

## Otras metodologías ágiles

Ingeniería del Software

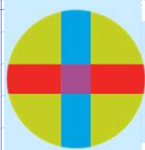
*Curso 2025/2026*

*Universidad San Pablo-CEU*

*Escuela Politécnica Superior*


*Campus de Montepríncipe*

1



## Scrum

2




## Scrum

- » Desarrollada por Ken Schwaber, Jeff Sutherland y Mike Beedle (1995)
- » No está concebido como método independiente
  - Define un marco para la gestión de proyectos
- » Especial para proyectos con un rápido cambio de requisitos
- » Enfatiza valores y prácticas de gestión
  - Conjunto de patrones organizacionales
- » Composición de un equipo Scrum ("*Scrum team*")
  - *Manager*
  - *Scrum Master*
  - *Product Owner*
  - *Scrum Team*

29-ago.-25 Ingeniería del Software página 3

3

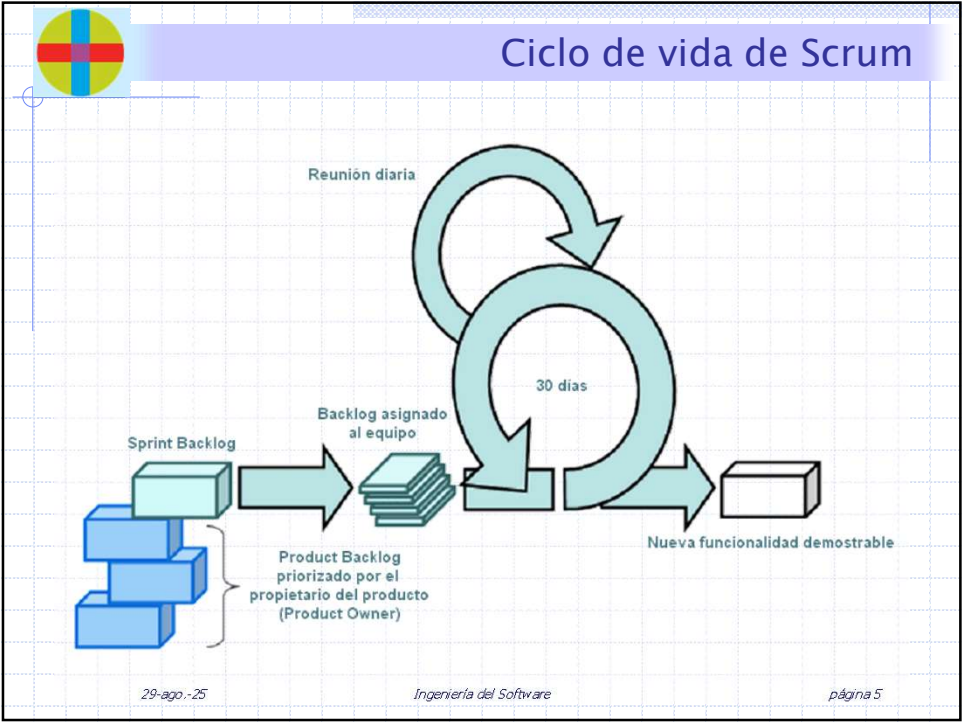


## Valores de Scrum

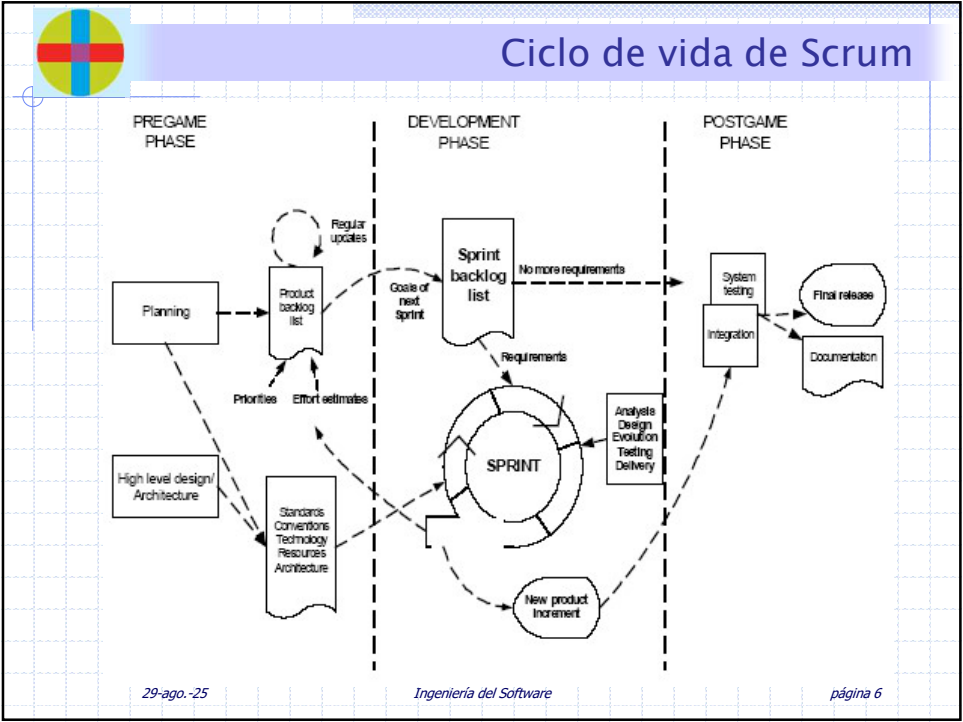
- » Equipos auto-dirigidos y auto-organizados
  - Una vez elegida una tarea, no se agrega trabajo extra
- » El equipo se reúne todos los días, unos 15', 30' máximo ("*scrums*") con tres preguntas
  - ¿Qué has hecho desde el último encuentro?
  - ¿Qué obstáculos hay para cumplir la meta?
  - ¿Qué harás antes del próximo encuentro?
- » La planificación es iterativa (y se hace énfasis en el seguimiento de procesos)
  - La planificación se adapta con la guía del cliente al principio de cada iteración
- » El desarrollo se realiza mediante iteraciones de 30 días llamados carreras ("*sprints*")
  - El resultado de cada *sprint* es un incremento ejecutable que se muestra al cliente

29-ago.-25 Ingeniería del Software página 4


4



5



6




## Ciclo de vida de Scrum (I)

- » Pre-Sprint
  - Creación de la lista *backlog* (documento)
    - › Funciones de usuario
    - › Tecnología
    - › Corrección de errores
    - › Mejoras requeridas
  - Diseño de alto nivel de la arquitectura del sistema
  - El *Product Owner*
    - › Revisa y hace estimaciones a la lista *backlog*
    - › Crea el *release backlog* (*backlog* del Sprint)
    - › Identifica el trabajo que el equipo realizará en el Sprint

