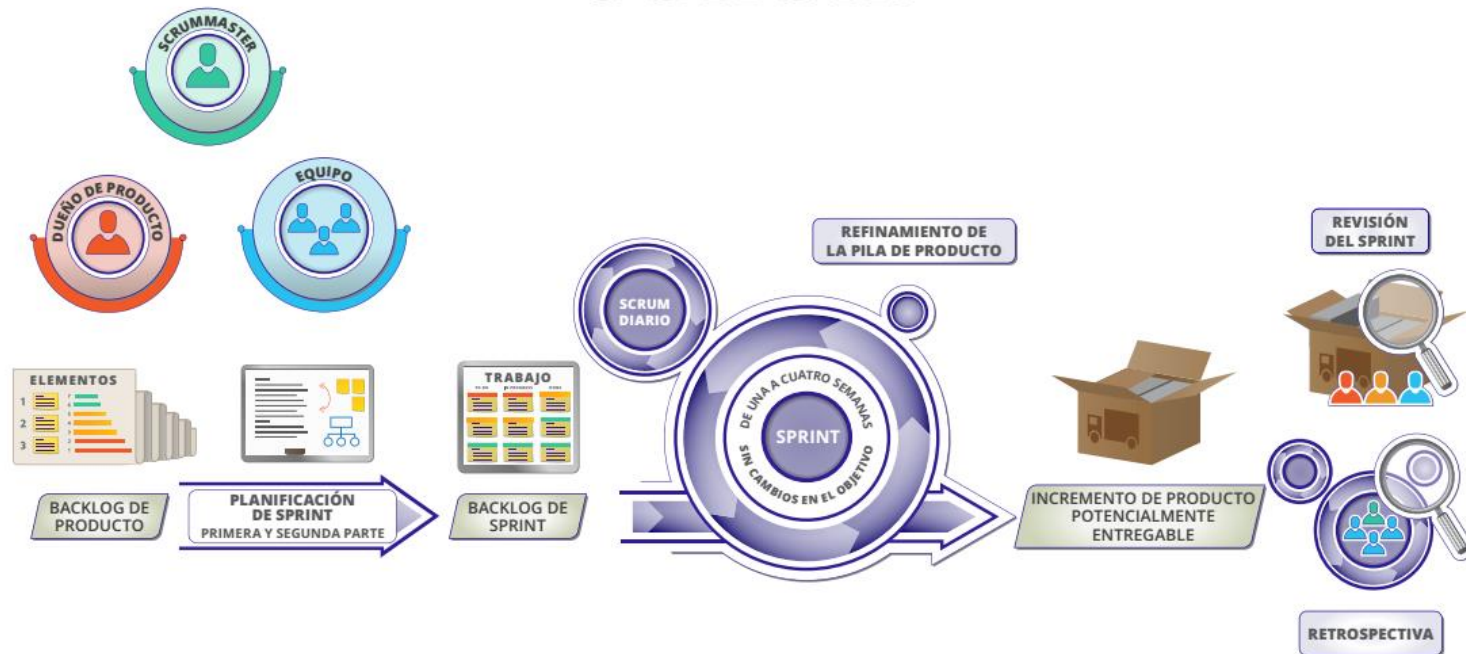
5.2 Metodologías: Scrum

- Roles:
 - **Scrum Master:**
 - Es el coach de la metodología Scrum.
 - Dirige el scrum meeting diario y la revisión del sprint.
 - También programa.
 - **Product Owner:**
 - Miembro del equipo que crea una Visión de lo que más aporta al cliente.
 - Responsable de crear, priorizar y ACTUALIZAR el Product Backlog.
 - Revisar junto a los interesados el incremento realizado al sistema tras el sprint.
 - **Equipo desarrollador**
 - Entre 5 y 9 programadores.
 - Auto-organizativos.
 - Pueden tomar las decisiones que sean necesarias para conseguir el objetivo del sprint (salvo cambiar el objetivo o su fecha).
 - Responsable de la selección de historias a incluir en el siguiente sprint.

5.2 Metodologías: Scrum

SCRUM

[Scrum Primer v5.0]



*'Caminante, son tus huellas
el camino y nada más;
Caminante, no hay camino,
se hace camino al andar.'*

A. Machado

5.2 Metodologías: Scrum

STATE OF SCRUM 2017-2018

SCALING AND AGILE TRANSFORMATION

http://info.scrumalliance.org/State-of-Scrum-2017-18.html?utm_source=agile42&utm_medium=blog&utm_campaign=state-of-scrum-2017-2018

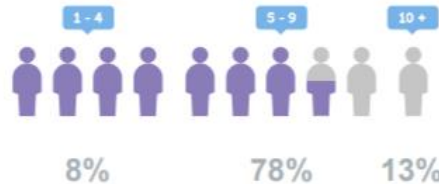
94% Respondents using Scrum in their Agile practice



78%
Use Scrum with other approaches

16%
Use Scrum exclusively

7.4 Average Scrum team size



2.4 Average length of a Sprint, in weeks

86% Hold a Sprint planning meeting prior to a Sprint

87% Have a daily Scrum meeting

2017 Top certification in 2017: ScrumMaster



85%
ScrumMaster



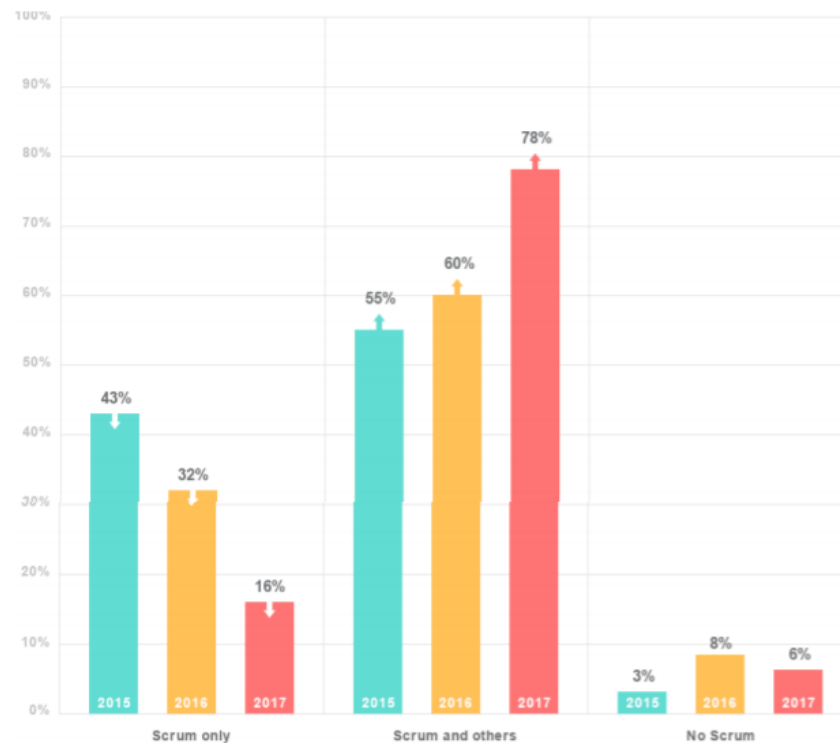
40%
Product Owner



15%
Scrum Teams

5.2 Metodologías: Scrum

AGILE APPROACHES



Other frameworks used

The average respondent reported using 3.5 frameworks

Kanban



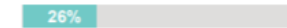
Hybrid (Waterfall and Agile)



Traditional Waterfall



Bimodal (Agile and Waterfall)



Lean



Watch this!

(Wrong) Explanation of Scrum Board - Silicon Valley S1x05

<https://www.youtube.com/watch?v=-nYdUDGcdM8>

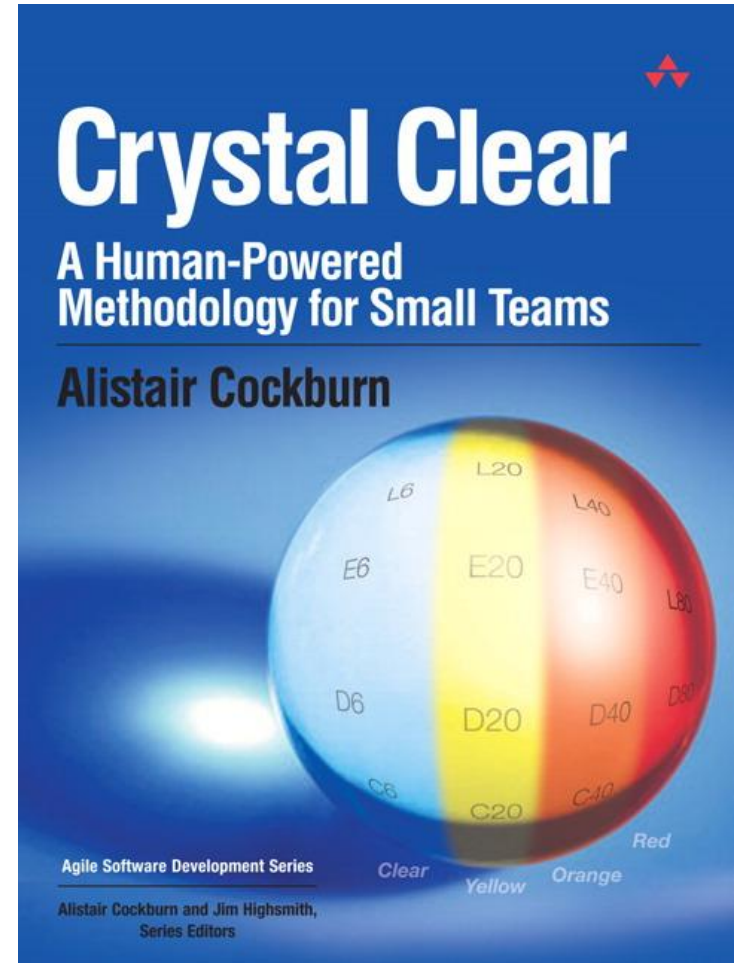
Podéis seguir a Scrum.org y ScrumAlliance en Twitter:

- @Scrumdotorg <https://twitter.com/Scrumdotorg>
- @ScrumAlliance <https://twitter.com/ScrumAlliance>



5.2 Metodologías: Crystal

- Crystal es una familia de metodologías ágiles, definidas por Alistair Cockburn en 1995. En 2004 publica Crystal Clear.
- Según el **tamaño del equipo** y el **impacto** que un fallo en el sistema puede tener en el contexto en el que se ejecuta, se proponen diferentes metodologías, siendo Crystal Clear la más ágil, y Crystal Red la más parecida a una metodología *plan-driven*.
- La más definida es Crystal Clear, y después Crystal Orange (para proyectos de 2 años) en su libro *Surviving Object-Oriented Project*. El resto están propuestas pero el autor aún no ha realizado especificaciones, y parece que han quedado en el aire.
- Todas **tienen en común**: entregas frecuentes y comunicación cercana que concluyan en mejoras del producto.



5.2 Metodologías: Crystal

- El autor critica el uso personalizado que permite RUP. Así, indica que si al seleccionar una metodología Crystal se comprueba que requiere otra más fuerte, deben ejecutarse todos los procesos y artefactos obligados por esta, sin puntos medios y sin plazos de tiempo adaptativos.

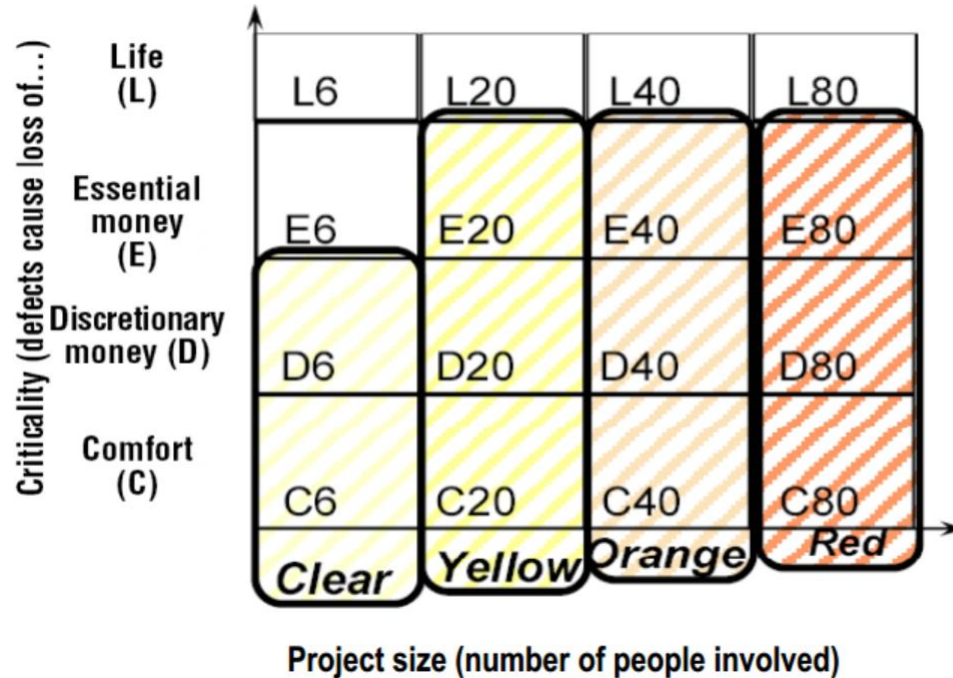


Imagen de [Cockburn, 2004].

5.2 Metodologías: Crystal

Crystal Clear: 3 a 8 personas con impacto de incomodidad o un poco de dinero.

- Todos juntos tan cerca que la **comunicación es osmótica**: puedes escuchar a tus compañeros cuando discuten cualquier tema.
- Entregas cada 3 meses y con usuario involucrado
- **Artefactos**: cronograma de revisiones, bocetos de diseño, casos de usos y modelo de objetos informales, casos de prueba y manual de usuario.
- **Roles**: cliente, diseñador/programador, usuario (puede ser a tiempo parcial).
- El **progreso** se mide en software entregado, no por artefactos completados.

Crystal Orange: 20 a 40 personas, con impacto muy caro en caso de fallos.

- Añade más seriedad a los artefactos y alarga los plazos de entregas.
- Todos los roles clásicos: PM, analista, arquitectos, diseñador de la GUI, programador, tester, cliente, usuario,...

5.2 Metodologías: XP

– Extreme Programming , XP (90s)

- Prioridad a dar respuesta rápida a cambios del cliente
- La documentación se realiza en forma de “comunicación oral” y código entregado.
- Todos los miembros del equipo son iguales (en jerarquía, pero sí hay roles)
- Programación durante corto período de tiempo + pruebas
- Código de alta calidad:
 - Desarrollo **test-driven**: tests unitarios cada pocas líneas, y test de aceptación al acabar una historia.
 - Programación en parejas
 - Refactorización
- XP consigue más productividad que Crystal Clear, pero es más difícil de cumplir para el equipo.



5.2 Metodologías: XP

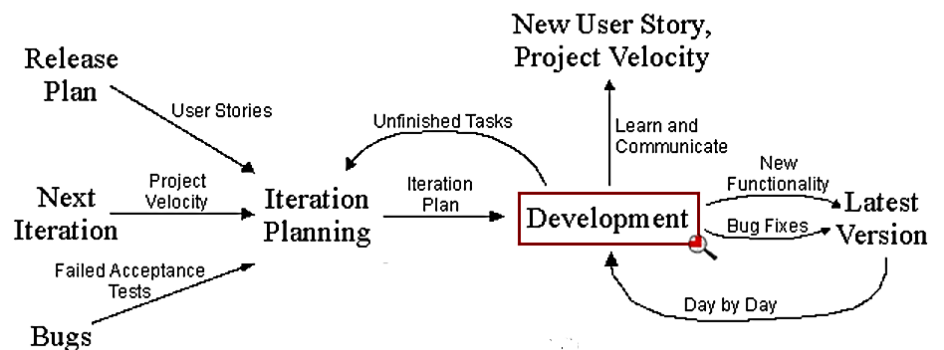
Se asume que los requisitos del cliente cambiarán en cualquier momento, y se reflejan en **user story cards**. La estimación de cada historia de usuario no debe exceder la **duración** de la iteración.

Extreme Programming Planning/Feedback Loops



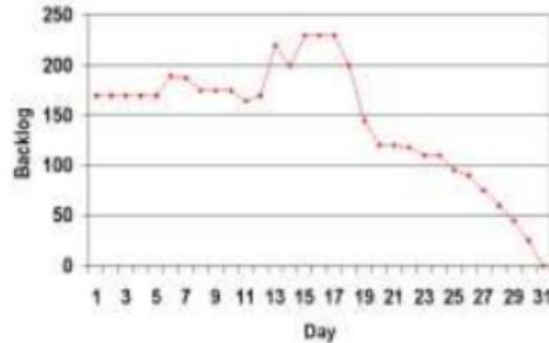
© J. Donovan Wells

Iteration



5.2 Metodologías: XP

Gráficos en la pared



Historia de usuario

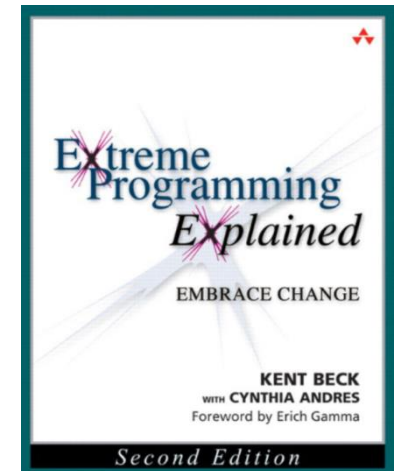
Title: Delete a Recipe

As the owner, I would like to be able to delete any of the existing coffee recipes so I can remove unpopular coffee types to make room for other selections.

