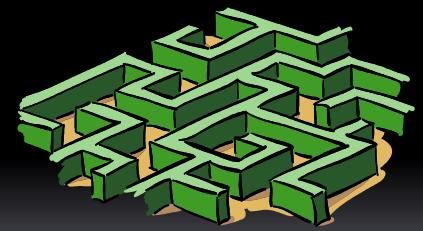


SCRUM y La Gestión de Proyectos WEB

Primera parte: Teoría orientada a la práctica

Por Jaume Ramonet (www.jramonet.com)

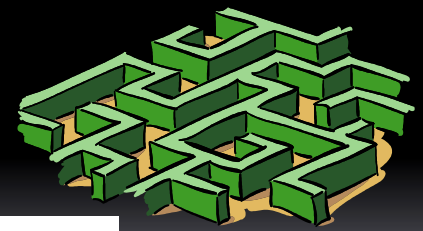




Programa de la sesión:

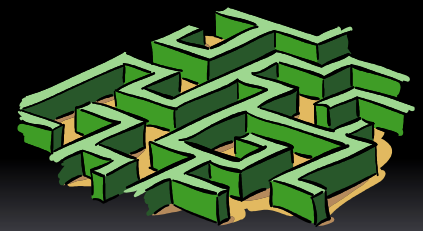
- Introducción.
- Los proyectos y su gestión.
- Taxonomía de los proyectos y de los métodos de gestión.
- El paradigma clásico frente al nuevo paradigma.
- Metodologías que responden al nuevo paradigma.
- Aplicación del nuevo paradigma a los proyectos **WEB**: ¿un caso de simbiosis mutualista?
- **SCRUM**: principios, metodología, técnicas y herramientas.
- **SCRUM**: Fortalezas, debilidades y factores clave de éxito.
- **SCRUM** y **Drupal**: ¿una pareja perfecta?.





Introducción:

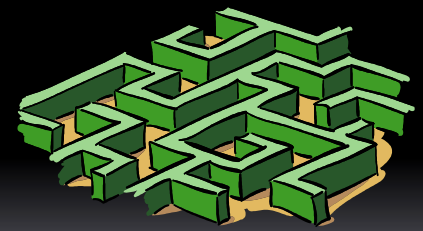




Quien soy . . . (1)

- Nací en Barcelona el año 1955 (tu mismo...)
- Ingeniero Industrial en ejercicio libre desde 1985.
- Nunca he tenido un Spectrum ZX, ni un Commodore VIC-20, ni un C-64, ni un Amiga, ni siquiera un Apple II...
- "Mi" primer ordenador fue, en 1976, el *Eclipse (Data General)* de la E.T.S.E.I.B. Y primer lenguaje el *Fortran IV* (PFC: + de 10.000 líneas en *Fortran 77*).
- 1977 - 1995: en temas relacionado con el desarrollo y los sistemas basados en UNIX, "C" y BB. DD. Relacionales.
- 1993 - Libro "*UNIX System V: Ordenes de usuario y programación Shell*"
- Otros S.O. Multiusuario: CP/M- 86, Oasis/Theos, PICK,...

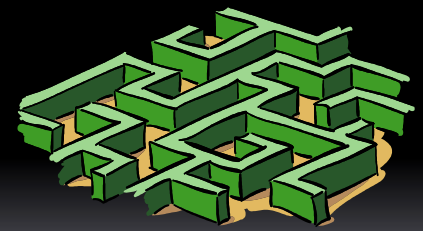




Quien soy . . . (2)

- [En 1989 hice *demos* del *Apple NeXT Cube* en diversas ferias españolas: *¿friki yo?*]
- Desde 1995: Consultor y Formador en temas de:
 - Gestión de Proyectos.
 - Gestión de Procesos y Calidad.
 - Aplicaciones estratégicas de las N.T.'s.
 - Y ahora... WEB 2.0 con **Drupal**.
- Y en 2010... luchando contra la *entropía*.
- Miembro de **DRUPAL.CAT** y de la *Drupal Association*.
- Mi *nick* en **Drupal**: *isaac.el.cec*

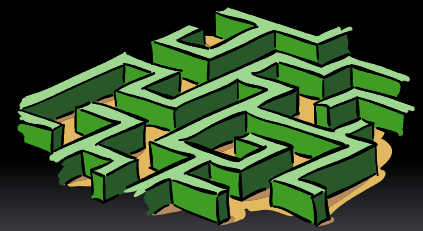




Qué pretendo . . . (1)

- Esta sesión no es formación específica de **SCRUM**. Pretendo entender en que se basa y como funciona **SCRUM**...
- No soy especialista en *metodologías ágiles* de desarrollo de software.
- Sí soy especialista en *metodologías* de desarrollo de proyectos en diversos ámbitos (Ingeniería simultánea, p.e.).
- Pretendo realizar una *aproximación pragmática* a **SCRUM** desde la **teoría** (!) de la Gestión de Proyectos, ... porqué:





Que pretendo . . . (2)

“No hay nada más práctico
que una buena teoría”

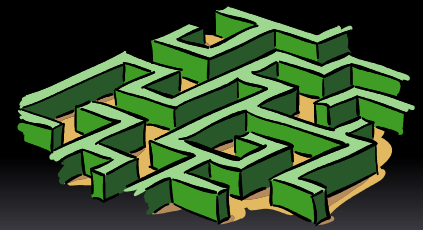
Kurt Lewin

Mis motivaciones:

- Una de las mejores formas de aprender es tener que enseñar.
- La Dirección y Gestión de Proyectos no es una ciencia exacta.
- Existen muchas formas de Gestionar los Proyectos y quiero conocer otras.
- Aprovechar el efecto sinérgico de los conocimientos y experiencias del grupo . . .

*“Los sabios aprenden de las experiencias de los demás,
Los inteligentes aprenden de sus experiencias,
Los estúpidos no aprenden nunca”.*



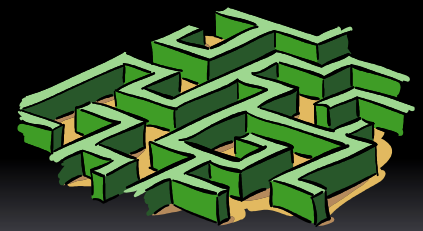


Mi deseo sobre esta sesión:



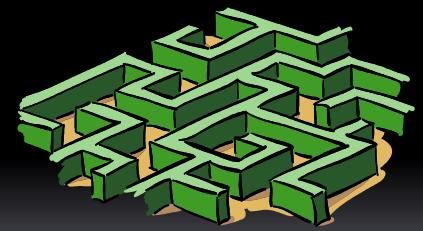
*"Donde todos piensan igual,
nadie piensa mucho"*
Voltaire



