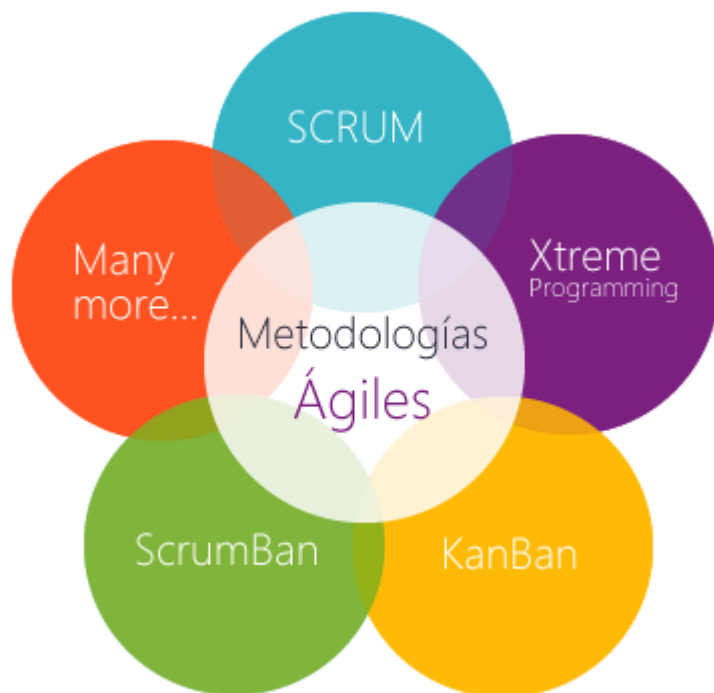


Qué son las metodologías ágiles y su **aplicación** en el mercado actual

Capgemini



Gonzalo Gabino Diestro
ggabinod@capgemini.com

Programador, analista, team leader, scrum master, jefe de proyecto, consultor técnico de preventa y actualmente manager del departamento de soluciones software en el CoE de Capgemini

NO soy un gurú de las metodologías ágiles, sólo me adapto.

Las especies que sobreviven no son las más fuertes,
sino aquellas que se adaptan mejor.
Darwin



Metodología Tradicional



a Capgemini production

En su nacimiento(1968), la **gestión de proyectos software** intentó imitar la gestión de proyectos de otras disciplinas, como la arquitectura, las industria o la ingeniería civil.

Hereda y adapta al mundo del software muchos de sus roles (arquitectos software) y tipos de organizaciones (fábricas de software).

Por lo tanto, la manera de gestionar los proyectos es una conclusión lógica de la elección de imitar a la Construcción:

modelo dirigido por la planificación, desarrollo tradicional, en cascada o **predictibilidad**.

Ciclo de Vida “tradicional” o en cascada

Se divide el proyecto en fases

No se pasa de una fase a otra hasta que no se acaba la anterior

Se intenta predecir lo que pasa en la siguiente fase

Se especializa cada fase a un rol distinto

toma de requisitos

analistas

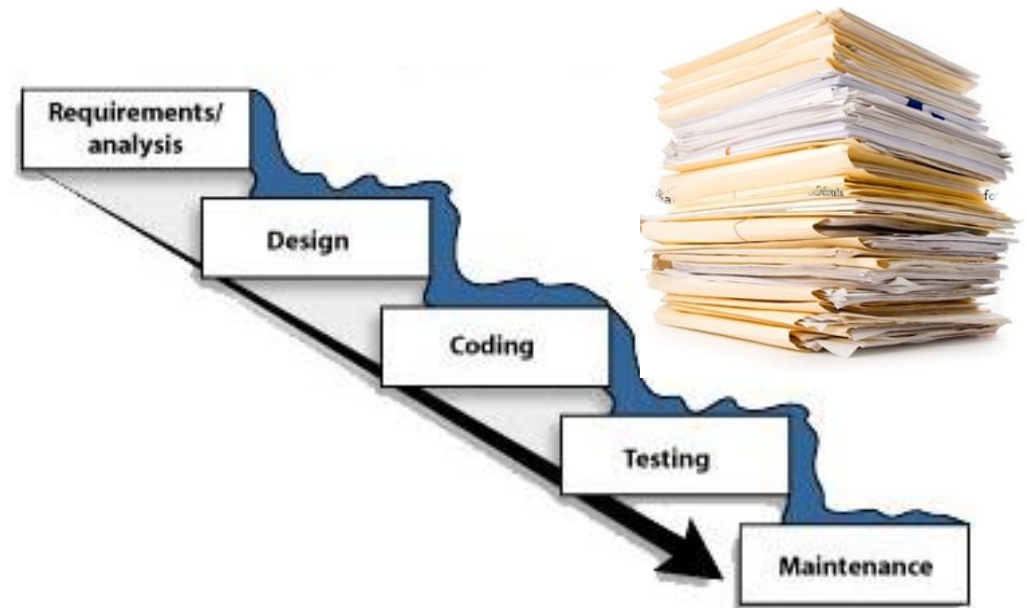
diseñadores

arquitectos técnicos

programadores

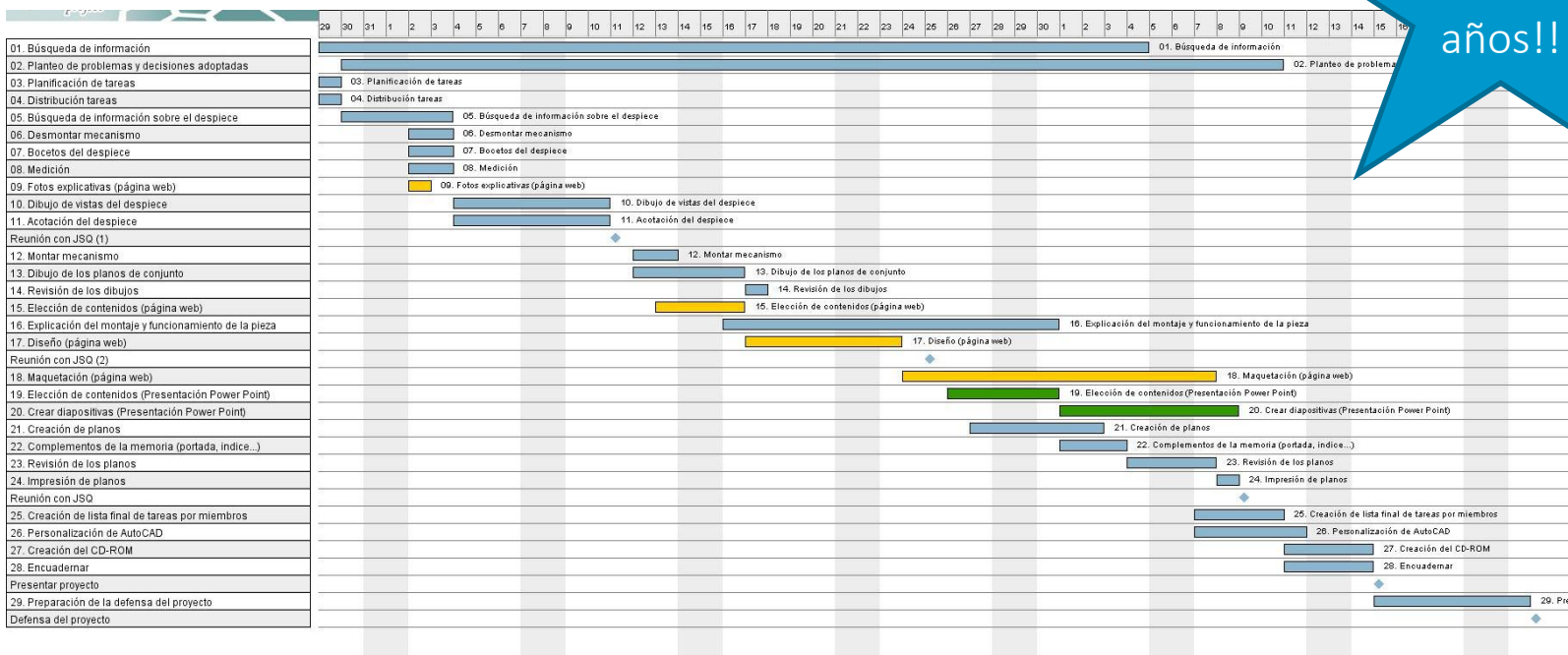
tester....