29-ago.-25 Ingeniería del Software página 7

7

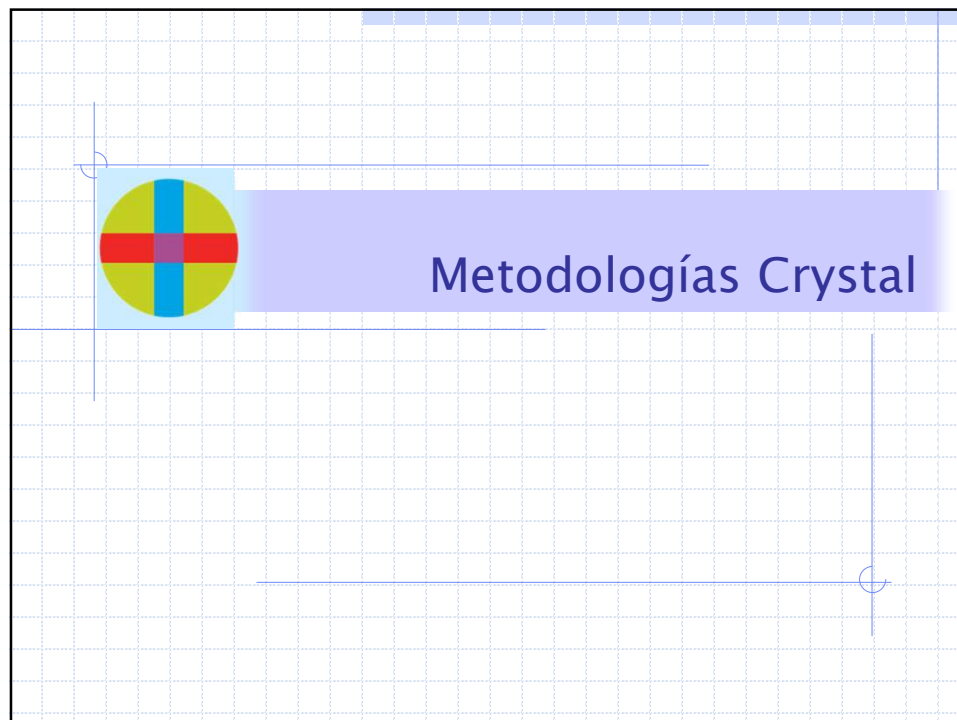


## Ciclo de vida de Scrum (II)

- » Sprint
  - El *Scrum team* es responsable de tareas de ingeniería
  - Reuniones diarias
    - › Los involucrados, desarrolladores y usuarios, deben estar presentes
  - Cerrado a cambios una vez acordado el *release backlog*
- » Post-Sprint
  - Reunión
    - › Revisa los logros y fracasos
    - › La funcionalidad ya completa es presentada
- » Principio del siguiente Sprint

29-ago.-25 Ingeniería del Software página 8

8




9

A presentation slide with a light blue grid background. On the left, there is a circular logo divided into four quadrants: top-left is yellow, top-right is blue, bottom-left is red, and bottom-right is green. To the right of the logo is a light purple rectangular box containing the text '¿Qué son las metodologías Crystal?' in a dark blue, sans-serif font.

- » Desarrolladas por Alistair Cockburn en los años 90
- » Dan vital importancia a las **personas** que componen el equipo de un proyecto, y por tanto sus puntos de estudio son:
  - Aspecto humano del equipo
  - Tamaño de un equipo (número de componentes)
  - Comunicación entre los componentes
  - Distintas políticas a seguir
  - Espacio físico de trabajo
- » Buscan reducir al máximo los **artefectos** producidos
- » El desarrollo de software se considera un **juego** cooperativo de invención-comunicación, limitado por los recursos a utilizar

29-ago.-25      Ingeniería del Software      página 10

10




## Características del equipo Crystal

- » El **equipo de desarrollo** es un factor clave, por lo que se deben invertir esfuerzos en mejorar sus **habilidades y destrezas**, así como tener **políticas de trabajo en equipo definidas**
  - Estas políticas dependerán del tamaño del equipo, estableciéndose una clasificación por colores
  - Crystal aconseja que el tamaño del equipo sea reducido (pocos componentes)
- » La mejora de la **comunicación** entre los miembros del equipo del proyecto:
  - Mismo lugar de trabajo → baja el coste de la comunicación

**Mejora individual → mejora global del equipo**

29-ago.-25 *Ingeniería del Software* página 11

11




## Diferentes políticas de equipo

- » “Se utilizarán **políticas diferentes** para **equipos diferentes**”
- » Codificación por colores de Crystal: según el tamaño del equipo:
  - **Crystal Clear**
    - › Color Blanco (3 – 8 personas)
  - **Crystal Orange**
    - › Color Naranja (25 – 50 personas)

3-8	10-20	25-50	50-100	100-200	200-500	800+
-----	-------	-------	--------	---------	---------	------

29-ago.-25 *Ingeniería del Software* página 12

12



## Roles y prioridades

» Roles de Crystal:

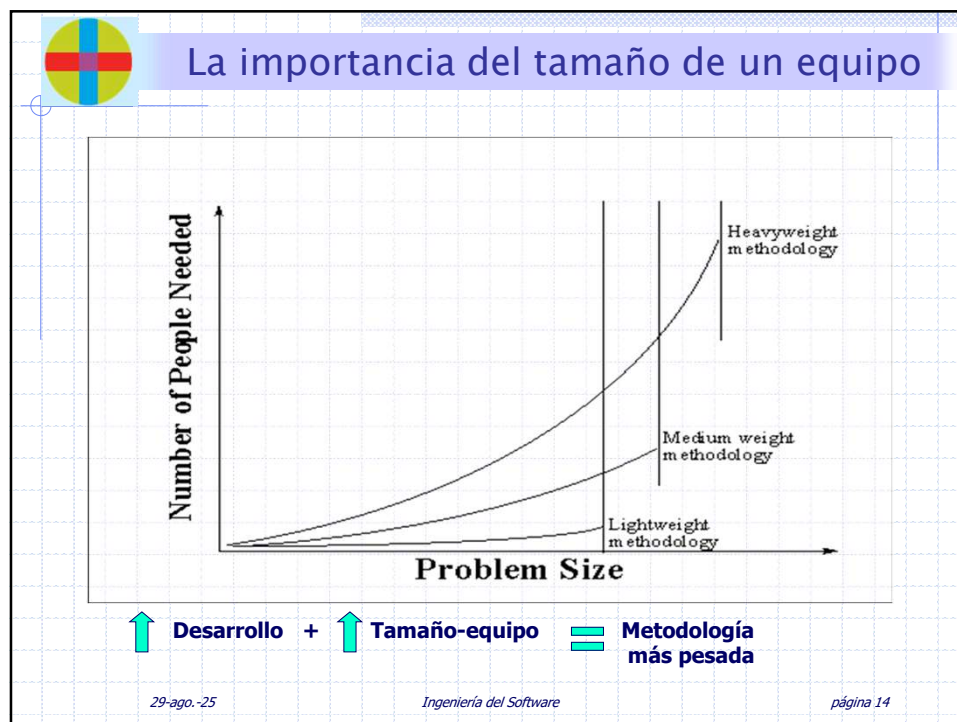
- *Executive Sponsor* (patrocinador ejecutivo)
- *Project Manager* (jefe de proyecto)
- *Domain Expert* (experto en el dominio)
- *Usage Expert* (experto en ergonomía)
- *Designer-Programmer* (programador-diseñador)
- *UI Designer* (diseñador de la interfaz de usuario)
- *Tester* (realizador de pruebas)
- *Technical* (programador técnico)

» Cada metodología tiene unas prioridades a la hora de intentar alcanzar el éxito de la aplicación:

- La familia de **Crystal** → combinación de productividad y tolerancia
- Metodología **XP** → aumenta la productividad pero disminuye la tolerancia (menos importancia al individuo)

29-ago.-25      Ingeniería del Software      página 13

13



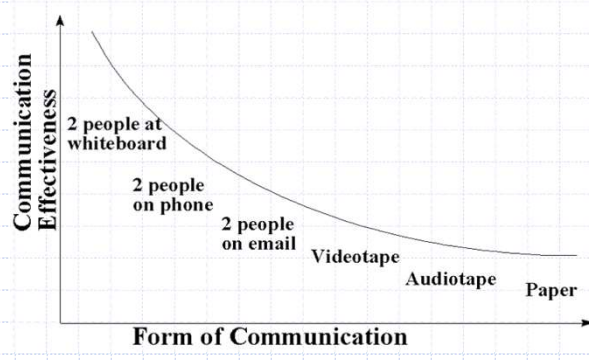
14



## La importancia de la comunicación

» La **comunicación** es más barata y mejor cuanto más “cercana” sea

- Crystal recomienda la **interacción cara a cara**, por ser éste el mejor método de comunicación



Form of Communication

página 15

15

## Conclusiones

» **Conclusiones**

- Cuantas más personas estén implicadas, más grande debe ser la metodología
- Si el proyecto tiene mucha densidad, un error no detectado puede ser crítico
- El aumento de tamaño o densidad añade un coste considerable al proyecto
- La forma más eficaz de comunicación es la interactiva (cara a cara)

» **Bibliografía**

- Diferentes artículos: <http://alistair.cockburn.us/Crystal>
- “Methodology Per Project” (Alistair Cockburn)

página 16

16






17

Dynamic Systems Development Method(DSDM)

- » Nace en 1994 con el objetivo de crear una metodología RAD unificada
- » Define el marco para desarrollar un proceso de **producción de software**
- » Es un proceso iterativo e incremental y el equipo de desarrollo y el usuario trabajan juntos
- » Propone **cinco fases**:
  - Estudio de viabilidad
  - Estudio del negocio
  - Modelado funcional
  - Diseño y construcción
  - Implementación
- » Las tres últimas son iterativas, además de existir realimentación a todas las fases