Programar en parejas



Seleccionar tareas



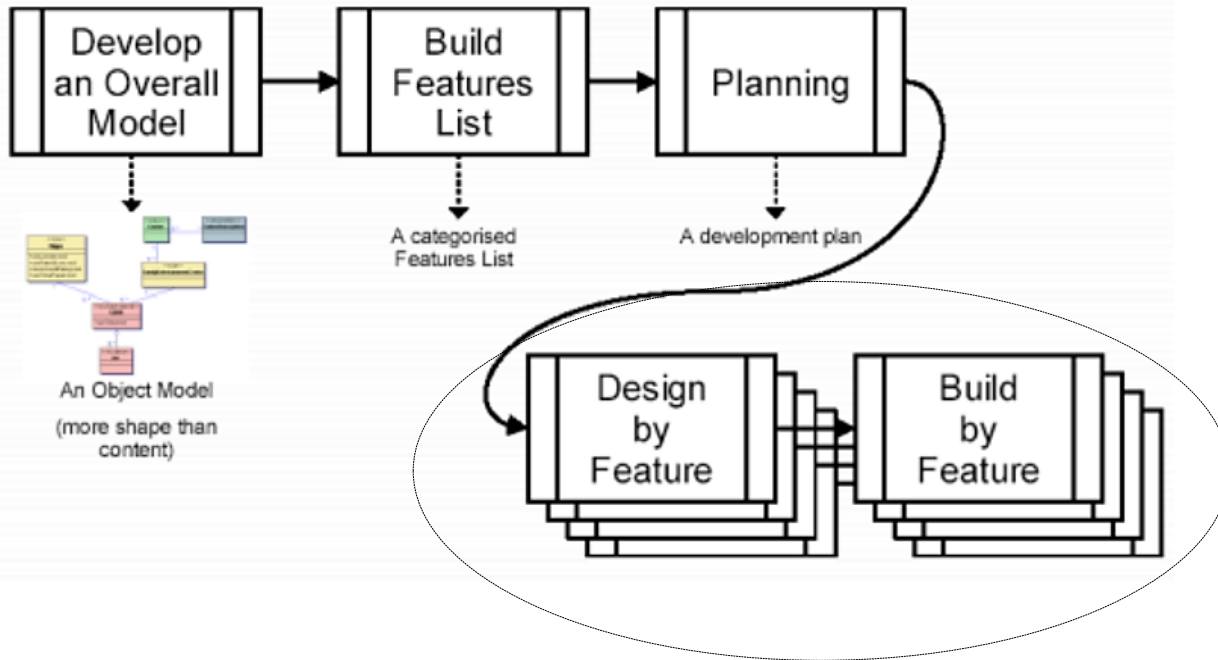
5.2 Metodologías: FDD

Feature-Driven Development (FDD)

- Jeff de Luca y Peter Coad, 1997.
- Se caracteriza por la abundancia de modelos UML y diseño+desarrollo centrado en características.
- Sugiere 8 buenas prácticas centradas en tener gente preparada y expertos en cada dominio.
- Artefactos:
 - **Listas de características:** un grupo de características son la descomposición de un requisito funcional de forma que cada característica puede implementarse en 1 ó 2 semanas.
 - **Paquetes de Diseño:** diagramas de clases y secuencia.
 - **Rastreo de Característica:** tabla que cruza las características y la fecha en que se han completado.
 - **Tabla “Burn-up”:** como la burn-down chart transponiendo los ejes.

5.2 Metodologías: FDD

Se definen 5 procesos: los 3 primeros procesos se repiten de forma iterativa



Procesos incrementales de 2 semanas

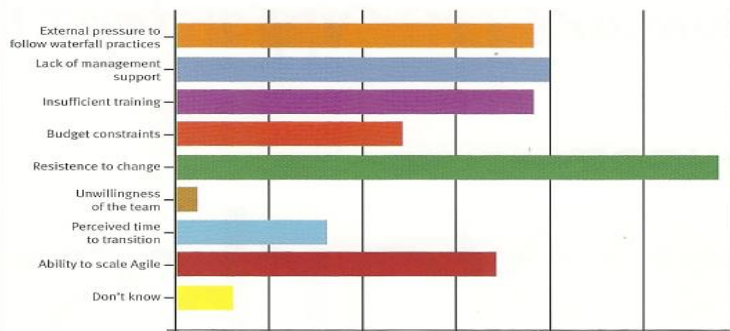
Proceso	Tiempo	
	1ª it	resto
1	10%	4%
2	4%	1%
3	2%	2%
4	34%	
5	43%	

Imagen: [Laurie Williams, 2009]

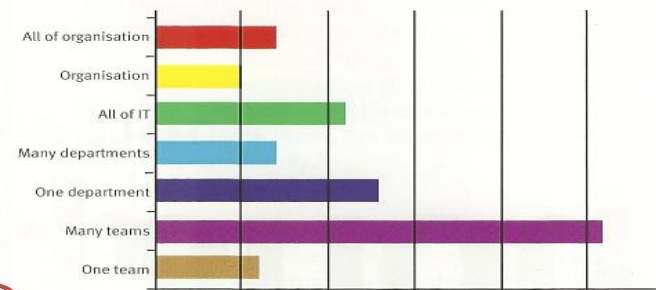
State of the (Agile) Nation 2012

At the Agile Australia 2012 Conference, delegates were asked to complete a brief survey about their Agile experience. Here are the results:

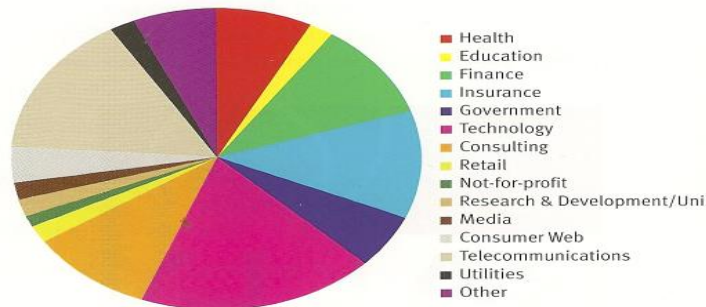
What are the biggest challenges for successfully utilising Agile projects in your organisation?



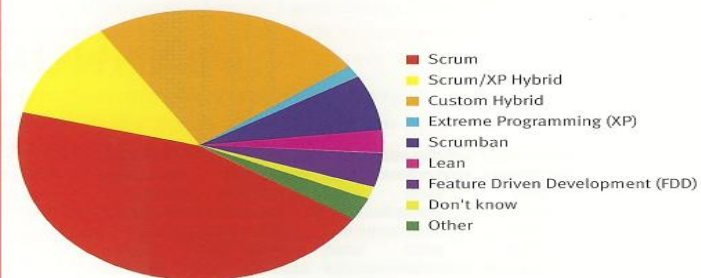
How widespread is the use of Agile in your organisation?



Indicate which of the following sectors your organisation is part of:



What flavour of Agile is used most by your company?



Q: To what extent does your organization use standardized project management practices?

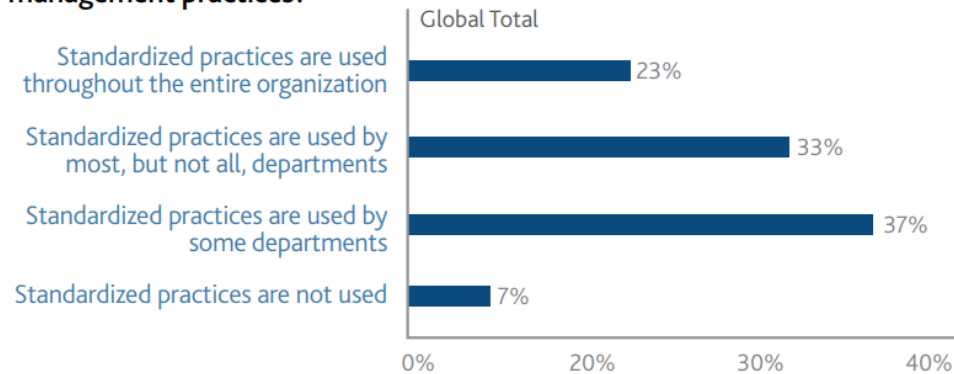


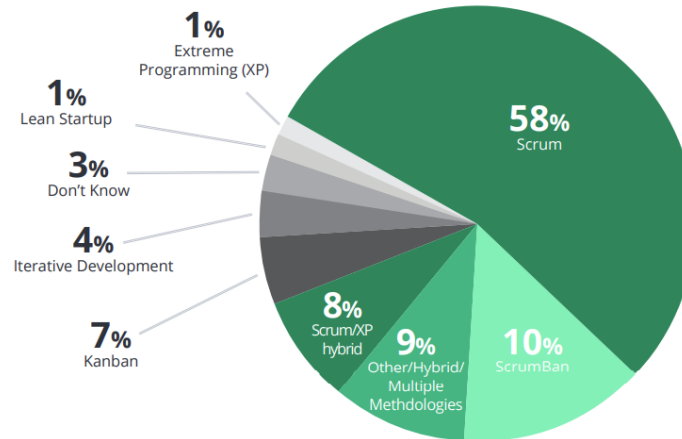
Figure 3: Champions Use the Approach Best Suited for Them

	Champions
Average percentage of projects using predictive approaches	44%
Average percentage of projects using agile approaches	30%
Average percentage of projects using hybrid approaches	23%
Average percentage of projects using "other" approaches	4%

5.2 Metodologías



2020
<https://stateofagile.com/>



Total exceeds 100% due to rounding.