No se puede volver hacia atrás



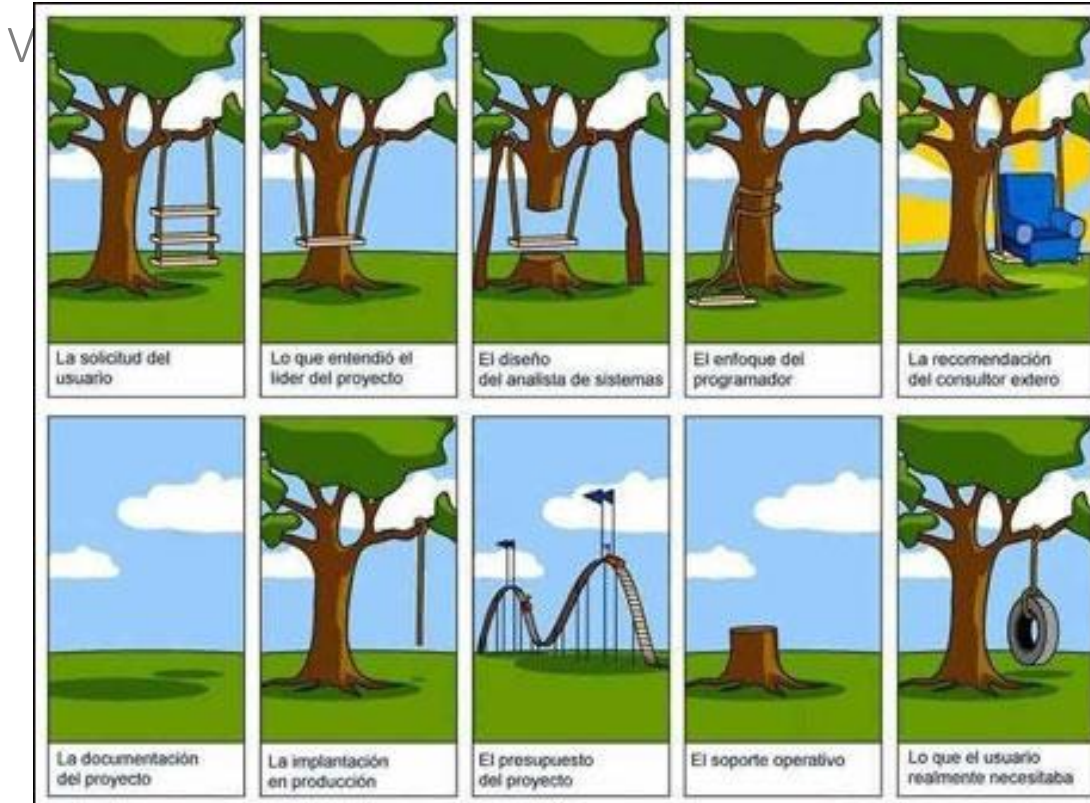
Ciclo de Vida “tradicional” o en cascada

Diagrama de Gantt



3
años!!

Ciclo de Vida “tradicional” o en cascada



io del ciclo de vida.

yecto.

l proyecto.

s de negocio en la entrega.

emisa es FALSA!!

El negocio cambia. La tecnología cambia. El equipo cambia.

El problema no es el cambio en sí, porque el cambio va a ocurrir; el problema es la incapacidad para adaptarse al cambio cuando llega.

eXtreme Programming

Estamos descubriendo mejores maneras de hacer software

Firmantes del Manifiesto Ágil 2001

Se fundamenta sobre 4 valores:

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

Aunque valoramos los elementos de la derecha, valoramos más los de la izquierda.

Metodologías Ágiles



a Capgemini production

...un Ciclo de Vida Iterativo e Incremental

Entregar **prototipos**, pequeños evolutivos que los usuarios puedan tocar.

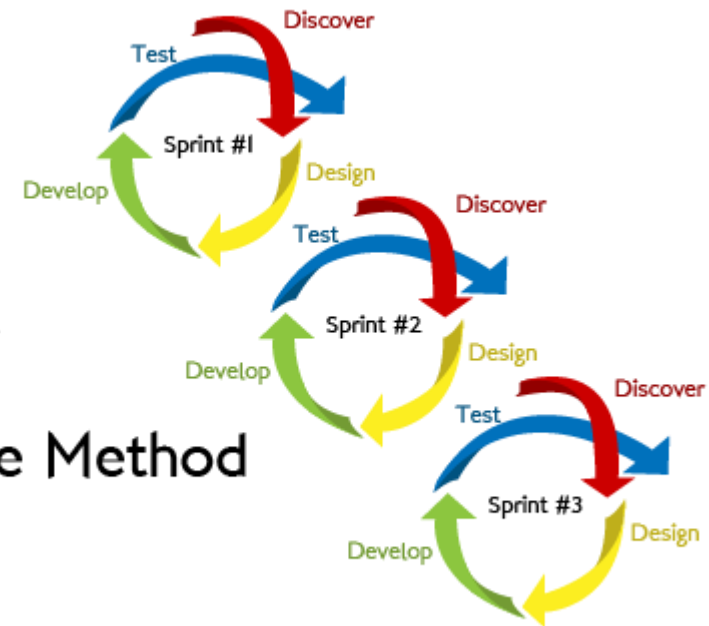
Con **funcionalidad añadida** en cada iteración.

Pero también **Refinando** y **Refactorizando** en base al feedback de los usuarios.

El ciclo de vida iterativo e incremental es una de las bases de un proyecto ágil.

Iteraciones **cortas en tiempo**, de pocas semanas

Agile Method

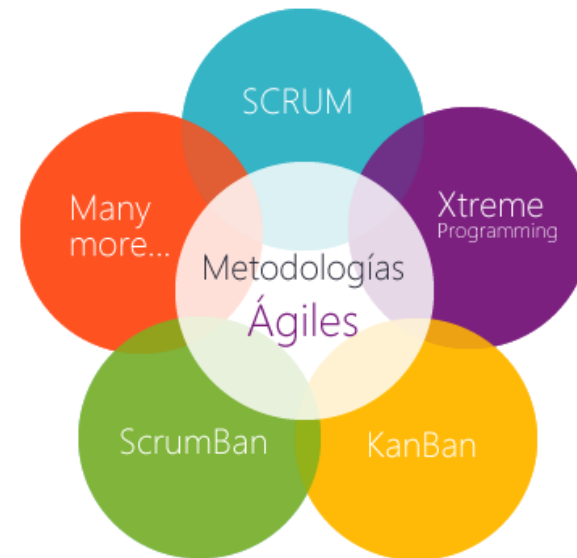


En 1950 se utilizó por primera vez en la NASA el ciclo de vida iterativo, en el desarrollo del software del avión X-15



La principal diferencia entre cascada, espiral y los ciclos de vida ágiles es que en estos últimos se asume que el análisis, diseño, etc., de cada iteración o Sprint son impredecibles. Los **Sprints**, o iteraciones cortas, no son (o a priori no tienen porqué) lineales y son flexibles.

Finalmente cada metodología de las llamadas ágiles, FDD, Crystal, Kanban, XP, Scrum, etc., matizará su ciclo de vida.



Entre las metodologías ágiles más usadas se encuentran:

SCRUM. Es un marco de trabajo que nos proporciona una serie de herramientas y roles para, de una forma iterativa, poder ver el progreso y los resultados de un proyecto.