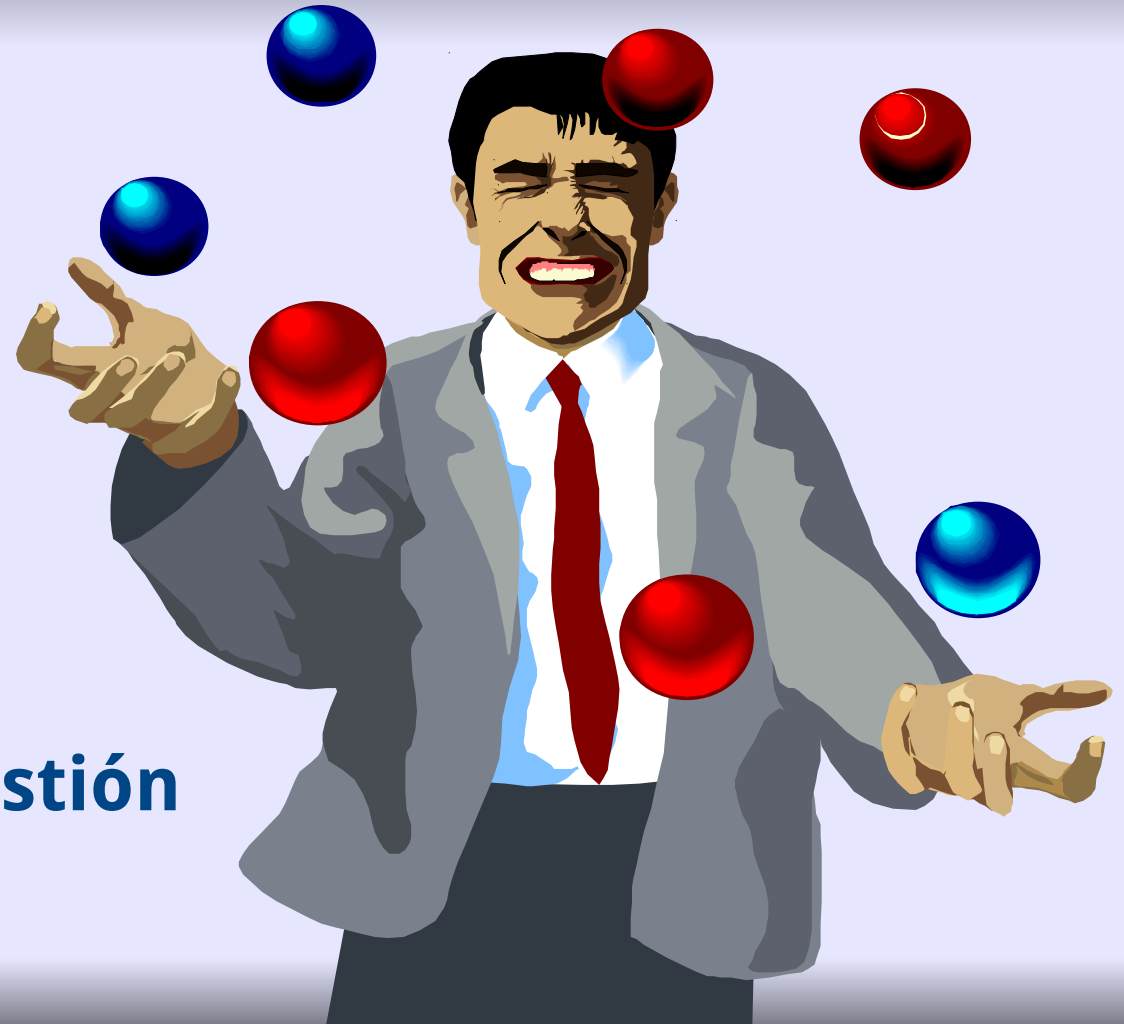


¿Y vuestras expectativas?



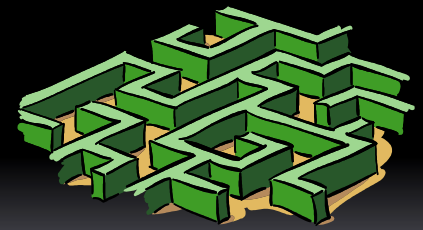


Primera Parte:

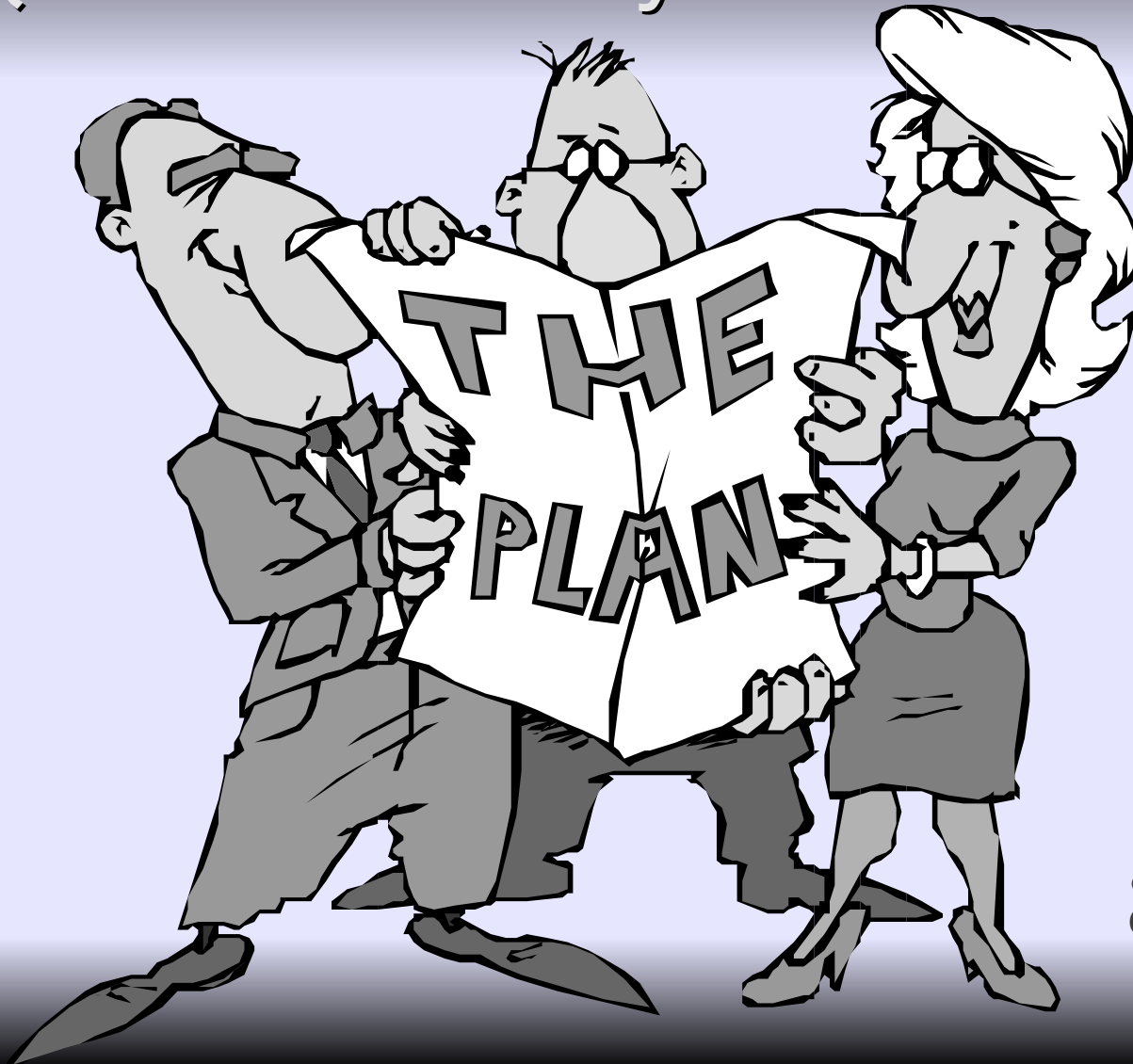


Los Proyectos y su gestión



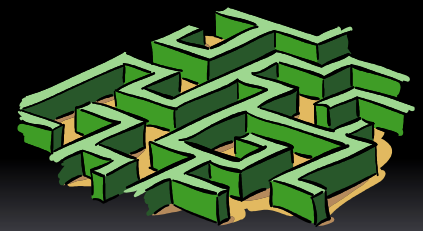


¿Que es un Proyecto?



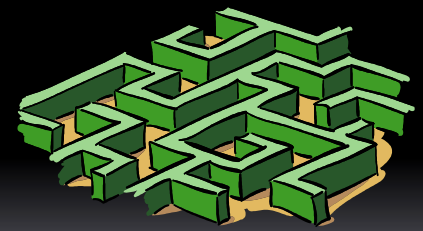
¿A quien le importan las definiciones oficiales?





El famoso Triangulo del Proyecto





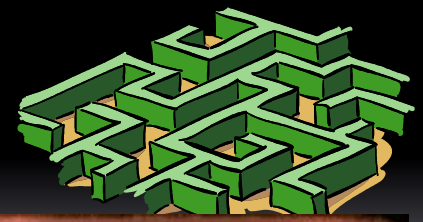
Gestión de Proyectos:

Conjunto de técnicas y herramientas que permiten:

- **Planificar, Organizar, Coordinar, Realizar, Seguir, Controlar e informar** de todos los aspectos de un proyecto y **MOTIVAR** a todos aquellos que estén involucrados en la consecución de los objetivos (*Resultado*).

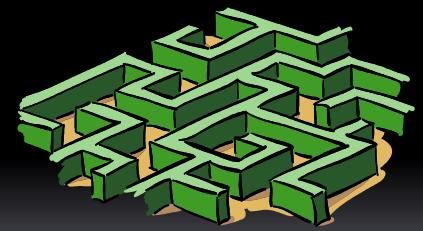
Adaptado de ISO 10.006, Aptd. 3.6



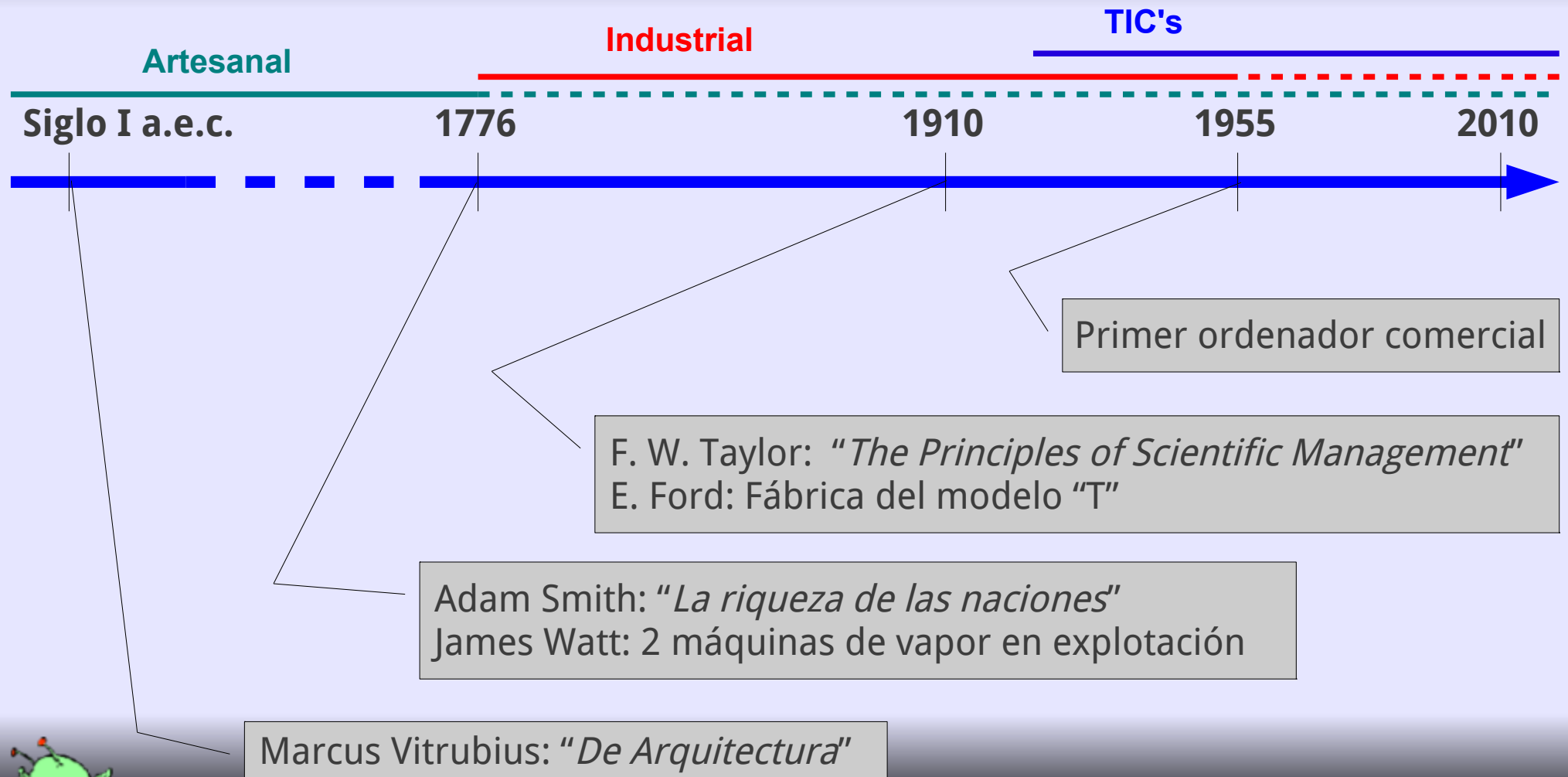


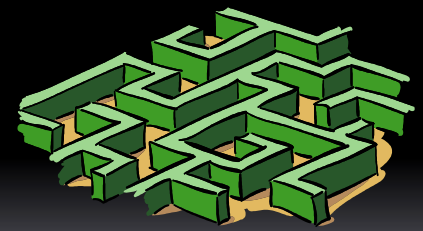
Ejemplo de RESULTADO de un PROYECTO
¿Qué *metodología* se utilizó?