29-ago.-25      Ingeniería del Software      página 18

18




## Adaptive Software Development (ASD)

- » Su impulsor es Jim Highsmith
- » Es iterativo, orientado a los componentes software más que a las tareas y tolerante a los cambios
- » El ciclo de vida que propone tiene tres fases esenciales:
  - Especulación:
    - › Se inicia el proyecto
    - › Se planifican las características del software
  - Colaboración
    - › Se desarrollan las características
  - Aprendizaje
    - › Se revisa su calidad, y se entrega al cliente.
    - › La revisión de los componentes sirve para aprender de los errores y volver a iniciar el ciclo de desarrollo

29-ago.-25 Ingeniería del Software página 19

19

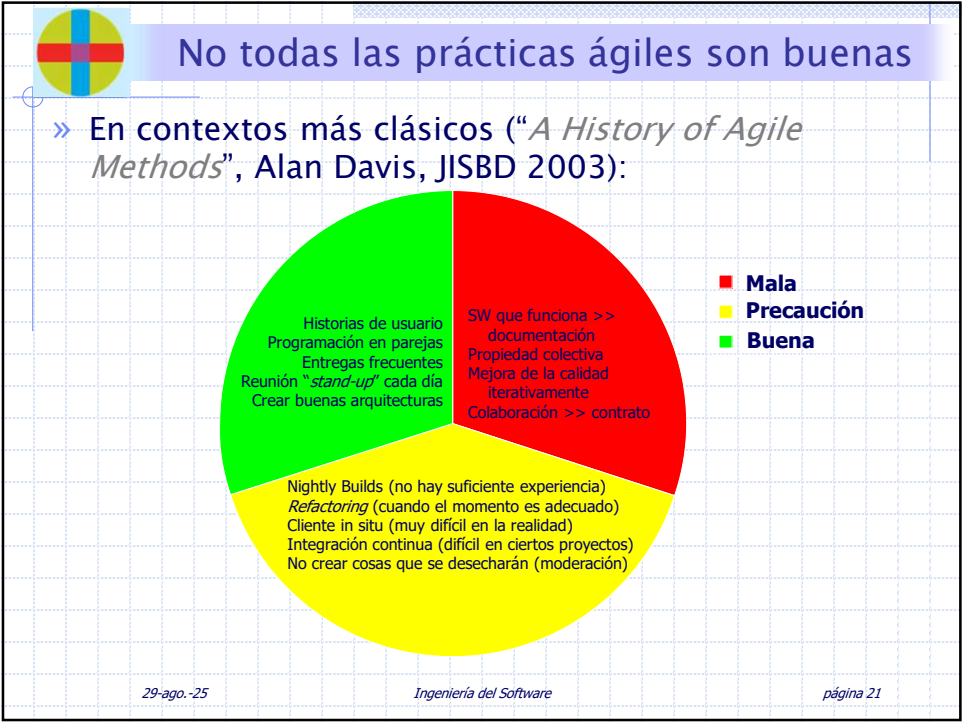


## Metodologías LD y FDD

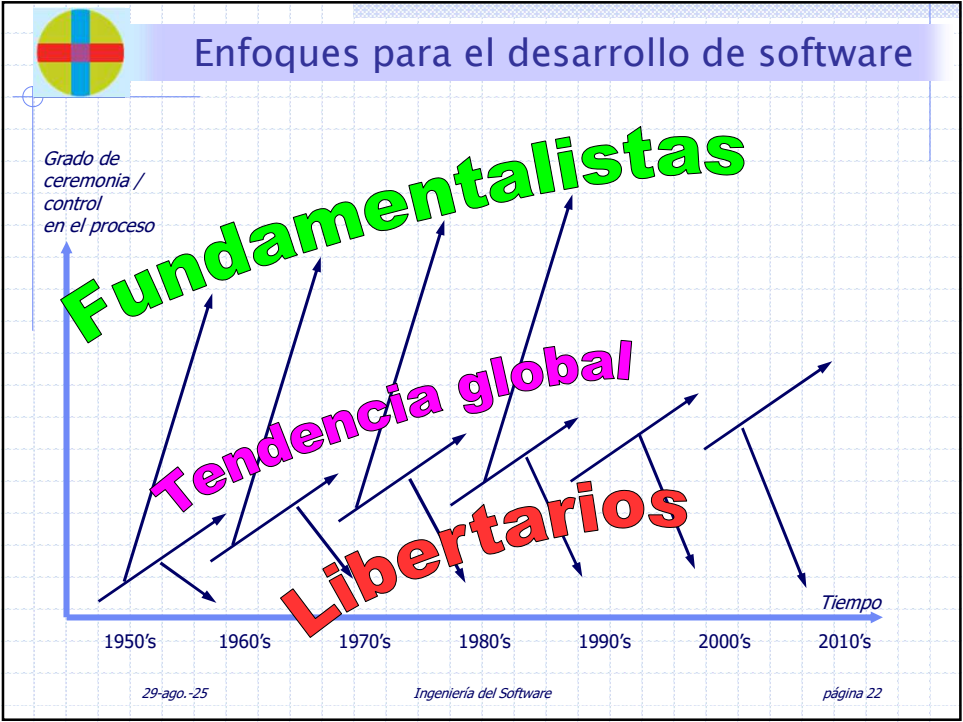
- » Lean Development (LD)
  - Definida por Bob Charette
  - En LD, los cambios se consideran riesgos, pero si se manejan adecuadamente se pueden convertir en oportunidades que mejoren la productividad del cliente
  - Su principal característica es introducir un mecanismo para implementar dichos cambios
- » Feature-Driven Development (FDD)
  - Sus impulsores son Jeff De Luca y Peter Coad
  - Define un proceso iterativo que consta de 5 pasos
  - Las iteraciones son cortas (hasta 2 semanas)
  - Se centra en las fases de diseño e implementación del sistema partiendo de una lista de características que debe reunir el software