KANBAN. Se basa en una idea muy simple. Ésta es que el trabajo en curso (Work In Progress, WIP) debería limitarse y sólo deberíamos empezar con algo nuevo cuando un bloque de trabajo anterior haya sido entregado o ha pasado a otra función posterior de la cadena.

eXtreme Programming (XP): Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores y propiciando un buen clima de trabajo.

Historias de usuario: lo que el cliente o el usuario quiere que se implemente; es decir, son una descripción breve, de una funcionalidad software tal y como la percibe el usuario.

Product backlog: es el conjunto de las historias del usuario, el alcance del producto.

Sprint: ciclo corto en el que se entrega un prototipo a usuario

Sprint backlog: las historias de usuario que entran en el sprint.

El “**product owner**” (o propietario del producto) es aquella persona con una visión muy clara del producto que se quiere desarrollar, que es capaz de transmitir esa visión al equipo de desarrollo y, además, está altamente disponible.

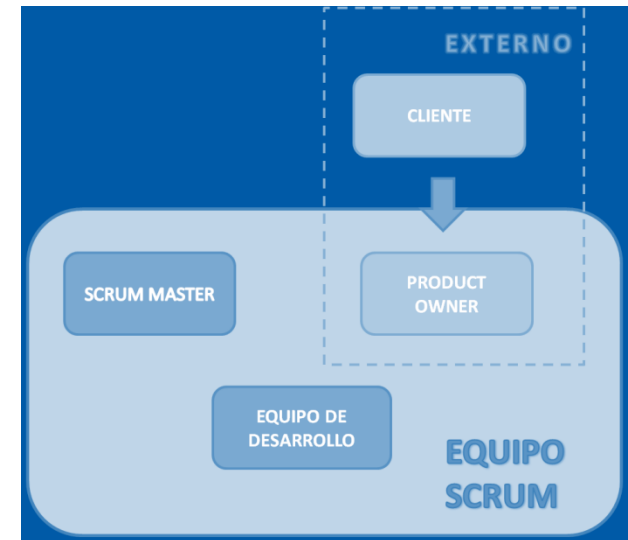
La figura del product owner es clave en un proyecto ágil, en su planificación y seguimiento. Debe ser quien decida que historias de usuario entran en el product backlog, cuáles no, y además la prioridad de las historias del product backlog.

Scrum se basa en tres pilares:

- **Transparencia**, potenciando la comunicación. Utilizando pizarras para mostrar toda la información
- **Inspección**, se debe controlar el avance, las historias de usuario, el equipo...
- **Revisión o retrospectivas**, que nos den feedback después de cada sprint

Todo **equipo** Scrum tiene tres roles principales:

1. **Scrum Master**: es el responsable de asegurar que el equipo Scrum siga las prácticas de Scrum
2. **Propietario del Producto (ProductOwner)**
3. **Equipo de desarrollo**, con las siguientes características:
 - Auto-gestionado
 - Multifuncional
 - No-distribuido
 - Tamaño óptimo: mínimo 3 personas



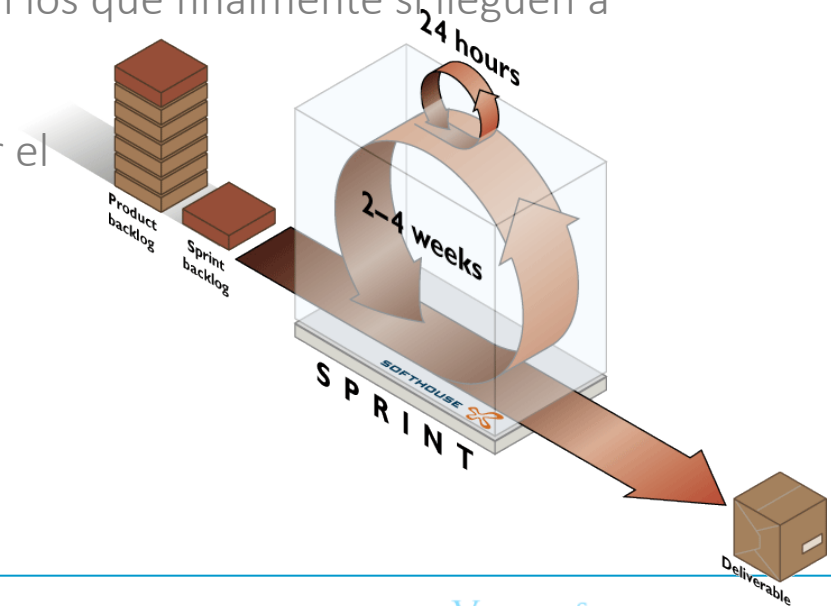
Una vez identificado el product backlog, el product owner **ordena** las historias de usuario.

Se determina la duración del sprint, entre 2 y 4 semanas.

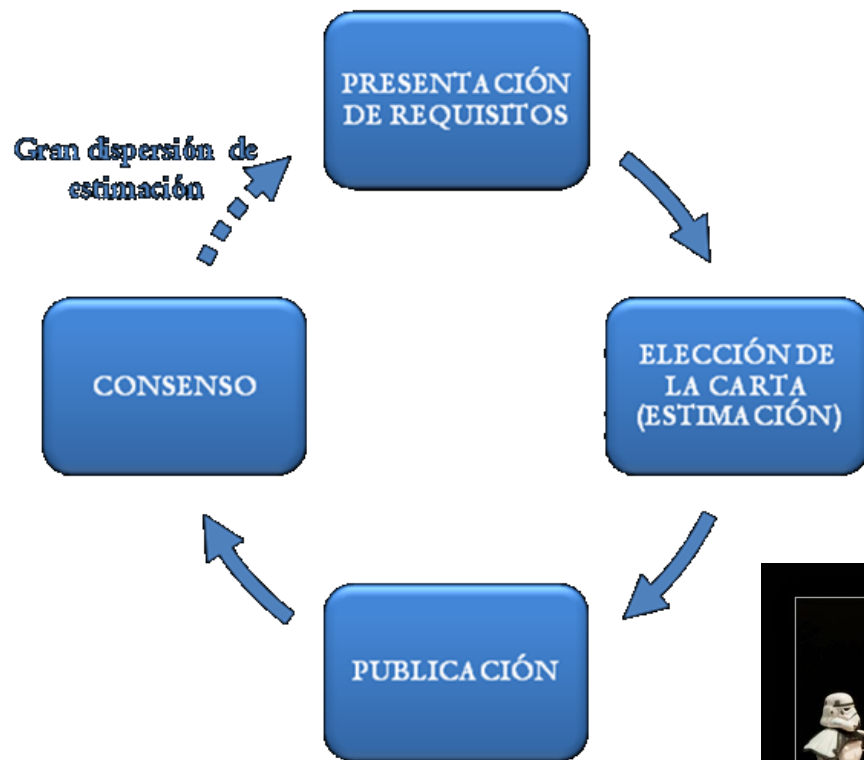
Del product Backlog, se escoge una serie de historias que formarán parte del sprint backlog

- Para estimar las historias de usuario es el Planning Poker
- Aunque teóricamente cada sprint conlleva una entrega en producción, en la actualidad se marcan unos sprints, como sprint de release que serán los que finalmente sí lleguen a producción.

El **Product Owner** puede visualizar, pero no puede modificar el Sprint Backlog. En cambio, puede modificar el ProductBacklog cuantas veces quiera con la única restricción de que los cambios tendrán efecto una vez finalizado el Sprint.



La técnica de Planning Poker está **basada en el consenso** y es utilizada para estimar el esfuerzo o el tamaño relativo de las tareas de desarrollo de software



Se presentan las historias de usuario
Tras este período de discusión cada una de las personas implicadas toma el mazo y estima.
Finalmente se publican todas las estimaciones a la vez.