Quien no conoce la historia está
condenado a repetirla . . .



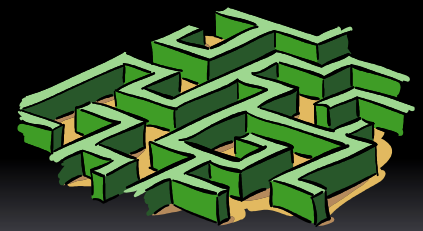


Niveles de Capacidad y Madurez (CMM) (casi la misma historia...)

- Nivel 0: Improvisación (!).
- Nivel 1: Artesanal.
- Nivel 2: Industrial 1a. revolución.
- Nivel 3: Industrial 2a. Revolución.
- Nivel 4: Proactivo.
- Nivel 5: Excelencia.

**¿Que nivel tenéis en la
la Gestión de Proyectos?**



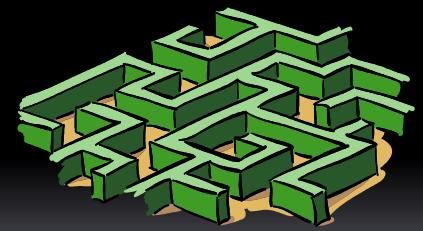


Taxonomía de los proyectos:

- Hay “ $n++$ ” posibles criterios!!!
- Criterios significativos para nuestro punto de vista:
 - **Cliente:** Interno «-» Externo.
 - **Complejidad:** Simple «-» Complejo.
 - **Objetivo final:** Cerrado «-» Abierto.
 - **Plan:** Determinista «-» Heurístico.

Pregunta:
¿Todos se deben gestionar igual?





Cliente: Interno «-» Externo

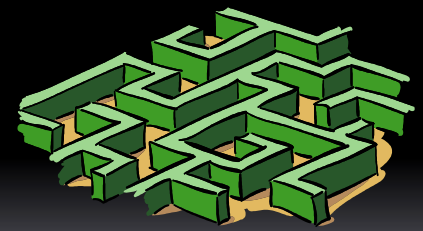
Cliente Interno:

- ¿uno mismo?
- Presión relativa si resultados no económicos.
- Organización informal.
- Poca metodología y mucha *política*.
- (!) Multi-proyecto.
- Comunicación matizada por las *relaciones*
- Control focalizado en "resultados".

Cliente Externo:

- (!) Precio cerrado (o no)
- Presión sobre: costos, plazos, resultados, ...
- (!) Comunicación con TODAS las partes interesadas.
- (!) Relación con el cliente.
- (!) Gestión *imprescindible* de los "poyaques".
- Exigencia de resultados ECONÓMICOS.
-

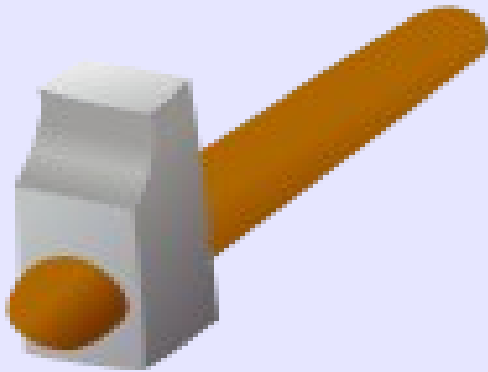




Complejidad: Simple «-» Complejo

Simple:

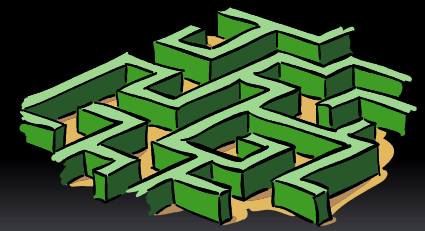
- !!! Que suerte iii
- . . .



Complejo:

- Multi-Proyecto.
- Múltiples partes interesadas con intereses divergentes
- Múltiples *especialidades* involucradas.
- (!) Método del Análisis Cartesiano (incremento de la complejidad por las interrelaciones).
- . . .





Objetivo final: Cerrado «-» Abierto

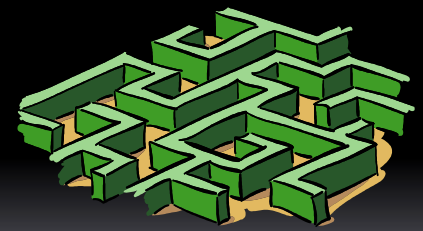
Objetivo cerrado:

- Ingeniería de Requisitos.
- (!) Especificación de los Requisitos: ¿correctos?, ¿completos?, ... ¿verificables?
- (!) ¿Verificables?: Criterios de Aceptación.
- Focalizado en la Planificación detallada y posterior seguimiento.

Objetivo abierto:

- Definición incremental del objetivo.
- (!) Gestión de la participación de las partes (*negociación*).
- Flexible, adaptable, dinámico.
- Focalizado en la comunicación y coordinación entre las partes.





Plan: Determinista «-» Heurístico

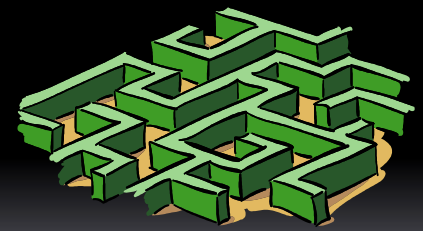
Determinista:

- Relaciones *Causa - Efecto* claras y precisas.
- Diagrama de GANT o PERT completo y detallado.
- Plan y presupuesto "cerrados".
- Ajuste al plan posibles (poco probable dado que *Murphy* esta en nómina!!!)
- Objetivos con gran carga "técnica" . . .

Heurístico:

- Acercamiento incremental hacia el objetivo a partir de estrategias basadas en la creatividad, pensamiento lateral, pensamiento divergente, etc.
- (!) Basado en métodos colaborativos (grupo).
- Objetivos con gran carga "creativa" o dinámica.
-





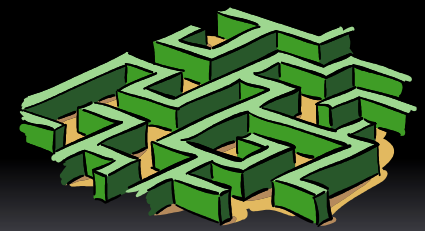
Reglas heurísticas de George Pólya

- Si no consigues entender un problema, dibuja un esquema.
- Si no encuentras la solución, haz como si ya la tuvieras y mira qué puedes deducir de ella razonando hacia atrás, a la inversa.
- Concretar: Si el problema es abstracto, prueba a examinar un ejemplo concreto.
- Generalizar: Intenta abordar primero un problema más general ("*paradoja del inventor*": el propósito más ambicioso es el que tiene más posibilidades de éxito).

¿No os suena?