Técnicas más utilizadas:

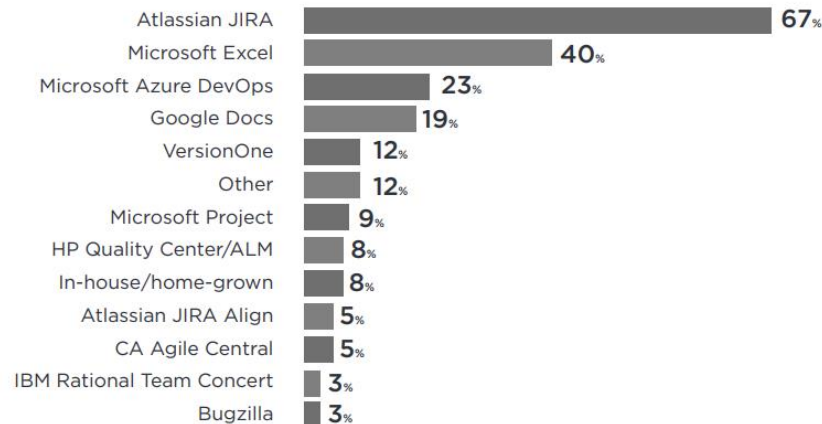
- Reunión diaria de pie 85%
- Retrospectiva 81%
- Planificación de las iteraciones 79%
- Revisión de la iteración 77%
- Kanban: 63%
- Planning Poker: 60%

Escalados más usados para gestionar más de 1 equipo:

- Scaled Agile Framework (SAFe) 35%
- Scrum/Scrum de Scrums 16%
- DAD (Disciplined Agile Delivery): 4%
- Less 4%
- Nexus 3%

USE OF AGILE PROJECT MANAGEMENT TOOLS

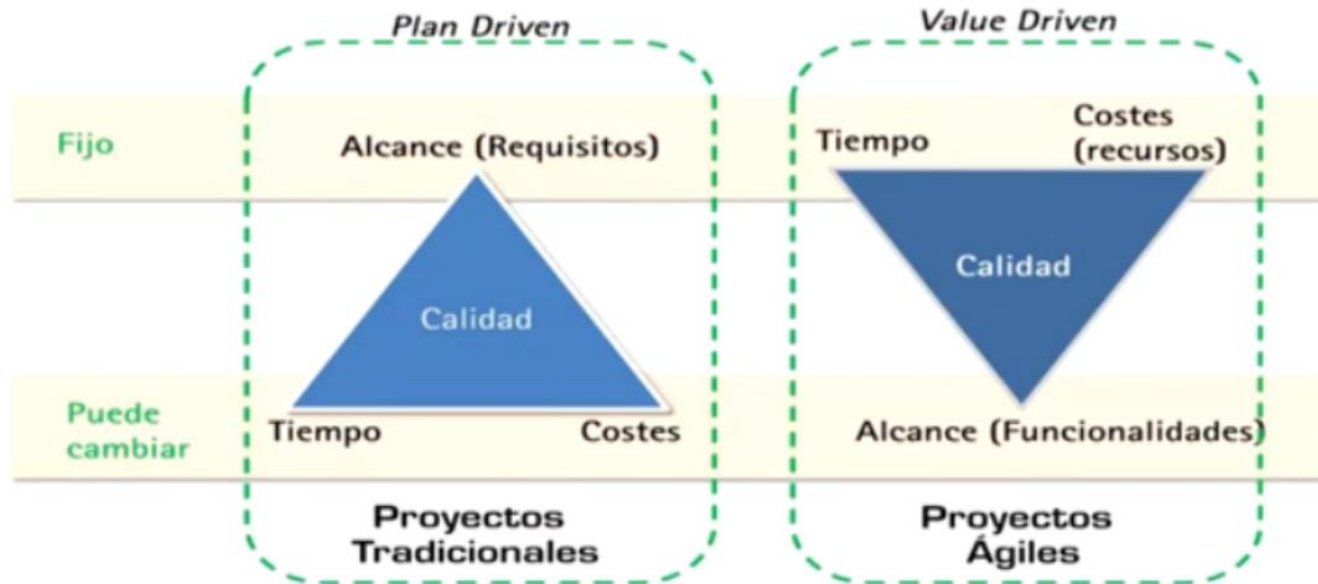
Respondents cited using a myriad of different tools to manage Agile projects.



5.2 Metodologías

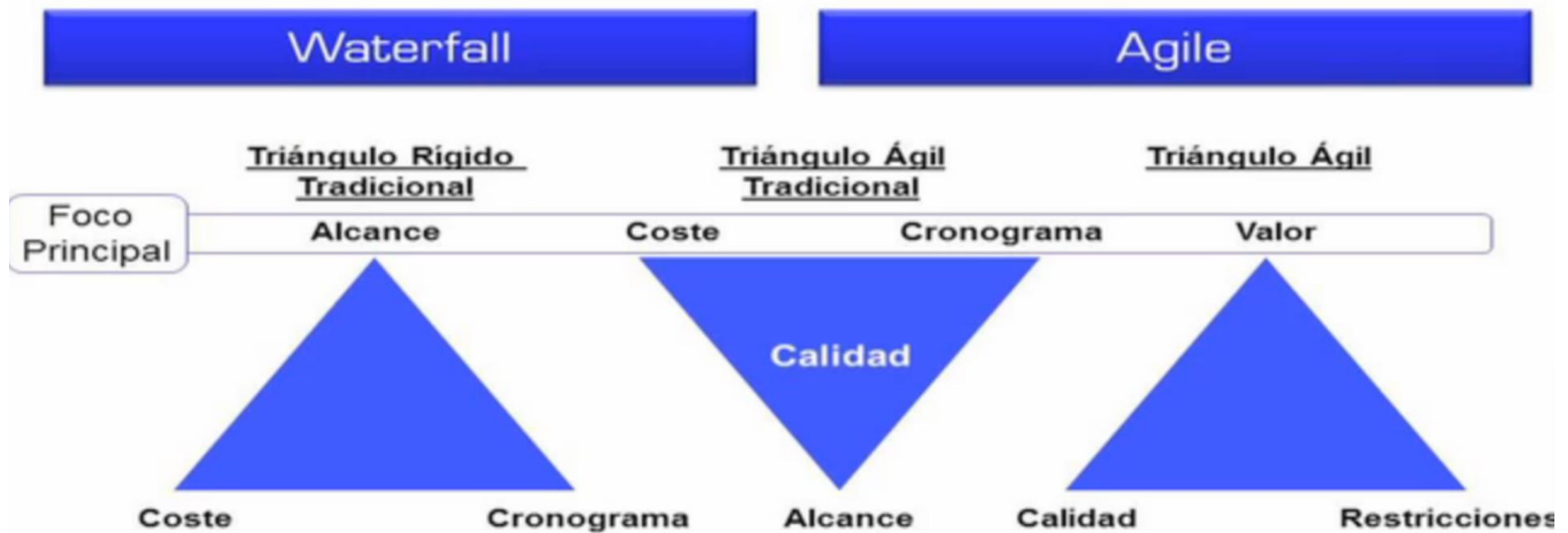
Proyectos Ágiles vs Tradicionales

"Triángulo de hierro"



Netmind

5.2 Metodologías



[Highsmith, 2010]

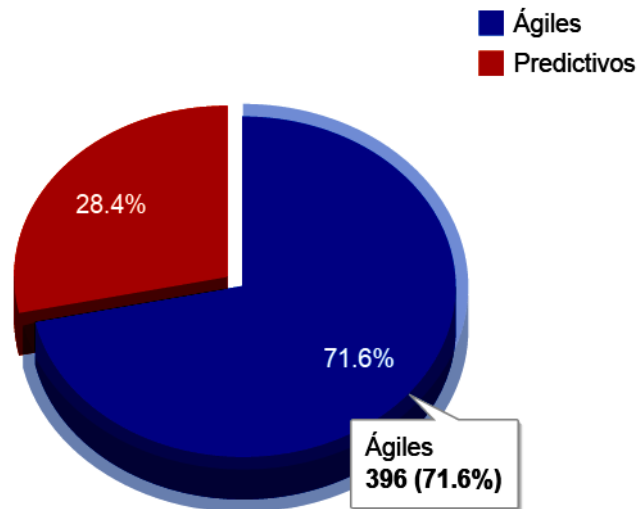
5.2 Metodologías

Encuesta Abril 2015.

Resultados: 157 vs 396

<http://www.laboratorioti.com/2015/05/06/somos-agiles-o-predictivos-descubrelo-en-el-resultado-de-la-encuesta-de-abril/>

¿Qué modelos de Gestión de Proyectos debemos usar?