Si hay mucha dispersión, se discute sobre esa historia de usuario otra vez hasta llegar al consenso



| | | | | | |
|---|-----|----|----|-----|---|
| 0 | 1/2 | 1 | 2 | 3 | 5 |
| 8 | 13 | 20 | 40 | 100 | ? |

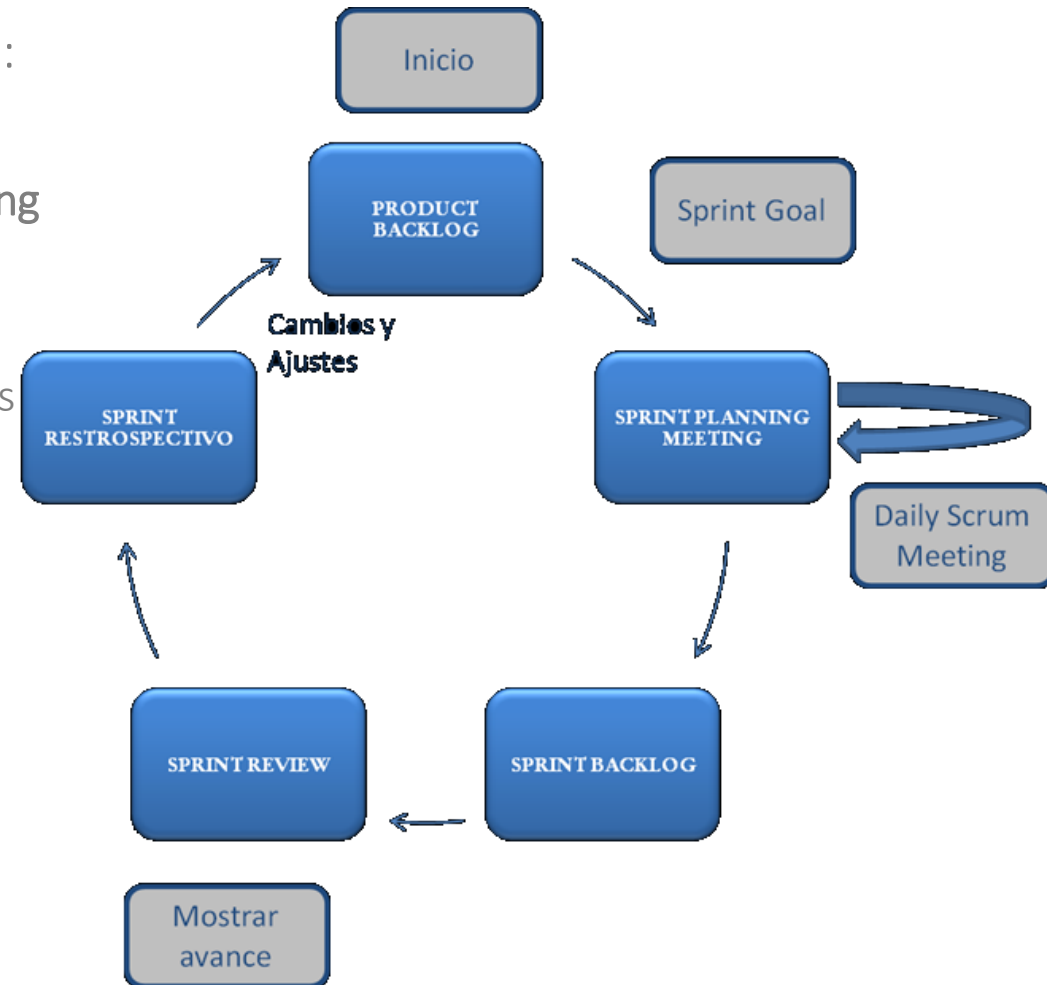
Un pilar importante es el sistema de reuniones:

Reunión de planificación del Sprint (Sprint Planning Meeting) 8h máx. → participan todos los roles.

Reunión diaria (Daily Scrum) 15min. Máx → todos menos PO

Reunión de revisión del Sprint (Sprint Review Meeting): 4h. Máx. → todos los roles

Retrospectiva del Sprint (Sprint Retrospective) 4h máx. → todos menos PO



Ventajas teóricas

Alcance totalmente abierto.

Por lo tanto, gran capacidad de adaptación al cambio.

Planificación de sprints cortos con prototipos en producción.

Por lo tanto, el coste se puede asociar a cada sprint.

Colaboración estrecha e intensa con el cliente (product owner, retrospectivas)

Producto completamente adaptado a requisitos de negocio en la entrega.

Estás Fuera de Mercado
(al menos del 90%)

La agilidad está muriendo...bienvenidos al *postagilismo*!!

El **postagilismo** no es ni mucho menos un movimiento tan organizado y amplio como el agilismo. No hay manifiesto postagilista 😊.

El término representa:

- “El **postagilismo** es simplemente hacer lo mejor para ti”, por encima de seguir una metodología ágil al detalle.
- Seguir los principios base de los métodos ágiles sin seguir una metodología ágil al pie de la letra.
- El postagilismo **no** es anti-agilismo. Es **evolución** del agilisimo.
- La realidad no es como la cuentan los libros de desarrollo ágil, el postagilismo es una visión liberal frente al desarrollo ágil. Hay empresas que trabajan bien de otra manera, no tan ágil, pero que tampoco es en cascada.

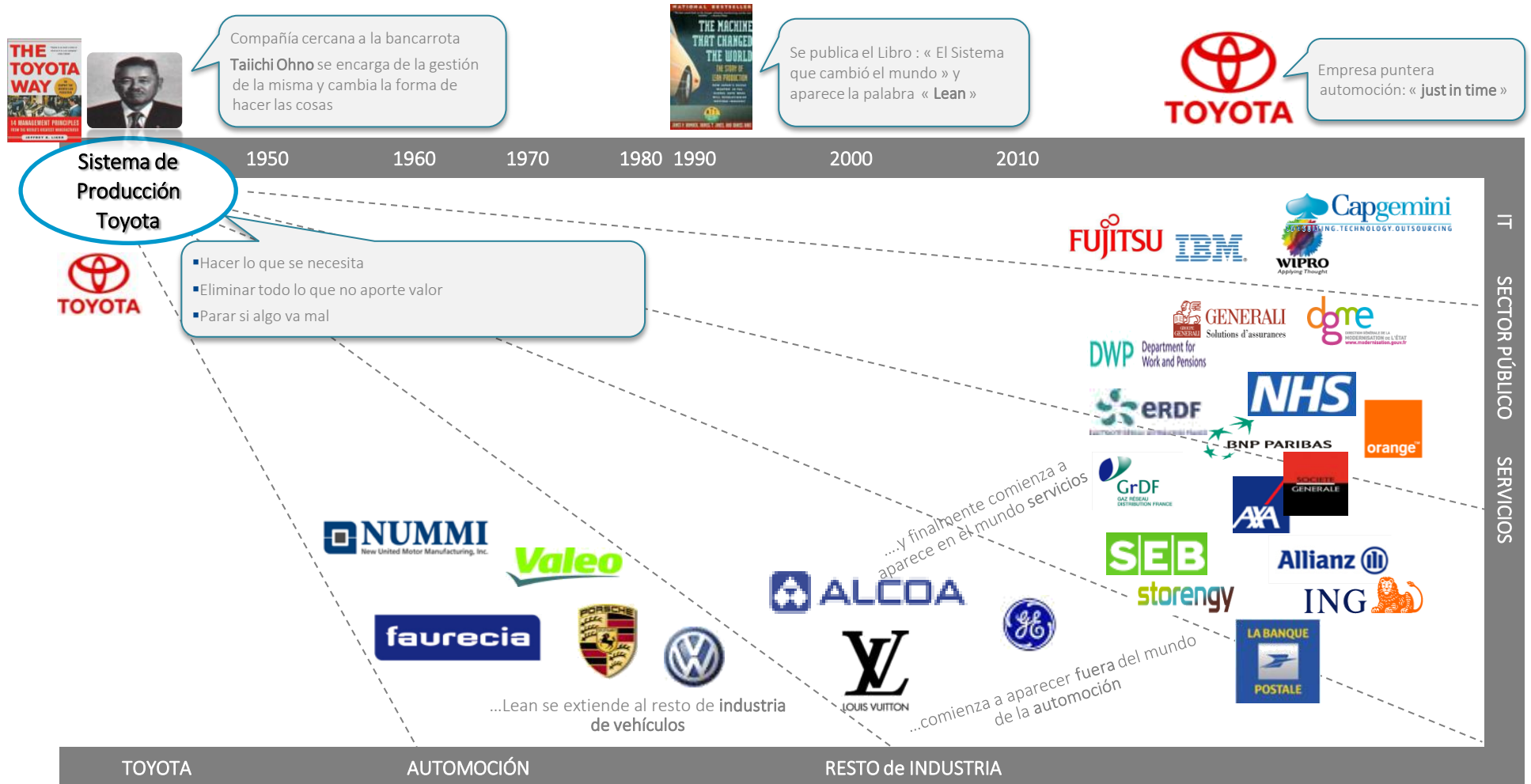
Sentido común. Conocimiento de todas las técnicas. Y saber aplicar lo que cada problema requiere, bien sea más ágil, menos ágil o con técnicas de ingeniería del software de toda la vida.