Adaptado de <http://es.wikipedia.org/wiki/Heurístico>



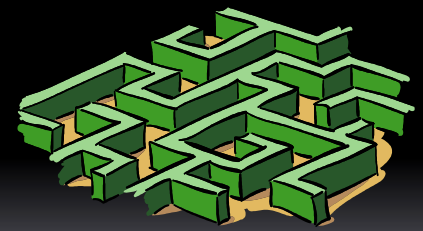


Los Grandes Proyectos del siglo XX: origen del paradigma *clásico*

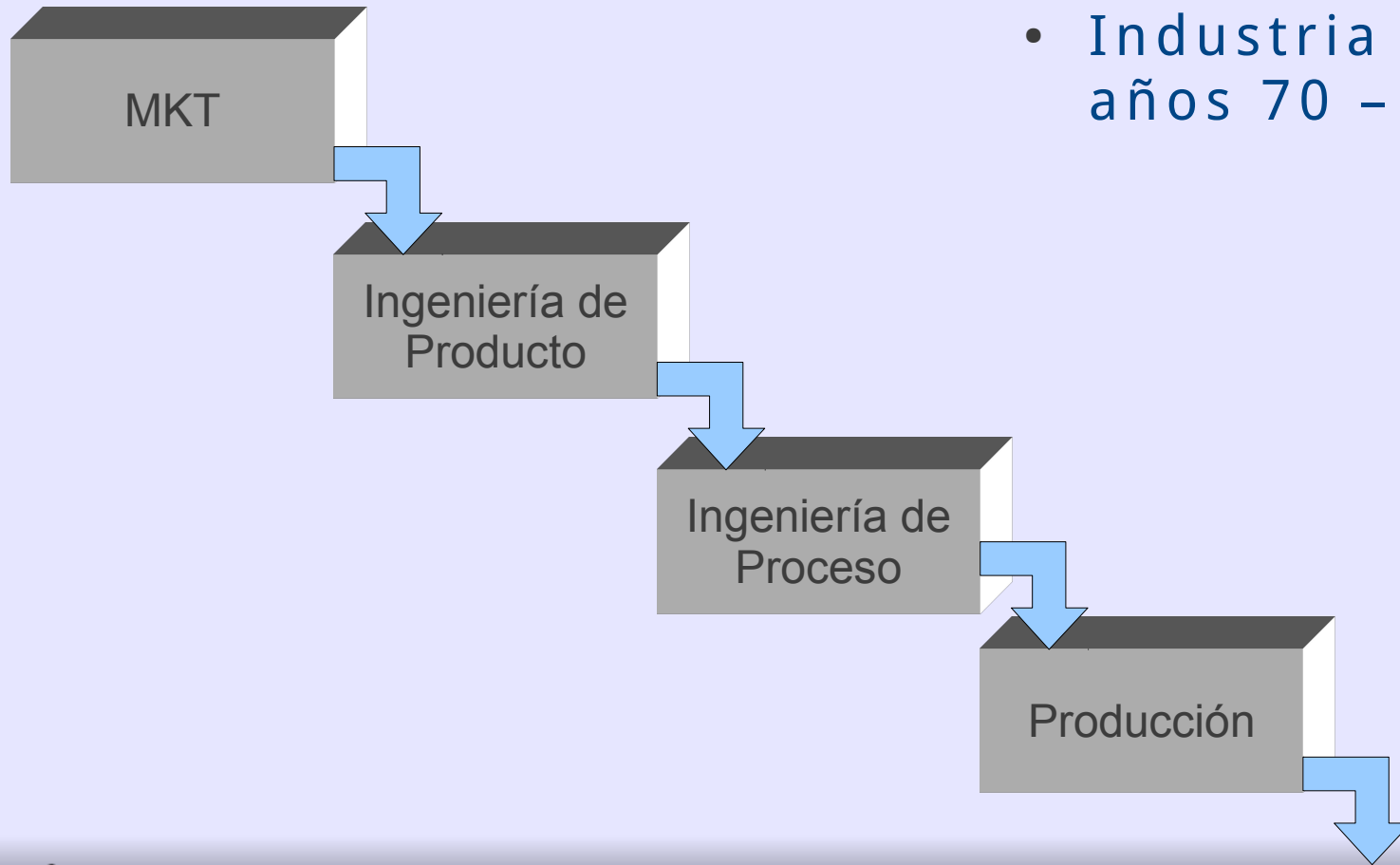
- 2 guerras mundiales (!): desarrollo espectacular de barcos, aviones, vehículos, armas diversas,...
- Proyecto *Manhattan* (1939 - 1952).
- Proyecto *Apolo* (1960 - 1971).
- Las grandes plataformas petrolíferas y los grandes buques (1960 - 1990)
- Usos pacíficos de la Energía Nuclear (1955 - 1990)
- Los edificios "singulares" (1930 - ...)

Aviso: Paradigma "clásico" <> "obsoleto" . . .





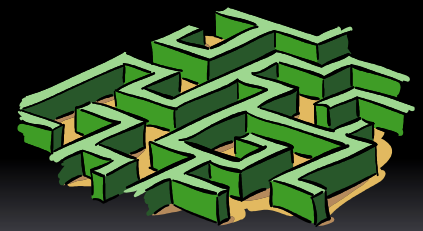
Diseño y desarrollo según el paradigma clásico:



- Industria occidental años 70 - 80 ...

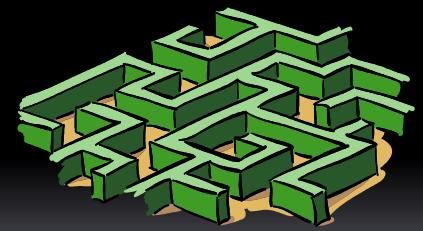
¿Os suena?





Ciclo del proyecto según el paradigma "clásico":

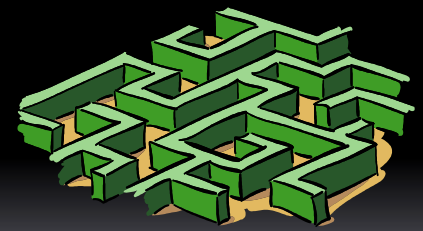




Pilares del paradigma clásico:

- La Ingeniería de requisitos.
- Las técnicas y herramientas de planificación
- Las técnicas y herramientas de estimación del esfuerzo.
- Las técnicas y herramientas de seguimiento / control y gestión de acciones correctoras.
- Las técnicas y herramientas de gestión de las modificaciones y gestión de la configuración.
- . . .

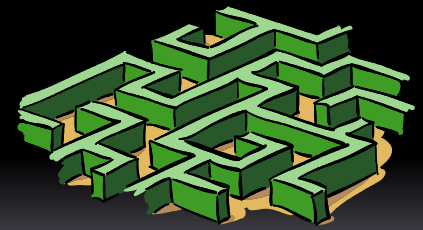




Adalides del paradigma clásico

- PMBOK[®] del PMI[®] (www.pmi.org)
- ISO 10006 e ISO 21500 (www.iso.org)
- MERISE, METRICA 3 y otros...
- . . .