5.2 Metodologías

Figura del Project Manager – Relación con roles Scrum

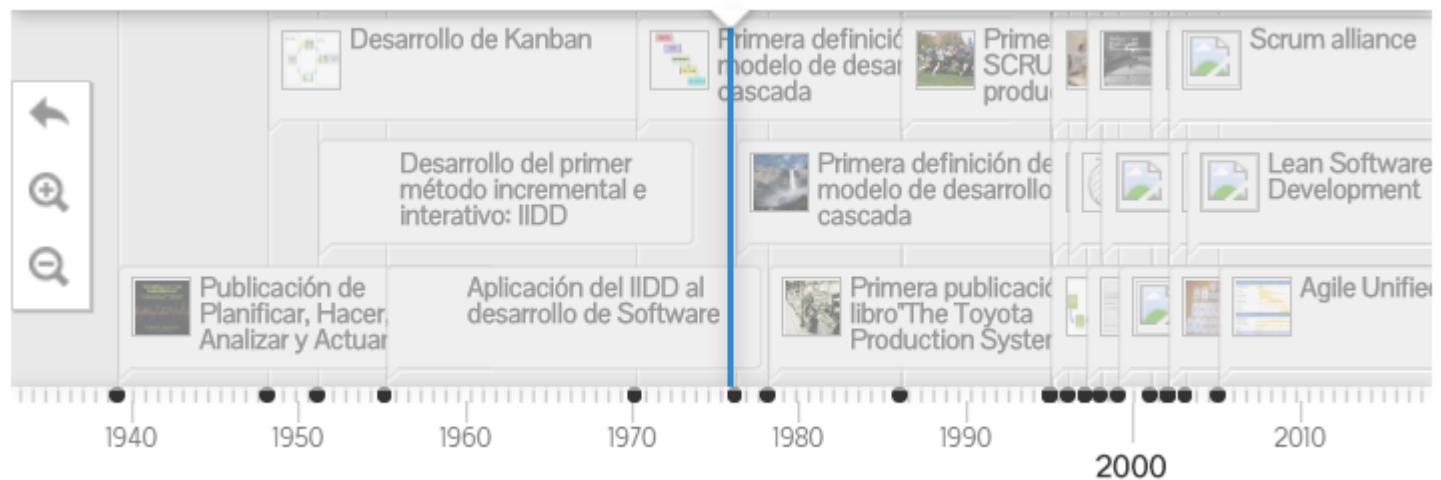


Netmind: El rol del Project Manager en entornos ágiles. Alfred Maeso.

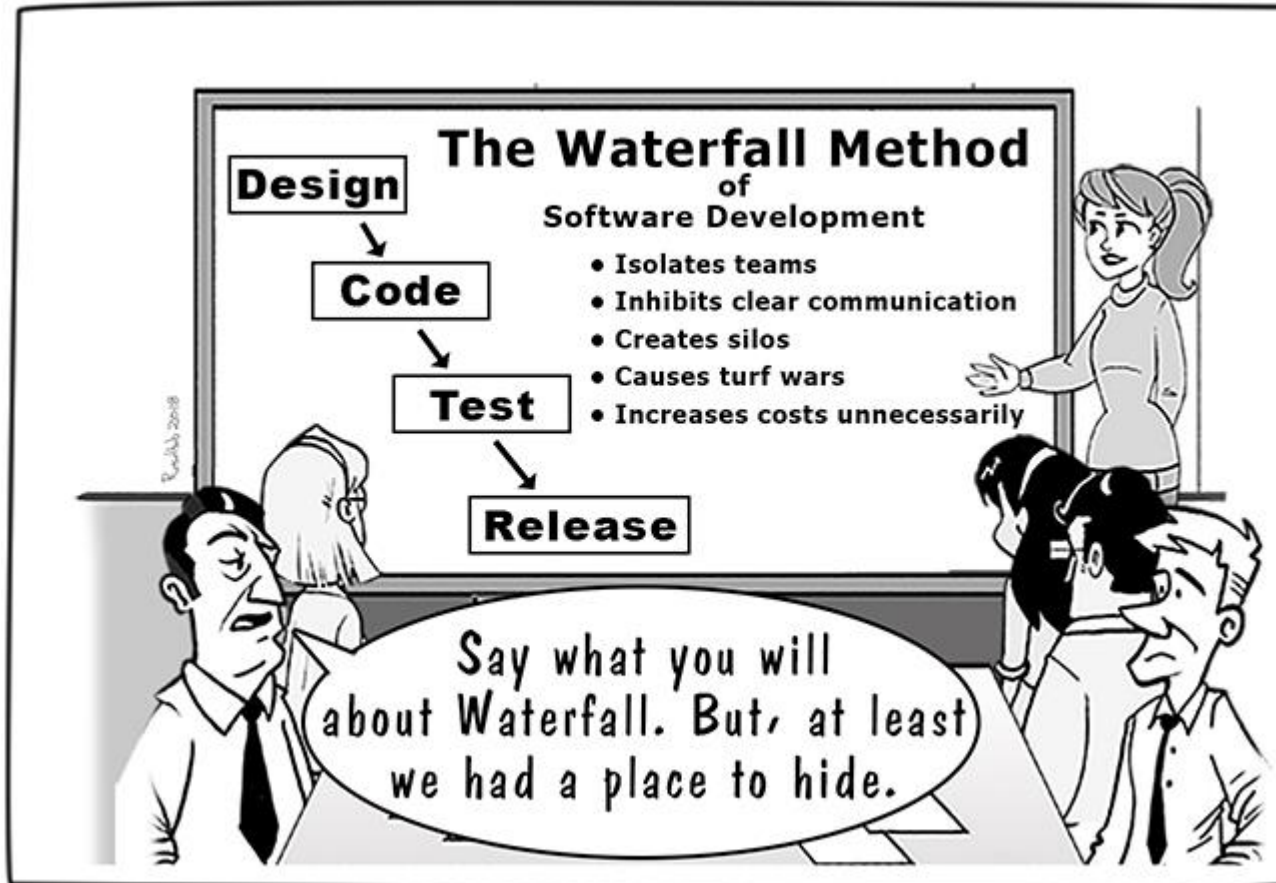
5.2 Metodologías

Saber más:

<http://www.laboratorioti.com/2014/02/17/historia-de-las-metodologias-agiles>



5.2 Metodologías

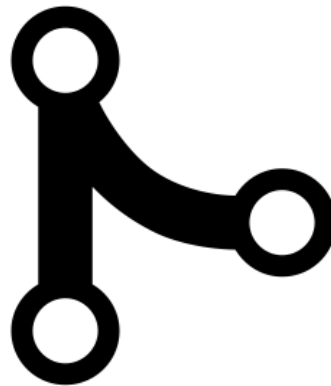


5.2 Metodologías

- Conclusiones :
 - El ciclo de vida describe el tipo de relación entre fases
 - La metodología es un conjunto muy definido de: fases, roles, entregables,...
 - El tipo de metodología depende del tipo de proyecto y grupo
 - No sientas como definitiva ninguna metodología.
 - La que hoy funciona, dentro de una década será *descabellado*.
 - Hay que conocer todos los tipos de fases y procesos que puedan existir, y ¡así estarás preparado!
 - ¡El **PM** no desaparece por usar una metodología ágil! Aunque **se convierte...**
- **No se trata de ágil sustituyendo a las plan-driven, sino de seleccionar según, en general:**
 - Incertidumbre: si de entrada hay mucha información podemos hacer un plan detallado
 - Si se estima mucha volatilidad de los requisitos: ágil
 - Si el equipo es muy grande: plan-driven, o escalamiento de Scrum
 - ... siendo todo lo anterior más o menos discutible conforme pasan los años...

5.2 Metodologías: Pull-based development

- No es una metodología en sí, más bien un flujo de trabajo para colaboradores que en verdad **no pertenecen al equipo** de trabajo de un **proyecto open-source** que se encuentra en un repositorio de control de versiones, como GitHub o BitBucket.
1. Cualquier programador que quiera ser colaborador externo a un repositorio, o simplemente hacer cambios para uso personal, primero desdoblará el fuente del proyecto haciéndose una copia personal (***fork***) sobre la que trabajar.