Filosofía Capgemini

El origen: Lean



Contexto Lean Management o Lean IT



Lean nace como una necesidad de mejora y eliminación de ineficiencias

El método **Lean Thinking** (aspecto 1/2)

Lean consiste esencialmente en la eliminación de “Waste” o desperdicios y en proporcionar el máximo valor al Cliente

- + La aplicación de metodologías Lean contribuye a hacer nuestros procesos de Delivery más eficientes
- + Nos ayuda a mejorar la comunicación en el equipo de trabajo, tanto en NearShore como OnShore.
- + Nos permite tener un control mucho más cercano de la Producción

PRINCIPIOS LEAN



VALOR

Entender qué es lo que **añade valor al Cliente**



INEFICIENCIAS

Eliminar “ineficiencias” y reducir el nivel de complejidad



DEMANDA

Unir el flujo de valor directamente con la demanda del Cliente



RESPONSABILIDAD

Otorgar la **responsabilidad** de la gestión del proceso a los empleados y hacerles partícipes de su mejora



MEJORA

Perseguir **constantemente** la mejora continua

La Mejora Continua es la actitud básica para asegurar la eliminación progresiva de esos desperdicios detectados

Lean consiste esencialmente en la eliminación de “Waste” o desperdicios y en generar una mentalidad de mejora continua



FUNDAMENTOS

Proveer más valor añadido al Negocio de nuestros Clientes

Calidad

Incrementar la calidad

- Producir de acuerdo a las necesidades del negocio
- Disminuir el número de errores durante el desarrollo
- “Do it right the first time”

Time to market

Reducir el Time To Market (TTM)

- Reducir el tiempo entre la petición de una funcionalidad y su puesta en producción

Productividad

Mejora de la productividad

- Producir más y a mayor velocidad
- Liberación de capacidad mediante la eliminación de “waste”



DESARROLLAR TAREAS CON VALOR



ELIMINAR INEFICIENCIAS



CONTROLAR LA DEMANDA



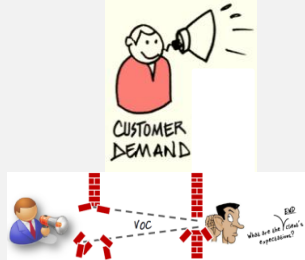
OTORGAR RESPONSABILIDAD



MEJORA CONTÍNUA

The Capgemini Lean Foundations 3.0

VOICE OF THE CUSTOMER



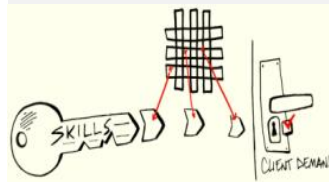
Conjunto de principios que nos ayuden a identificar aquello que realmente **añade valor al cliente** y al negocio

LEADERSHIP ENGAGEMENT



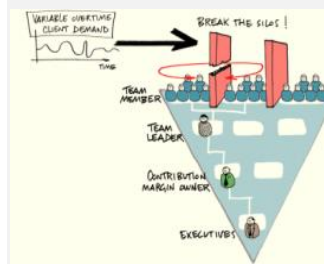
8 Comportamientos Lean para cuestionar el 'Status Quo', **formar** a los equipos y promover la mejora continua

SKILLS MANAGEMENT



Un proceso riguroso que facilite el desarrollo de capacidades y a la vez alinear los Skills del equipo con la demanda del cliente

ORGANIZATION FLEXIBILITY



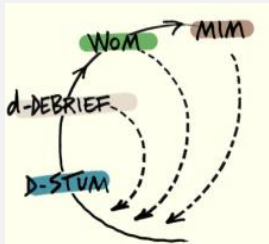
Conjunto de mejores prácticas para romper silos y **acelerar** la generación de valor y reutilización de capacidades.

FLOW MANAGEMENT



Conjunto de principios y prácticas para **alinear** la demanda del negocio con la entrega, y gestionar el flujo de valor extremo a extremo.

OPERATIONS MEETING FRAMEWORK



Sistema de reuniones para gestionar la operativa de forma efectiva e **implantar** la mentalidad de Mejora Continua

VISUAL MANAGEMENT



Un marco estructurado de paneles visuales para compartir problemas y promover la transparencia

OPERATIONS DASHBOARD



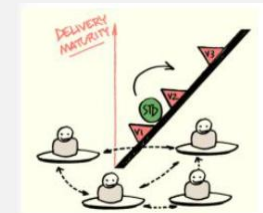
Una selección de KPIs operacionales que facilite una foto real del desempeño e **implantar** la mentalidad de "medir para mejorar"

CONTINUOUS IMPROVEMENT



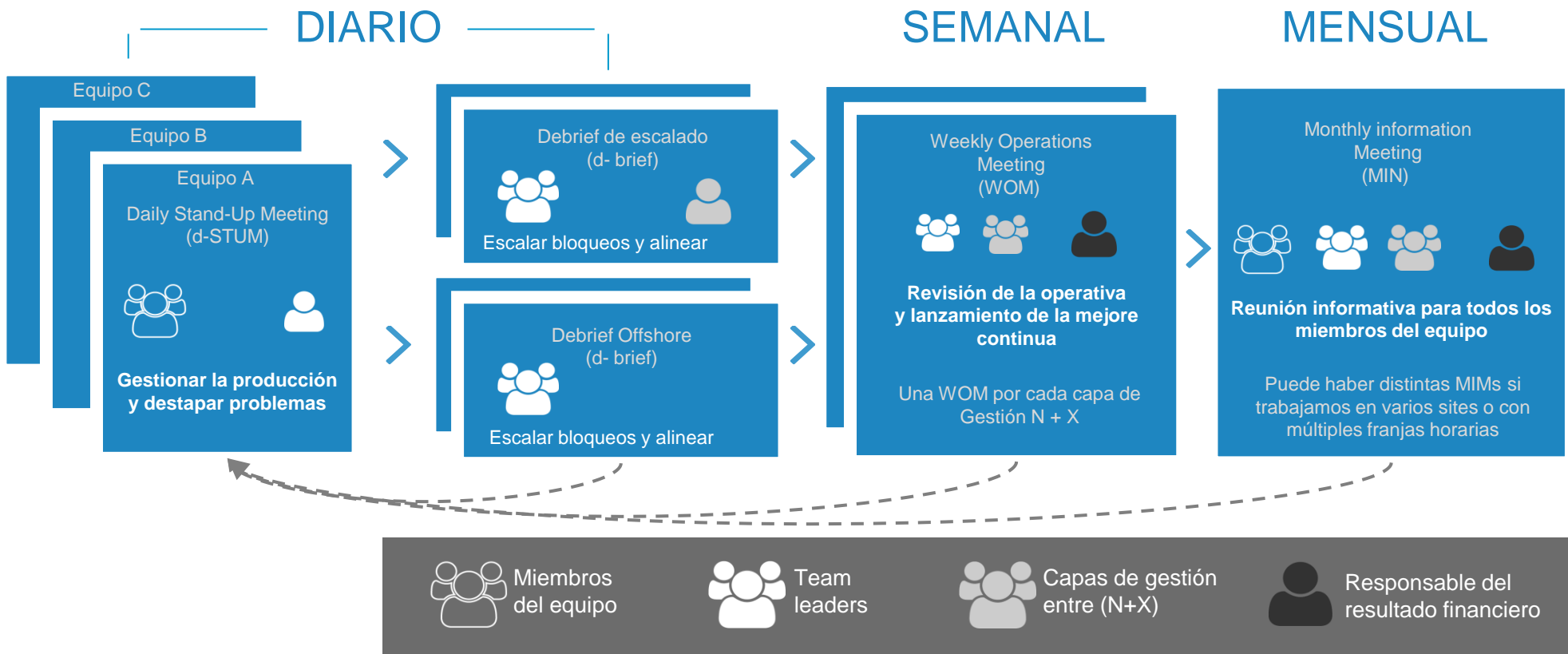
Método que nos permita alinear las mejoras "bottom-up" con los objetivos "top-down" de la organización