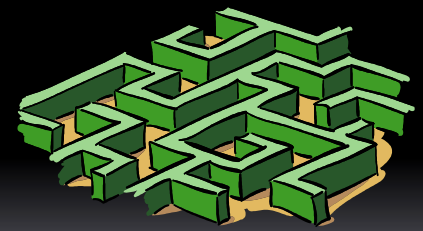


El nuevo paradigma: Diseño y Desarrollo Concurrente.

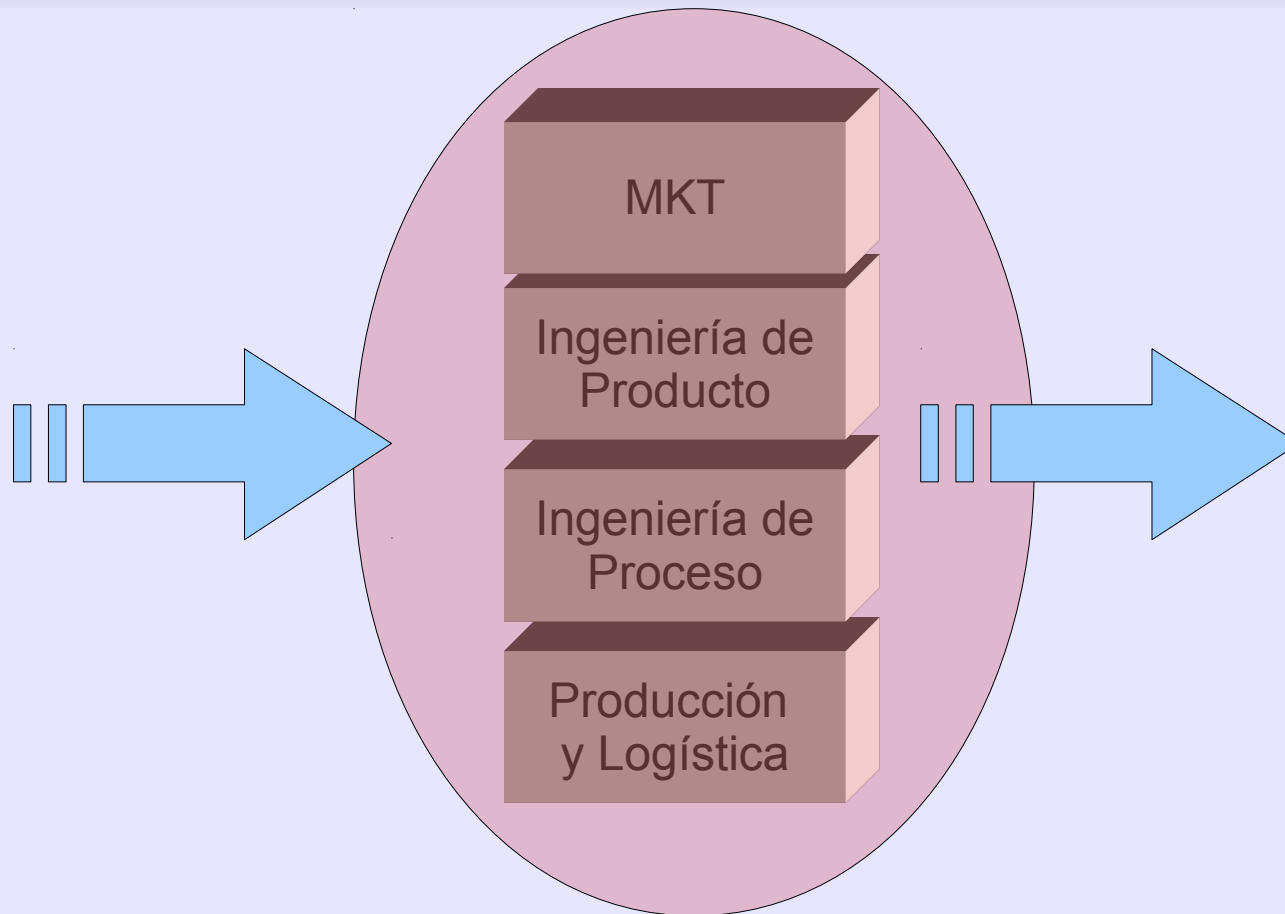


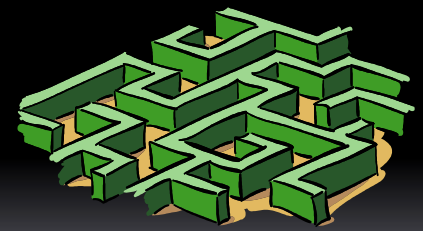
- Revolución del Japón (60 - 70's)
- Reducción «dramática» del *Time to Market*.
- Múltiples metodologías, técnicas y herramientas:
 - QFD
 - SCRUM
 - DFMA
 - AMFE
 - Diseño de experimentos (Taguchi)
 - KAIZEN
 - KANBAN
 -





El nuevo paradigma para el Diseño y Desarrollo: Ingeniería Concurrente

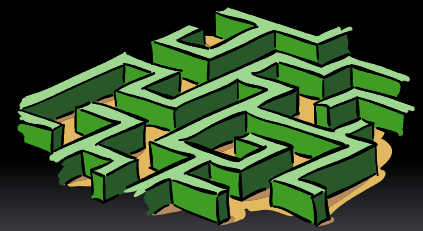




Bases del nuevo paradigma:

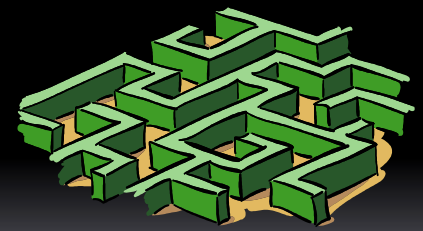
- Equipo multifuncional.
- Líder de equipo muy competente.
- Coordinación muy exigente.
- Cultura de participación abierta y sincera. No barreras organizacional.
- Comunicación, rigor y profesionalidad.
- Confianza entre miembros equipo.
- Formación en ámbitos *soft*.
- Trabajadores del conocimiento conscientes de su labor.
- Colaboración en la cadena "cliente - proveedor".
- Entorno inestable que requiera rapidez y flexibilidad.
- . . .





Tipo de proyecto y Paradigma de Gestión





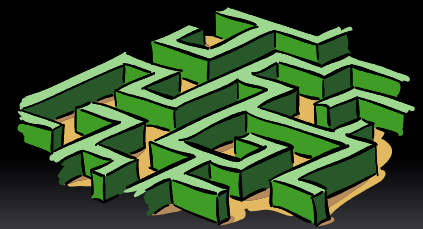
Técnicas y Herramientas para el
nuevo paradigma . . .