29-ago.-25 Ingeniería del Software página 20

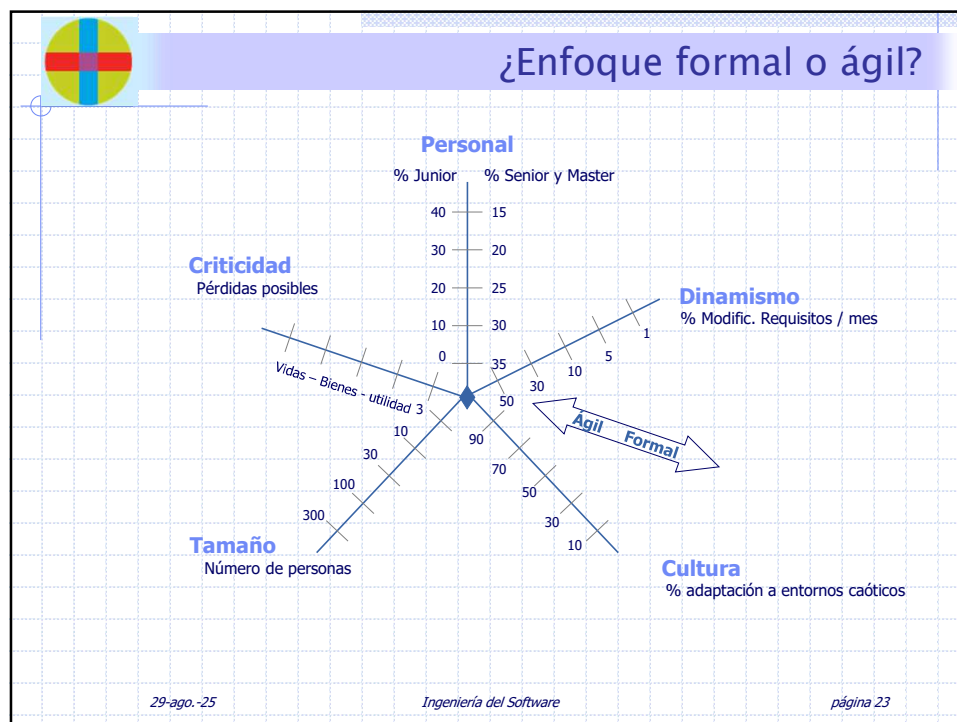
20



21



22



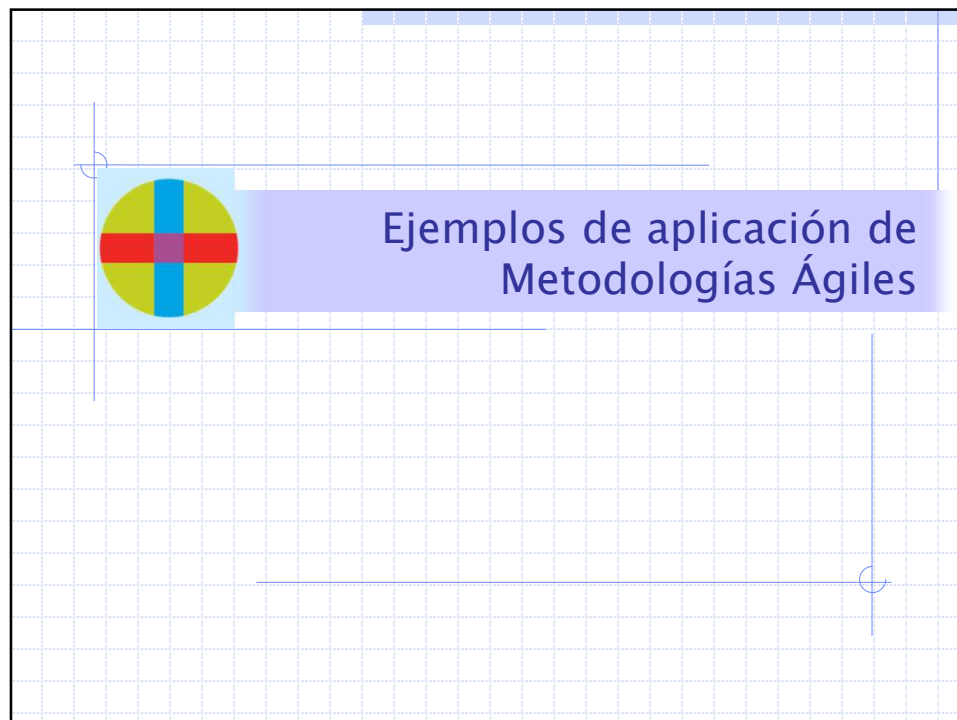
23

### Resultados de las Metodologías Ágiles


- » Hay pocos datos del índice de éxito de proyectos
- » Las *AMs* están teniendo un gran auge
  - Aumento en el número de proyectos
- » ¿Por qué?
  - Tienen el apoyo de muchos gurús en ingeniería de software
  - Son procesos para gente que odia los procesos
  - Tienen sentido
  - ¿Política? ... Pugna entre comunidades
    - > XP
    - > Scrum
    - > ...

29-ago.-25      Ingeniería del Software      página 24

24



25




## Ejemplo – Empresa 1

» Contexto

- Dos grandes proyectos basados en aplicaciones Web que se han mudado a una metodología ágil (XP)
- La empresa certifica nivel 3 (CMM) en organizaciones de proceso.
- Ambos proyectos son grandes (más de 1 millón de líneas de código), distribuidos y complejos
- Programados por equipos de desarrolladores geográficamente distribuidos en múltiples zonas horarias
- Los proyectos son manejados por consideraciones de mercado
- Nunca tendrán sus requisitos totalmente definidos hasta la entrega
- Existen presiones muy fuertes por la entrega funcional a tiempo

29-ago.-25      Ingeniería del Software      página 26

26



## Empresa 1 – Proyecto 1


- » **Objetivo**
  - “Construir servidores de aplicación para comerciales en EEUU y Europa”
    - › Los comerciales usan estos programas analíticos para perfilar mercados que varían de país a país, donde nuevos algoritmos proveen tendencias y otra información que cada usuario valora de forma diferente de acuerdo con las preferencias nacionales
- » **Motivación**
  - Acelerar la entrega de los productos al mercado acortando los tiempos y reduciendo los costes
- » **Desarrollo**
  - Se ha tenido cuidado para escalar las prácticas ágiles de XP como programación en parejas en este largo proyecto
  - Las parejas se conforman diariamente según la destreza de cada desarrollador en cada segmento específico del mercado

29-ago.-25

Ingeniería del Software

página 27

27



## Empresa 1 – Proyecto 1


- » **Resultados**
  - El personal se resistió a usar procesos que su grupo de desarrollo había establecido como demasiado formales y con demasiada documentación
  - El manager de negocios tuvo que proteger al grupo agilista (XP) de las críticas del grupo de procesos, sosteniendo que se trataba de un proyecto piloto
  - El grupo de procesos generó una discusión con el *senior manager*, argumentando que su proceso ya representaba las mejores prácticas de la industria