STANDARD WORK



Método de trabajo establecido y común para estandarizar las mejores prácticas.

Marco de reuniones para una gestión adecuada de plazos de entrega y control de riesgos dentro de la organización



Con este marco de reuniones se abordan los temas rápidamente, se desarrolla la confianza y transparencia en el equipo y se fomentan comportamientos de mejora continua

Proporcionar visibilidad en el delivery para una gestión más eficiente

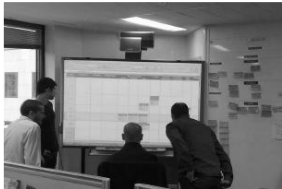


VENTAJAS DEL VISUAL MANAGEMET



Kanban

LA CARGA DE TRABAJO ES VISIBLE
A LO LARGO DEL
CICLO DE VIDA



« Mismo punto de partida,
mismos objetivos »

SE COMPARTE EL MISMO PUNTO DE VISTA
SOBRE EL DÍA A DÍA DE UN EQUIPO



Transparencia

PERMITE ASIGNAR RESPONSABILIDADES A
LOS MIEMBROS DE LOS EQUIPOS



ESTA FILOSOFÍA PERMITE

CONOCER LA EVOLUCIÓN DE LAS
TAREAS EN PROGRESO Y LOS CUELLOS
DE BOTELLA

LA GESTIÓN DE TAREAS ES DÍNAMICA EN
BASE A LA SITUACIÓN ACTUAL

ASEGURA QUE LOS MIEMBROS DE LOS
EQUIPOS ESTÁN ALINEADOS Y
COMPARTEN INFORMACION RELEVANTE
INDEPENDIENTEMENTE DE LA
LOCALIZACIÓN

LA PRODUCTIVIDAD, CALIDAD, TIME TO
MARKET, MOTIVACIÓN DEL EQUIPO,... SE
COMPARTE

TODAS LAS TAREAS TIENEN UN
RESPONSABLE ASOCIADO

TODOS LOS PROBLEMAS TIENEN UN
PLAN DE ACCIÓN DEFINIDO, UNA FECHA
DE EJECUCIÓN Y UN ARTÍFICE DE LOS
MISMOS

El Visual Management aplica a todas las capas de gestión de una organización



CONTENIDO SEGÚN EL TIPO DE REUNIÓN

d-STUM

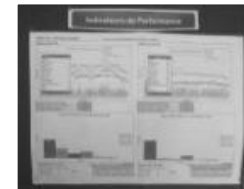
- Estado de ánimo de los integrantes de un equipo
- Kanban de la Producción
- Problemas y Soluciones



en multi sites !

WOM

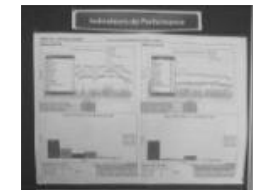
- Indicadores de control,
- Gestión de riesgos escalados
- Seguimiento de Planes de Acción relacionados



KPIs prácticos

MIM

- Voz del Cliente
 - Grado de cumplimiento de SLAs
 - Satisfacción
 - Nuevo negocio
- Indicadores de producción o financieros
- A3s de Mejora continua



Teambuilding



Clara visión general de
nuestro Delivery!

OMF & VVM:
Delivery visible
end-to-end



VVM

UN BOARD POR REUNIÓN

ORGANIZACIÓN EN PANELES
CONFIGURABLES:

Kanban Risk, issues, Actions
Metodología Producción

COMPARTIDO POR TODO EL GRUPO

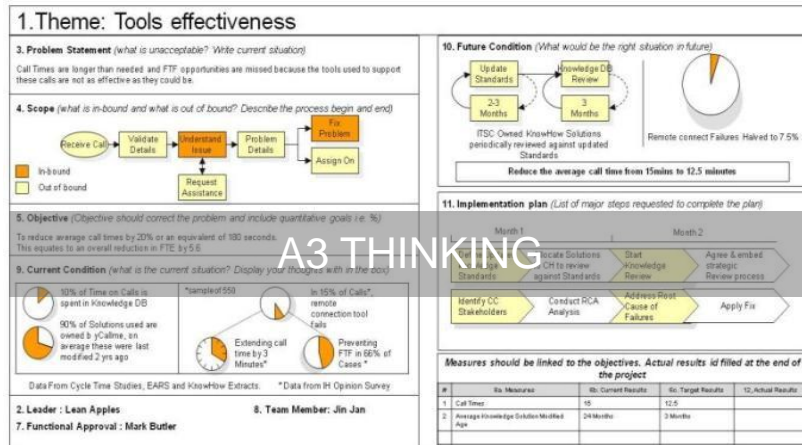
GESTIÓN DE TAREAS – Kanban Ticketing



Incorporar una cultura de mejora continua en la organización



UN ENFOQUE PARA LA RESOLUCIÓN DE PROBLEMAS Y LA MEJORA CONTINUA CLARAMENTE ESTRUCTURADO



A3 THINKING

1. ANÁLISIS DE LA CAUSA RAÍZ
2. CUESTIONAR EL STATUS QUO
3. PROMOVER LA FORMACIÓN Y LA MENTORIZACIÓN
4. ALINEAR LA MEJORA BOTTOM-UP CON LAS METAS DE LA ORGANIZACIÓN
5. APOYAR LA CREACIÓN DE UNA RED DE *SUBJECT MATTER EXPERTS*



UN RIGUROSO PROCESO DE EVALUACIÓN DEL NIVEL DE IMPLANTACIÓN DE LOS FUNDAMENTOS BÁSICOS LEAN CON PLANES DE MEJORA CONTINUA