“Quien conoce el porqué
encontrará el como”

“Crítica de la razón práctica” - E. Kant (probablemente apócrifo)

Vamos ahora a centrarnos en “nuestro” sector



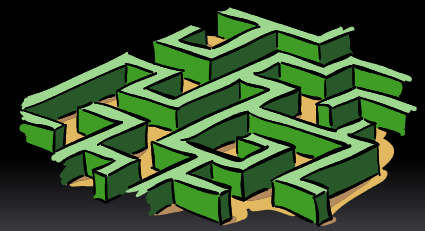


Software: Metodologías para el nuevo paradigma

The screenshot shows a Wikipedia page with the following structure:

- Navigation:** Main page, Contents, Featured content, Current events, Random article.
- Search:** A search box with 'Go' and 'Search' buttons.
- Interaction:** About Wikipedia, Community portal, Recent changes, Contact Wikipedia, Donate to Wikipedia, Help.
- Toolbox:** What links here, Related changes, Upload file, Special pages.
- Article Content:**
 - Buttons: article, discussion, edit this page, history.
 - Title: List of software development philosophies
 - Text: From Wikipedia, the free encyclopedia
 - Text: This is a list of approaches, styles, and philosophies in software development.
 - List of philosophies:
 - Agile software development
 - Agile Unified Process (AUP)
 - Aspect-oriented Programming
 - Behavior Driven Development (BDD)
 - Big Design Up Front (BDUF)
 - Blind Men And Elephant Approach (BMAEA)
 - Brooks's law
 - bytecode/interpreter platform independent code
 - Cathedral and the Bazaar (see also Release early, release often)
 - Code and fix
 - Cone of Uncertainty
 - Constructionist design methodology (CDM)
 - Continuous integration
 - Control tables
 - Conway's Law
 - Cowboy coding
 - Crystal Clear
 - Dependency injection
 - Design-driven development (D3)
 - Domain-Driven Design (DDD)
 - Don't repeat yourself (DRY) or Once and Only Once (OAOO), Single Point of Truth (SPoT)
 - Dynamic Systems Development Method (DSDM)
 - Evolutionary Model
 - Extreme Programming (XP)
 - Feature Driven Development





Metodologías para el nuevo paradigma Manifiesto Ágil (marzo 2001)

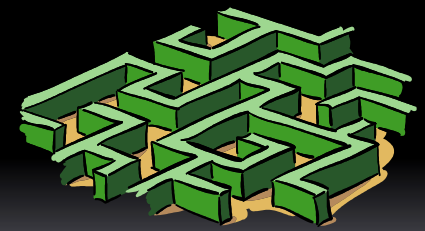
Estamos poniendo al descubierto mejores métodos para desarrollar software, haciéndolo y ayudando a otros a que lo hagan. Con este trabajo hemos llegado a valorar:

- A los individuos y su interacción, por encima de los procesos y las herramientas.*
- El software que funciona, por encima de la documentación exhaustiva.*
- La colaboración con el cliente, por encima de la negociación contractual.*
- La respuesta al cambio, por encima del seguimiento de un plan.*

Aunque hay valor en los elementos de la derecha, valoramos más los de la izquierda.

Fuente: http://es.wikipedia.org/wiki/Manifiesto_Ágil





Proyectos WEB y el nuevo paradigma: ¿Un caso de simbiosis mutualista?

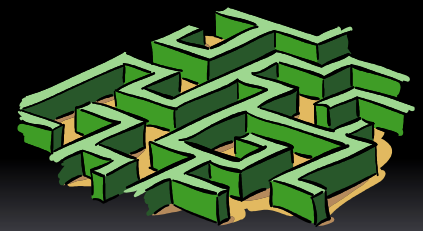
- Nuevo Paradigma:

- Objetivo abierto
- Plan Heurístico
- Equipo Multidisciplinar de trabajadores del conocimiento.
- Colaboración "Cliente - Proveedor"
- Comunicación intensa
- Coordinación "dura"
- Rigor y profesionalidad

- Proyectos WEB

- ¿Objetivo?
- ¿Plan?
- ¿Equipo?
- ¿Necesaria la colaboración "cliente - proveedor"?
- ¿Comunicación?
- ¿Coordinación?