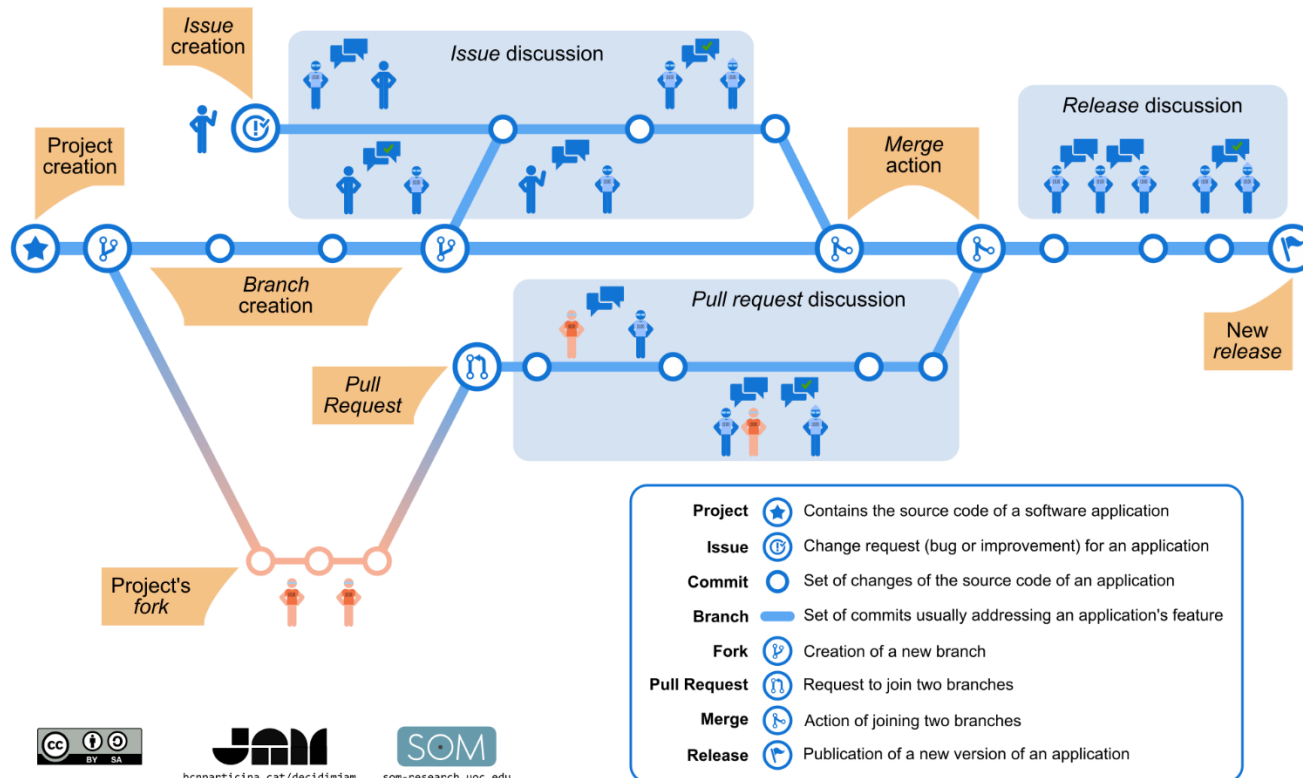


5.2 Metodologías: Pull-based development

2. El colaborador potencial trabajará en local y subirá a su repositorio personal los cambios deseados (**push**).
3. Cuando crea que sus cambios están completos y funcionan correctamente, envía una petición al responsable del repositorio original (**pull request**).
4. Dicho responsable asigna la revisión de dicha contribución a un miembro del equipo (*open*) que hará el papel de *integrator* y será quien decida si fusionar (*merged*) o no (*closed*) la contribución con la rama solicitada por el contribuidor (normalmente la principal).

5.2 Metodologías: Pull-based development

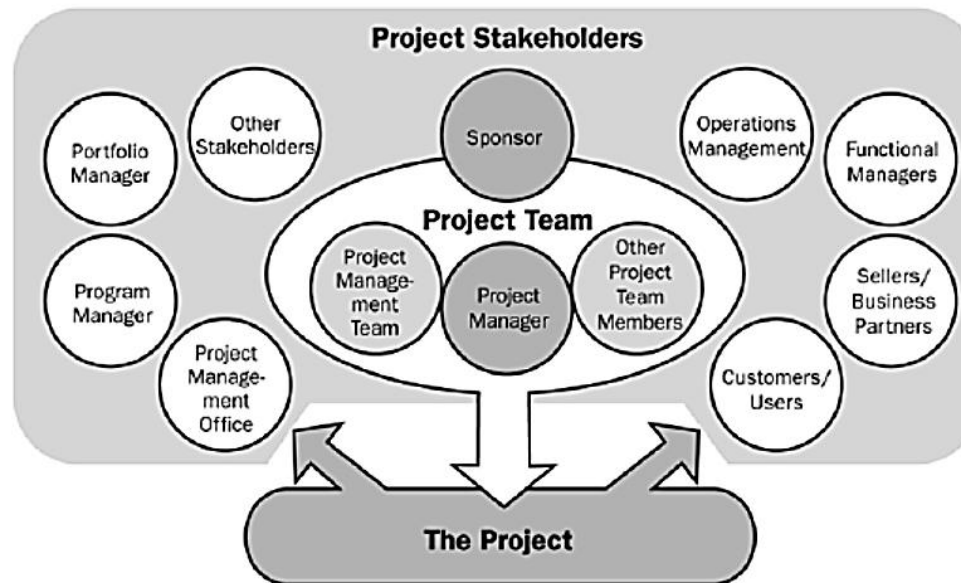
How GitHub projects are developed?
Where are the main discussion points?



- 5.1 Ciclos de vida**
- 5.2 Metodologías**
- 5.3 Organización Estructural**
 - 5.3.1 Equipo
 - 5.3.2 Interesados
- 5.4 Arquetipos de organización empresarial

5.3 Organización Estructural

- Independientemente de la metodología aplicada a nuestro proyecto, hay un conjunto de entidades que influyen directamente:
 - El *equipo de proyecto*, que será creado por el PM.
 - Los *interesados*, que serán identificados por el PM.



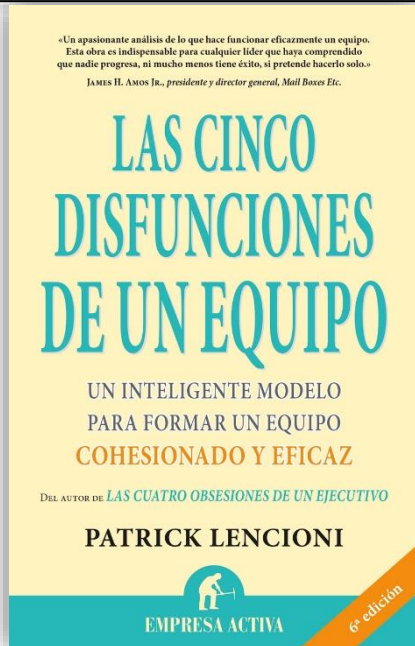
- 5.1 Ciclos de vida**
- 5.2 Metodologías**
- 5.3 Organización Estructural**
 - 5.3.1 Equipo**
 - 5.3.2 Interesados
- 5.4 Arquetipos de organización empresarial

5.3 Organización Estructural

- **El Equipo de Proyecto** debería ser creado y dirigido por el PM.
- El PM es seleccionado por el jefe de departamento, gestor de programa o gestor de portafolio, dependiendo del tipo de empresa.
- El PM debe procurar que su **equipo cumpla las siguientes cualidades**:
 - Cada individuo *competente* en su tarea.
 - Deben ser *comprometidos* con el proyecto. Es conveniente que sean personas con voluntad propia para la resolución de problemas inesperados que puedan aparecer, sin necesidad de que el PM lo ordene:
 - Programador enfermo
 - Máquina estropeada
 - Cronograma reajustado
 - *Confianza para ser sinceros*. Tienen que ser capaces de evitar la tentación de ocultar errores, por miedo a afectar al plan del proyecto. El PM debe evitar crear un ambiente hostil que favorezca esas situaciones.

‘Hard workers are easy to find. What is rare is an individual who is driven to finish the job’.

5.3.1 El Equipo



5.3.1 El Equipo

Deficiencia 1: Falta de confianza

- Los miembros del equipo deben confiar en las intenciones de los demás.
- Es necesario tener la confianza suficiente para mostrar nuestras vulnerabilidades sabiendo que no serán utilizadas contra nosotros:
 - Debilidades.
 - Deficiencias en capacidades.
 - Defectos en relaciones personales.
 - Errores y peticiones de ayuda.
- Cuando los miembros de un equipo se sienten cómodos estando expuestos unos a otros, comienzan a **esforzarse sin preocuparse de protegerse a sí mismos**.
- El director del equipo debe trabajar para construir confianza, lo cual lleva tiempo, y ser el primero en dar ejemplo de que se puede mostrar vulnerabilidad.



5.3.1 El Equipo

Deficiencia 2: Temor al conflicto

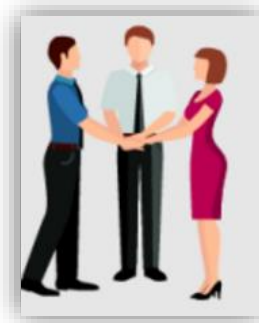
- Tener debates apasionados es algo esencial para que el equipo madure y tome decisiones.
- Al existir confianza, los integrantes que discuten saben que no es una lucha destructiva y personal, sino un conflicto ideológico y productivo.
- Los equipos que evitan el conflicto para no herir sentimientos:
 - Acaban creando tensión entre los miembros.
 - Vuelven una y otra vez a los mismos puntos.
- El director del equipo debe:
 - Dejar que la discusión discurra de manera natural hasta llegar a un acuerdo.
 - Recordar los beneficios del conflicto al observar que está incomodando a alguien.



5.3.1 El Equipo

Deficiencia 3: Falta de compromiso

- Los grandes equipos adoptan decisiones claras y permanentes con la aceptación de todo el equipo, incluso por aquellos miembros que votaron en contra.
- Hay 2 factores que pueden influir en el compromiso con una decisión tomada por el equipo:
 - *Necesidad de consenso*: no estar dispuesto a comprometerse si no se ha aprobado una decisión por consenso. Todas las ideas del equipo deben ser escuchadas y discutidas, y se impondrá la votada por mayoría, no por consenso.
 - *Necesidad de certidumbre*: aunque no haya seguridad de si la decisión es la correcta, debemos comprometernos en claros cursos de acción. *Una decisión es mejor que ninguna*.
- El director del equipo debe asegurarse de que se cumple el programa establecido.



5.3.1 El Equipo

Deficiencia 4: Evitación de responsabilidades (en el sentido de *accountability*)

- Ser responsable significa estar dispuesto a **pedir cuentas** a nuestros compañeros sobre desempeños o conductas que pueden estar perjudicando los objetivos del equipo.
- A veces se evita esta responsabilidad por miedo a dañar una relación personal; sin embargo, esto hace que la relación se deteriore.
- El medio más eficiente para mantener altos niveles de desempeño es el miedo a decepcionar a respetados compañeros de equipo.
- El director del equipo no debe ser la única fuente de responsabilidad, pero sí tendrá la última palabra.