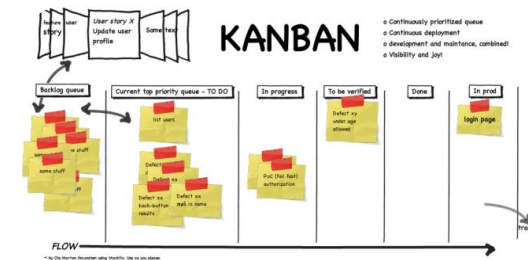


Aplicación en el mercado actual



a Capgemini production

- Para la construcción de SW, Capgemini propone Metodologías Ágiles con el objetivo de dar valor al negocio y optimizar el TCO. Para ello, Scrum-Devon hace uso de técnicas, principios y prácticas, tales como:
 - Colaboración directa del cliente con delivery en los comités funcionales
 - Para lograr más participación e involucración del cliente
 - Identificación y desarrollo de releases cortos e incrementales
 - Para lograr una **reducción temprana del riesgo** (técnico, funcional o de negocio)
 - Llevando un **seguimiento al detalle** del avance del desarrollo
 - Para **mantener la tensión** necesaria en el equipo de desarrollo
 - Ciclos cortos de “Inspección y Adaptación”: El cliente ‘moldea/toca’ la solución
 - Demos al cliente mediante máquinas virtuales portables para que pueda ir tocando el producto
 - Para **mejorar la capacidad de adaptación** ante cambios de prioridades, alcance o requerimientos
 - Para **mejorar el time-to-market y las expectativas del cliente**
 - Uso de mejores prácticas de ingeniería de software con soporte de herramientas
 - Para lograr un **aumento en la visibilidad** del proyecto
 - Confección de equipos multidisciplinares
 - Para garantizar calidad y eficiencia.
 - Con **comunicación diaria** con el equipo: Standup meeting
 - Roles: Product Owner (Analista Funcional), Scrum Manager (Jefe Proyecto) y Equipo.



El usuario es inevitable, escuchemos sus respuestas

¿Porqué es tan importante la participación de UX desde la primera etapa del proyecto?

¿Qué ventajas aporta interactuar con usuarios reales?

¿Cómo reducir los costes a largo plazo mejorando la productividad?

¿Cómo asegurar la continuidad en la excelencia de los diseños?

¿Qué métricas puedo parametrizar sobre el uso de mi aplicación?

UX

En **Capgemini** contamos con un equipo especializado en experiencia de usuario y profesionales en desarrollos de interfaz gráfica que validan el buen comportamiento funcional de las aplicaciones. Cuidamos cada detalle de nuestros productos para garantizar la calidad en las entregas, procurando la satisfacción de los usuarios. Para ello ofrecemos:

Usabilidad y accesibilidad

- Es la claridad y la elegancia con la cual se diseña la interacción de un sistema para alcanzar unos objetivos específicos de efectividad, eficiencia y satisfacción.

Arquitectura de la información

- Es la ciencia de organizar los espacios en estructuras de navegación con el fin de ayudar a los usuarios a satisfacer sus necesidades de información.

Prototipado y diseño

- Proponemos la comunicación directa y continua mediante prototipos para validar los procesos de interacción con el usuario.

Test con usuarios

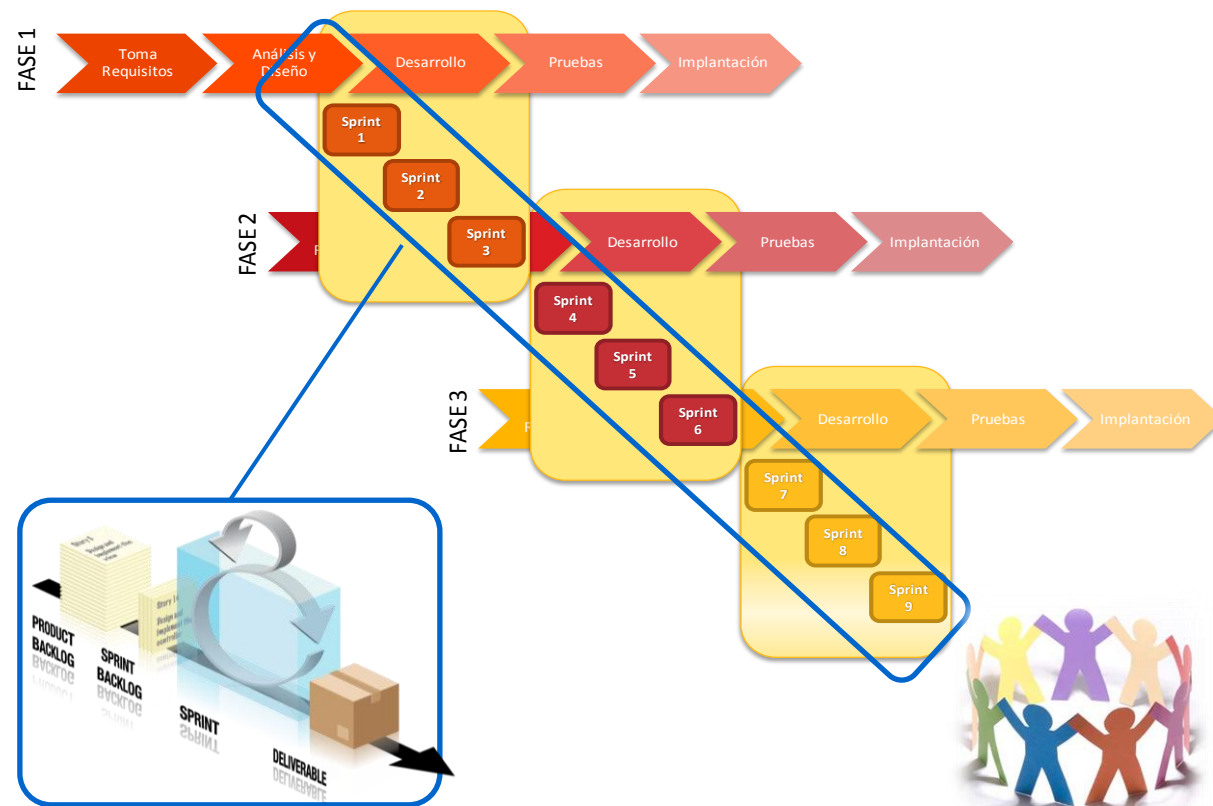
- Son pruebas de usabilidad que se basan en la observación y el análisis de cómo un grupo mínimo de usuarios reales utiliza los sistemas. Anticipan problemas comunes de uso, localizando mejoras evolutivas y optimizando la productividad de los componentes gráficos de interacción.