29-ago.-25

Ingeniería del Software

página 28

28



## Empresa 1 – Proyecto 2


- » **Objetivo**
  - “Desarrollar aplicaciones Web para usar en toda la empresa”
    - › Se redefine la arquitectura
    - › Cada aplicación puesta en la web debe ser compatible con la anterior y proveer equivalentes prestaciones
- » **Motivación**
  - Ahorro de costes producidos por viajes
  - Guardar un registro en la Web
  - El *senior manager* ha dictaminado que todos los sistemas de compañías, desde los sistemas de contabilidad (*accounting*) hasta los sistemas de gestión de comercio, deben residir en servidores Web, ubicados alrededor del mundo en los próximos 18 meses

29-ago.-25

Ingeniería del Software

página 29

29



## Empresa 1 – Proyecto 2

- » **Desarrollo**
  - Para satisfacer los requisitos de la dirección se planificó más de un año de trabajo por parte de varios grupos de desarrollo
  - Estos grupos eligieron prácticas ágiles (XP) porque les daba esperanza de llegar a la planificación
  - Las experiencias en procesos pasados (con métodos clásicos) predecían que sus modelos les llevarían por lo menos 5 años en poner estas aplicaciones en su portal
  - La orden de la dirección de la compañía es un tiempo de desarrollo de 18 meses, no más


29-ago.-25

Ingeniería del Software

página 30

30






## Empresa 1 – Proyecto 2

### » Resultados

- Los desarrolladores modificaron las practicas ágiles (XP) en el área de pruebas (práctica *test-first*)
  - › Formaron grupos independientes de pruebas
  - › Generaron documentación adicional
  - › La documentación de pruebas generada fue menor a la requerida por el proceso de la organización (nivel 3 CMM)
- El equipo SQA (*System Quality Assurance*) planteó al *senior manager* la carencia de documentación de pruebas
  - › Apuntaron un alto contenido de error
- El director de proyecto XP sacó fuera del proyecto al grupo de SQA
  - › No agregaban valor al proyecto y su rol pasaba desapercibido
- Como resultado, el proyecto se detuvo

29-ago.-25      Ingeniería del Software      página 31

31



## Motorola

### » Contexto


- Se experimentan numerosas readaptaciones al proceso en proyectos con tiempos cortos de entrega, cuando los requisitos no son estables o cambian frecuentemente
- El objetivo prioritario de la organización es el de mantener su nivel CMM, por lo que no está dispuesta a sacrificar el uso de prácticas maduras para introducir Metodologías Ágiles

### » Motivación

- Reducir los tiempos de finalización del producto para salir antes al mercado, manteniendo al mismo tiempo una alta calidad
- Las Metodologías Ágiles prometen desarrollo de proyecto en tiempo récord, a la vez que enfatizan la satisfacción total del cliente

29-ago.-25      Ingeniería del Software      página 32

32



Motorola

» Resultados

- Las métricas seleccionadas fueron basadas en indicadores tradicionales de duración, costes y defectos en el producto final
  - Se obtuvieron valores ligeramente mejores que la media organizacional, sin llegar a ser estadísticamente significativos

» En resumen


- Gracias a estas prácticas se incorporó a la organización
  - planificación diaria
  - entregas frecuentes
  - integración continua
  - uso de *backlogs* de requisitos
- No incorporar de estas prácticas hubiera implicado un sobreesfuerzo sumado a un alto riesgo de no cumplir los compromisos
- El resultado en conjunto resultó altamente positivo

29-ago.-25

Ingeniería del Software

página 33

33



Resumen


	Proyecto 1	Proyecto 2	Proyecto Motorola
Equipo de desarrollo distribuido	+	-	-
Posee certificación CMM	+	+	+
Corto tiempo de desarrollo	+	++	+
Conflicto entre CMM y XP	+	+	-
Motivaciones de adopción de Metodologías Ágiles	Reducir tiempo y costes	Reducir tiempo	Reducir tiempos Adaptarse a proyectos pequeños

29-ago.-25


Ingeniería del Software

página 34

34



¿Preguntas?



29-ago.-25

Ingeniería del Software

página 35