Díselo

Fases del Proyecto y Organización



Escúchale

5.3.1 El Equipo

Deficiencia 5: Falta de atención a los resultados

- Hay que dirigir el esfuerzo en conseguir **objetivos globales** para el equipo o la empresa.
- Esto no se consigue cuando nos centramos en conseguir objetivos personales, como un cambio de rol o ascenso, no hacer algo que beneficie a alguien en concreto, o simplemente trabajar para sobrevivir en la empresa.
- Los miembros de un gran equipo siempre anteponen los intereses del equipo ante los personales.
- Si los miembros del equipo notan que el director valora algo antes que a los resultados globales, se lo tomarán como un permiso para hacer lo mismo.



Fases del Proyecto y Organización



5.3.1 El Equipo



**"It is teamwork that remains the
ultimate competitive advantage,**

**BOTH BECAUSE IT IS SO POWERFUL
AND SO RARE."**

Patrick Lencioni

- 5.1 Ciclos de vida**
- 5.2 Metodologías**
- 5.3 Organización Estructural**
 - 5.3.1 Equipo**
 - 5.3.2 Interesados**
- 5.4 Arquetipos de organización empresarial**

5.3.2 Los interesados

- **Los interesados** pueden ser personas, departamentos o empresas. Su relación con el proyecto es una de estas tres:
 - Participan en el proyecto
 - El éxito del proyecto les afecta positivamente
 - El éxito del proyecto les afecta negativamente (*interesados negativos*)
- Algunas veces los intereses de los interesados son contradictorios, y el PM deberá saber balancearlos
- Pueden ser interesados:
 - Cliente y usuario final
 - Patrocinador
 - Director de portafolio y programa – Oficina de Gestión de Proyectos
 - Equipo de proyecto

5.3.2 Los interesados

- Algunos interesados tendrán *autoridad* sobre el PM, otros *responsabilidad* dentro del proyecto, y otros simplemente *interés*. TODOS DEBEN SER IDENTIFICADOS Y CONTROLADOS.
 - Encuestas al usuario final
 - Reuniones con los analistas
 - Informe de presupuesto al jefe de departamento
 - Departamento o límites jurídico

- 5.1 Ciclos de vida**
- 5.2 Metodologías**
- 5.3 Organización Estructural**
 - 5.3.1 Equipo**
 - 5.3.2 Interesados**
- 5.4 Arquetipos de organización empresarial**

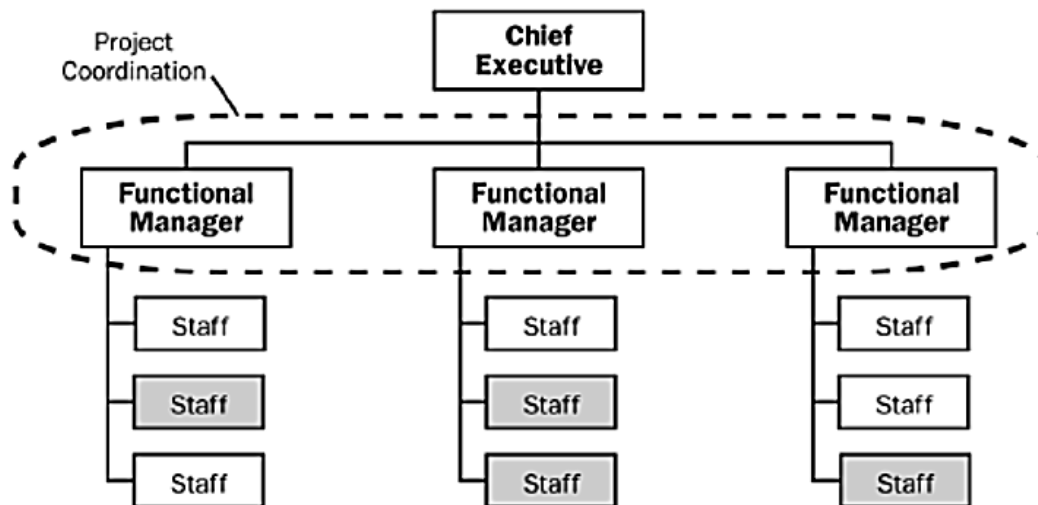
5.4 Arquetipos de organización empresarial

- Las **influencias** dentro de una empresa **puede variar** mucho **de la oficial a la real**:
 - Culturas únicas en cada empresa:
 - Normas no escritas
 - Tipo de relación correcta con las autoridades
 - Procedimientos a seguir
 - Las autoridades reales no son siempre las oficiales. Como PM o empleado, si eres nuevo en la empresa necesitarás identificar rápidamente quién tiene autoridad sobre tu proyecto.
- Aparte de conocer cómo fluyen las influencias, es necesario ser consciente del tipo de **organización de los departamentos**, respecto a la gestión de proyectos [PMBOK,6Ed]
 - Funcional
 - Matricial
 - Orientada a proyectos o mercado

5.4 Arquetipos de organización empresarial

- **Empresa Funcional**

- ✓ – Cada empleado tiene un *superior claro*
- ✓ – *Acceso directo* a los recursos internos.
- ✓ – Resulta *barato* para la empresa ya que el personal es contratado a tiempo parcial o, por lo menos, no están destinados únicamente al desarrollo de proyectos.



(Gray boxes represent staff engaged in project activities.)

5.4 Arquetipos de organización empresarial

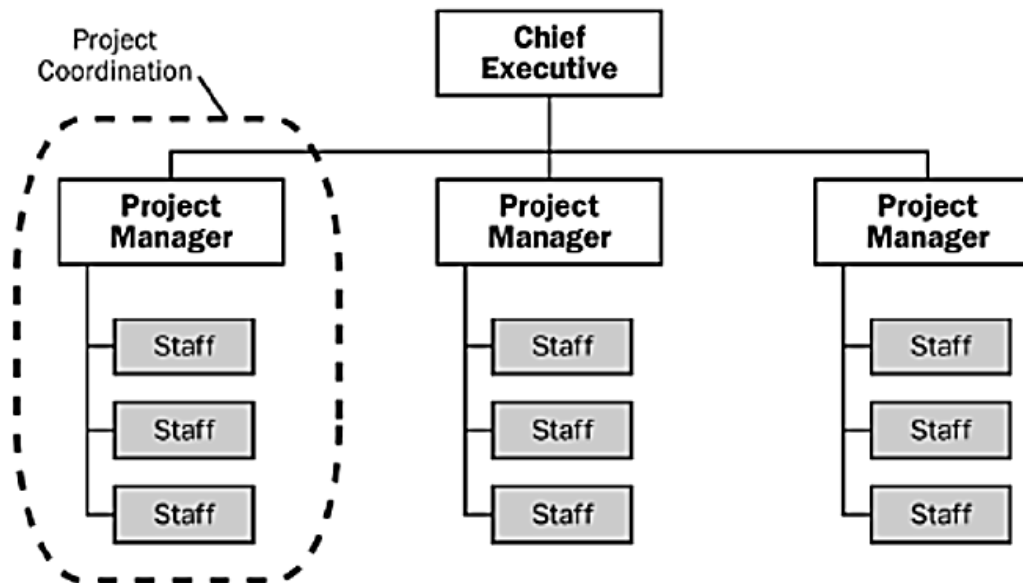
(...funcional)

- ✗ – Cada departamento desarrolla su trabajo correspondiente del proyecto de forma *independiente* a los otros departamentos:
 - Problemas al necesitar asistencia: técnica, programadores, renegociación de presupuesto,...
 - Profundidad tecnológica, pero no extendida.
- ✗ – La *comunicación* interdepartamental es extremadamente lenta, y generalmente ha de pasar por el jefe de departamento:
 - Retrasos en el proyecto
 - Comunicación con el cliente más lenta
- ✗ – El proyecto *no es prioritario* en ninguno de los departamentos en los que se reparte.

5.4 Arquetipos de organización empresarial

- **Empresa Orientada a Proyectos o al Mercado**

- ✓ – Los miembros del equipo siguen una *misma distribución* física y lógica
- ✓ – Los *recursos* de la empresa están dirigidos al desarrollo de los proyectos
- ✓ – *Especialista* de cada área dentro del proyecto



(Gray boxes represent staff engaged in project activities.)