Guías de estilo

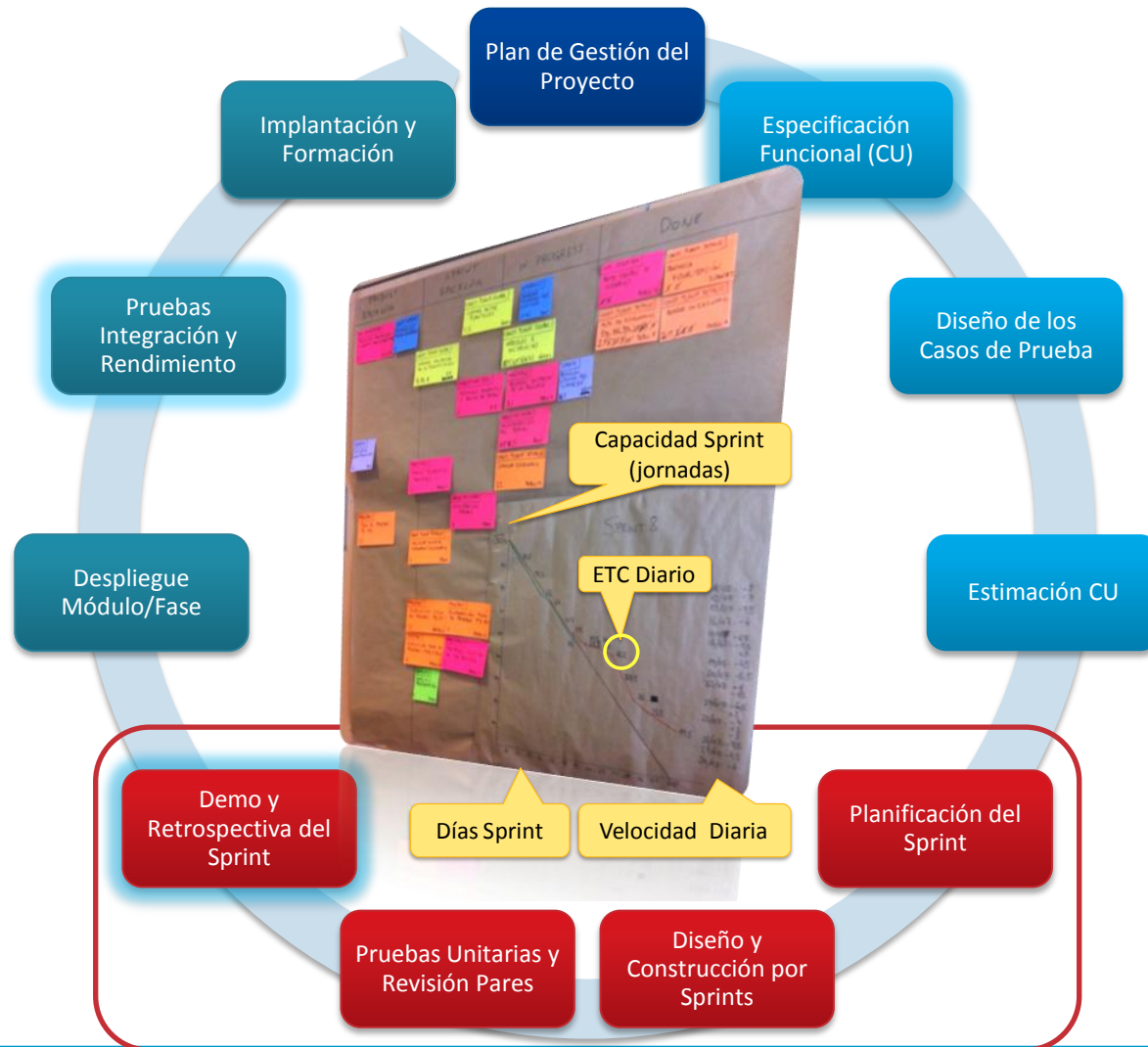
- Desde UX recomendamos la utilización de guías de estilos alineados a desarrollo, agilizando los cambios ante cualquier tipo de evolución gráfica o funcional.

Devon: Metodologías Ágiles en construcción

- La experiencia en múltiples proyectos nos ha hecho llegar a una **'metodología mixta'**, entre la tradicional y la **ágil**, que permite obtener los objetivos marcados en la estrategia:
- Dentro del ciclo de vida del software, **solo se aplica a la fase de desarrollo**.
- Se sigue trabajando por fases**, pero buscamos el **solapamiento** del desarrollo mediante los **sprints**.
- En la fase de análisis se especifica y diseña en detalle los requisitos analizados, pero teniendo **siempre la visión global** del producto (Product Backlog).
- Aunque se sigue una planificación por fases, **se permite cierta flexibilidad** en la adaptación a los cambios en cada uno de los sprints.
- Las **entregas del producto** obtenido en los sprints no se lleva a producción, **son versiones 'de desarrollo'** para revisión con el cliente / usuario final o venta interna.



Método Ágil en la construcción

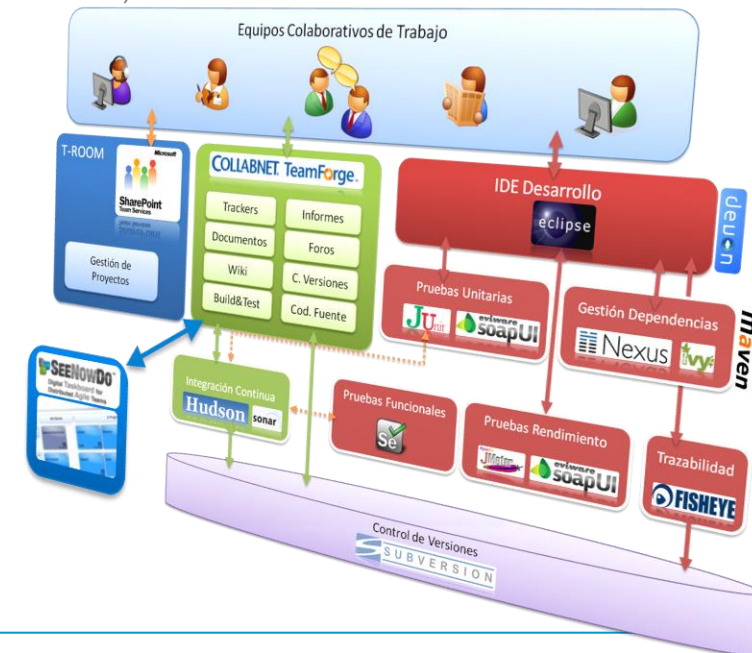


- Ejecución de Pruebas Unitarias (**Casos de prueba y Revisión de Pares**):
 - Programación de las Pruebas Unitarias de los CU, y ejecución de las mismas (CP/RP).
 - Ejecución de los casos de prueba funcionales, y revisión de pares, de forma cruzada por el equipo.
 - Debe ser rotatoria para asegurar la calidad, pero al mismo tiempo aprender de los compañeros.
- Ejecución de Pruebas de Calidad estática del Software (**Sonar**):
 - Captación, análisis y explotación de métricas del código para la gestión de su calidad.
 - Objetivos:
 - Medir la complejidad y tamaño de la aplicación.
 - Controla las violaciones de arquitectura.
 - Comprobar de reglas de consistencia, robustez y mantenibilidad del código (PMD)
 - Corregir evidencias y mejorar el código.
 - Lograr asegurar y garantizar las mejores prácticas para la mejora continua del equipo de desarrollo.
 - Gestionar y consolidar el conocimiento en las Wikis de Proyecto y/o en la del framework (**Bits**)
- Ejecución de Pruebas de Integración continua (**Jenkins**):
 - Integración continua del código desarrollado para detectar errores con la mayor antelación posible.
 - Cada commit: Se descarga el proyecto, se compila, detecta si hay fallos de compilación y lanza los jUnits.
 - Enlace con el Sonar: Cada hora, se descarga el código y se pasan los test de calidad de código.
- Pruebas de rendimiento: volumetría y entorno equivalente a PRODUCCIÓN (**jMeter**).



Mismo entorno de desarrollo: herramientas ágiles de calidad

- Herramientas relacionadas con la gestión de proyectos:
 - **T-ROOM**, para la gestión del proyecto y servir de mecanismo de comunicación entre los miembros del equipo de trabajo.
 - **TeamForge**, para seguimiento de tickets, reporting y trabajo colaborativo.
- **Entorno único de desarrollador:** Herramienta de desarrollo (**Eclipse**).
- Herramientas de pruebas y calidad : proponiendo el empleo de tecnologías Open-Source, tanto para el desarrollo de software como para las actividades de certificación de calidad.
 - **Hudson/Jenkins** y **Sonar**: Integración continua y la medición de la calidad del SW.
 - **JUnit** para pruebas unitarias de Java, **Selenium** para pruebas de la capa de vista, y **soapUI** para las pruebas de los webservices.
- Herramienta **JMeter** para las pruebas de rendimiento y carga.
- Herramienta de control de versiones (**Subversion**).
- Herramienta para garantizar el control de dependencias con la herramienta **MAVEN**, así como **NEXUS**.
- Herramienta para el trazado y análisis de las modificaciones de código fuente.
- Pruebas automáticas de capa vista con **Sahi**.





Pregunta & Debate

ggabinod@capgemini.com

www.uv.es/capgeminiuv ← web de la cátedra con la PPT

www.agilemanifesto.org

www.javiergarzas.com ← el GURÚ 😊

blog.leanmonitor.com

| Desarrollo Tradicional | Desarrollo Ágil |
|------------------------|--------------------------|
| Especialización | Equipo multidisciplinar |
| Fases | Solapamiento |
| Requisitos detallados | Visión del Producto |
| Seguimiento del Plan | Adaptación a los cambios |