5.4 Arquetipos de organización empresarial

- (...orientada a proyectos o mercado)
 - ✗— *Cuando el proyecto acaba*, el equipo vuelve a su ubicación a la espera de otro proyecto:
 - El gestor de presupuesto no es necesario en todo momento
 - Poco trabajo frustra tanto como el exceso
 - Esta organización es útil para grandes proyectos o empresas con tráfico continuo de proyectos
 - ✗— Pocos especialistas y *profundidad tecnológica baja*: en ocasiones es necesario realizar *outsourcing* o intentar salvar la falta:
 - Cuando un especialista asignado no es competente
 - Cuando toda la empresa está envuelta en proyectos hace falta contratar temporalmente recursos humanos externos

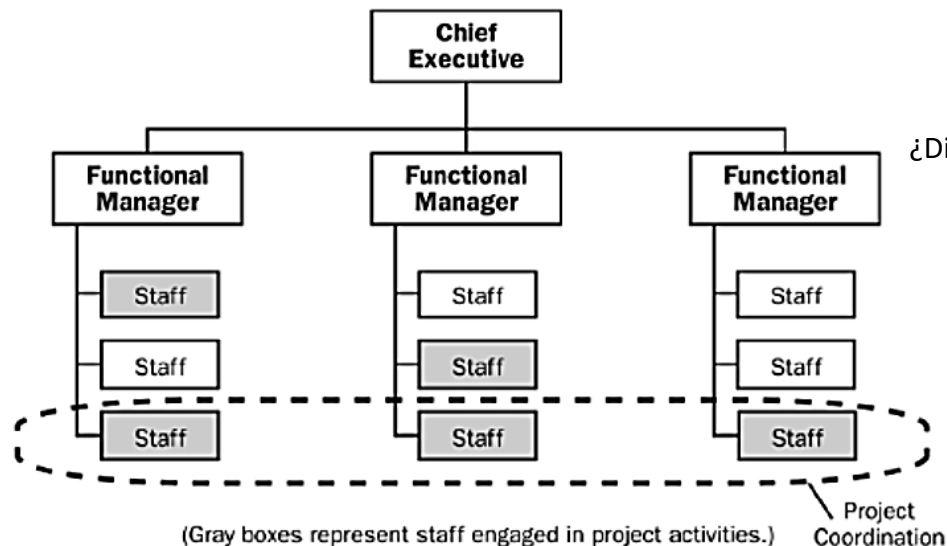
5.4 Arquetipos de organización empresarial

- **Empresa Matricial: débil, equilibrada y fuerte**

- Distintas maneras de aprovechar las ventajas de las distribuciones funcional y orientada a proyectos.

Matricial débil

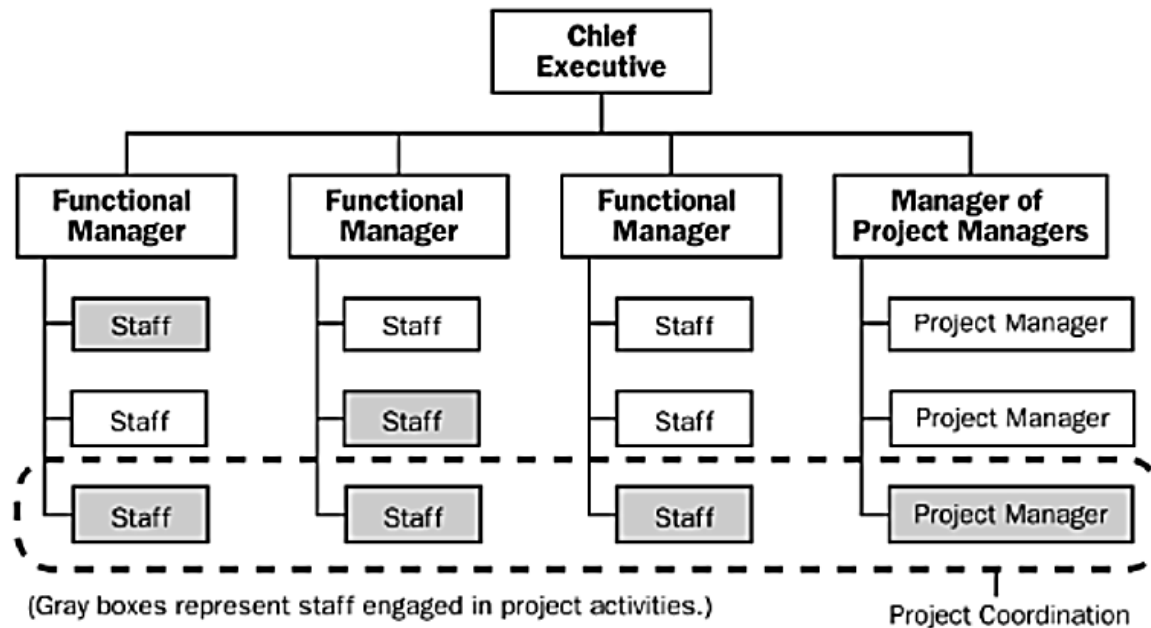
- El PM es seleccionado dentro de un departamento, y es el encargado de coordinar los departamentos para el desarrollo del proyecto. **PM coordina pero no ordena.**
- Por lo demás sigue una organización funcional.



¿Diferencia en la gestión del proyecto entre funcional y matricial débil?

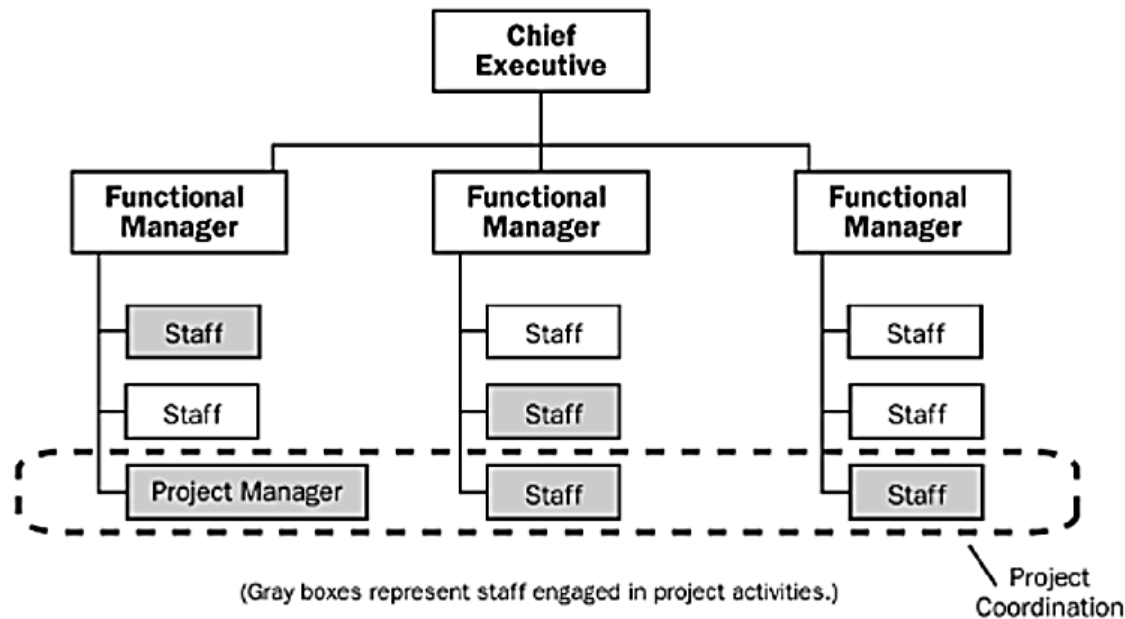
5.4 Arquetipos de organización empresarial

- Matricial fuerte
 - PM y personal a tiempo completo
 - ¿Cuántos jefes tendrá cada miembro?



5.4 Arquetipos de organización empresarial

- Matricial equilibrada
 - El PM tiene autoridad, pero no plena, en especial sobre el presupuesto.
 - El PM y personal no tienen por qué ser a tiempo completo



5.4 Arquetipos de organización empresarial

Estructura de la Organización Características del Proyecto	Funcional	Matricial			Orientada a Proyectos
		Matricial Débil	Matricial Equilibrada	Matricial Fuerte	
Autoridad del Director del Proyecto	Poca o Ninguna	Baja	Baja a Moderada	Moderada a Alta	Alta a Casi Total
Disponibilidad de Recursos	Poca o Ninguna	Baja	Baja a Moderada	Moderada a Alta	Alta a Casi Total
Quién gestiona el presupuesto del proyecto	Gerente Funcional	Gerente Funcional	Mixta	Director del Proyecto	Director del Proyecto
Rol del Director del Proyecto	Tiempo Parcial	Tiempo Parcial	Tiempo Completo	Tiempo Completo	Tiempo Completo

5.4 Arquetipos de organización empresarial

Activos de la empresa

- Algunas **actividades** y **entregables** indicados por la metodología seleccionada para el desarrollo del proyecto ya están definidos por la empresa, y no pueden realizarse con un estilo propio:
 - Plantillas
 - Auditorías tras el proyecto
 - Procedimientos de control de riesgo
 - ...
- Tras finalizar un proyecto, es responsabilidad del PM añadir a **la Base de Conocimiento de la empresa**, la información adquirida:
 - Puntos base para medir el esfuerzo humano
 - Puntos base para la estimación del coste
 - Lecciones aprendidas:
 - Personas clave para el éxito
 - Clientes o personal con los que no conviene trabajar

- Podéis seguir en twitter a:
 - [@Scrumdotorg](#)
 - [@pmiagile](#)

¿Te acuerdas de...?

- ¿En qué difieren el ciclo de vida y la metodología de un proyecto?
- ¿Qué metodología está considerada como la mejor?
- ¿Recomendarías Scrum para un proyecto a largo plazo sin incertidumbre?
- ¿Cómo se programa en Extrem Programming?

¿Te acuerdas de...?

- ¿Qué relación (ciclo de vida) tienen entre sí las fases de la metodología RUP?
- ¿Qué problema puede crear la falta de confianza en las discusiones de equipo?
- ¿Qué organización en empresas funciona mejor para la gestión de proyectos?
- ¿En qué tipo de empresa necesitará más habilidad negociadora el PM?