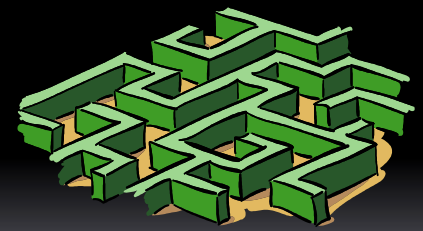




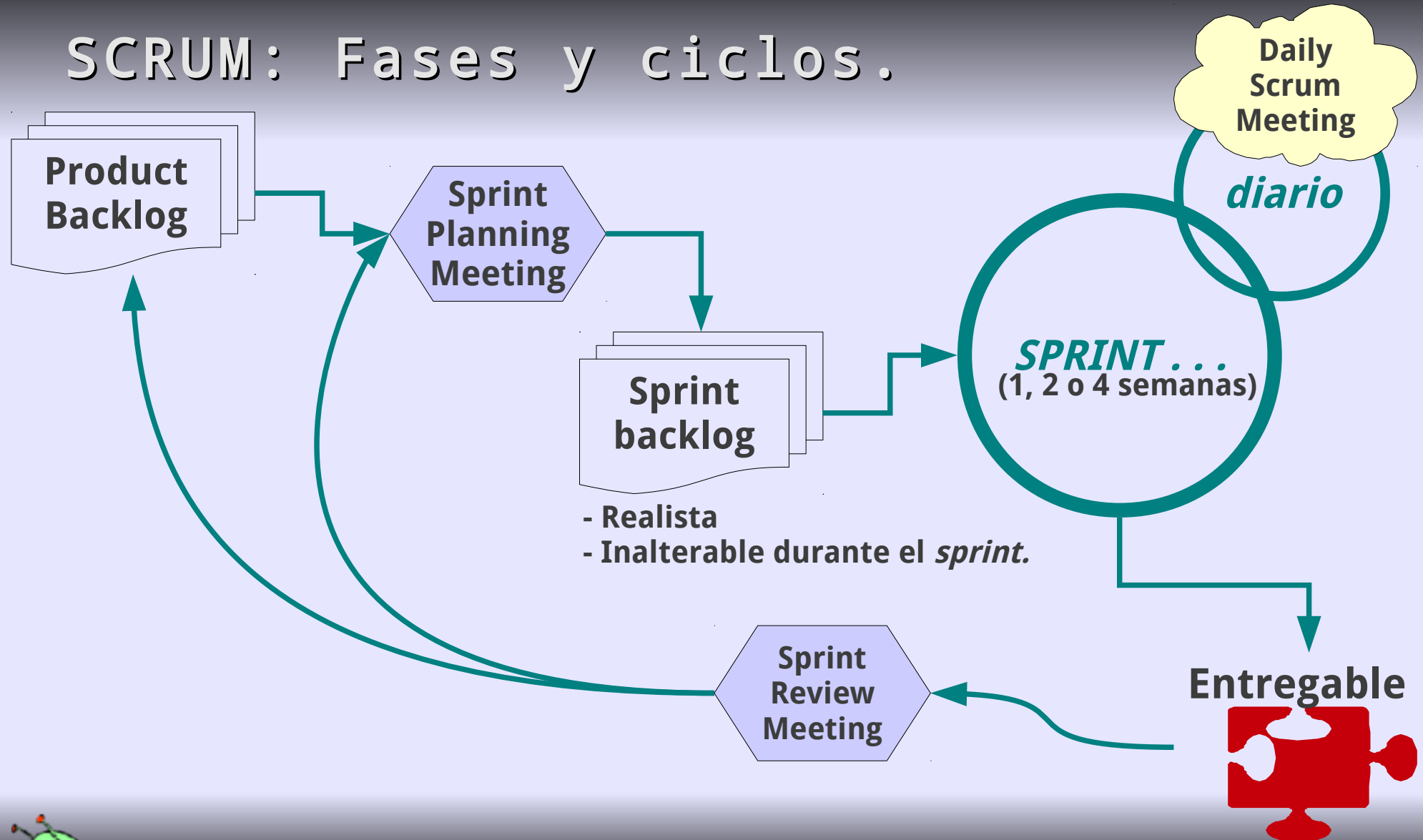
SCRUM

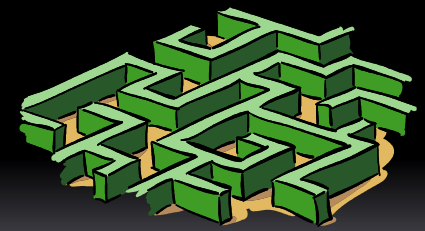
¿La «*bala de plata*»?





SCRUM: Fases y ciclos.

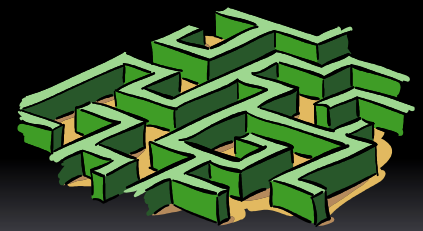




SCRUM: Principios, métodos, técnicas y herramientas . . .

- **SCRUM** (del argot del rugby: "*melé*"): Proceso iterativo e incremental de diseño y desarrollo.
- Origen: Japón 1986 - desarrollo de producto.
- Occidente: 1991. Primer referencia al nombre "*scrum*".
- Punto Fuerte: Define una serie de roles, de actividades y de documentos de soporte bastante estructurados.
- Equipo multidisciplinar transversal pequeño.
- Comunicación verbal entre los miembros del equipo.
- Punto Fuerte: Considera que un proyecto puede sufrir modificaciones en su objetivo a lo largo de su realización.

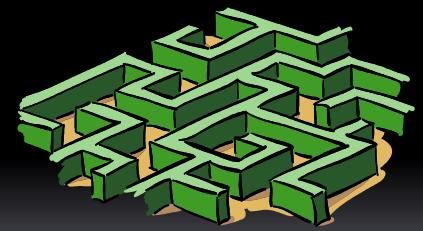




SCRUM: *Daily Scrum Meeting*

- Reunión diaria del equipo. Informal. De pie.
- Tiempo máximo 30' (con 15' ha de ser suficiente).
- Las 3 *auto*-preguntas de cada miembro del equipo:
 - ***¿Qué hice ayer?*** (seguimiento de tareas o requisitos completados por el *Scrum Master*)
 - ***¿Qué voy ha hacer hoy?*** (elección de tareas por los miembros del equipo)
 - ***¿Qué dificultades, obstáculos o riesgos preveo?***
/ ¿Qué ayuda necesito? (Soporte y coordinación por el *Scrum Master* = facilitador).



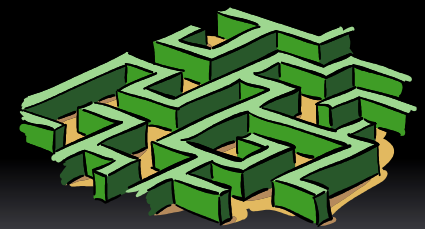


SCRUM: Roles *comprometidos*



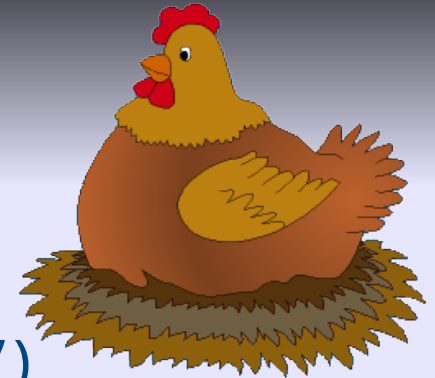
- El **Product Owner**: Representa "*la voz de las partes interesadas*". Establece las prioridades sobre las especificaciones (historias de usuario) en el **Product Backlog**.
- El **Scrum Master**: Lider, coordinador y facilitador del equipo de **scrum**. Experto en **SCRUM** (y en Gestión de Proyectos !)
- Los miembros del **Equipo**. Responsables de la realización de las tareas a las que se comprometan en cada **spring**.
- Los roles comprometidos *ponen su carne* en el proyecto...

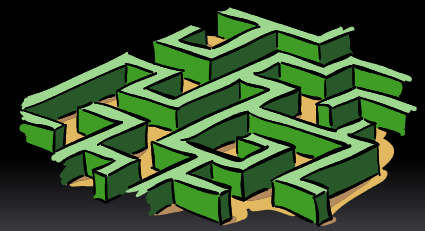




SCRUM: Roles *involucrados*

- Los **usuarios**.
- Otras **partes interesadas** ("*stakeholders*")
- Los **expertos** en aspectos concretos del tema . . .
- No forman parte del proceso de **SCRUM** pero sus opiniones son tenidas en cuenta: es muy importante que aporten retroalimentación en las revisiones del *sprint*.
- Su "abogado" es el **Product Owner**.
- Los roles involucrados no se juegan gran cosa: se tiene en cuenta su opinión pero no se permite que afecten, distorsionen o entorpezcan el proyecto.

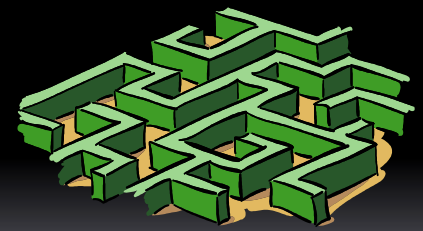




SCRUM – Documentos: *Product Backlog*

- Especie de cuaderno de especificaciones genéricas del producto final (requerimientos, funcionalidades, etc.). Es el documento de + alto nivel del proyecto.
- Cada especificación es priorizada por el **Product Owner** (responsable del documento) desde el punto de vista del valor aportado al cliente / negocio.
- Cada especificación es valorada la dificultad de su consecución (plazo) por el **Scrum Master**.
- El plan temporal se ajusta en función de estos dos valores.
- Documento abierto a las aportaciones de todos los roles.

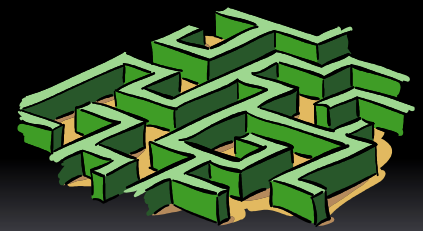




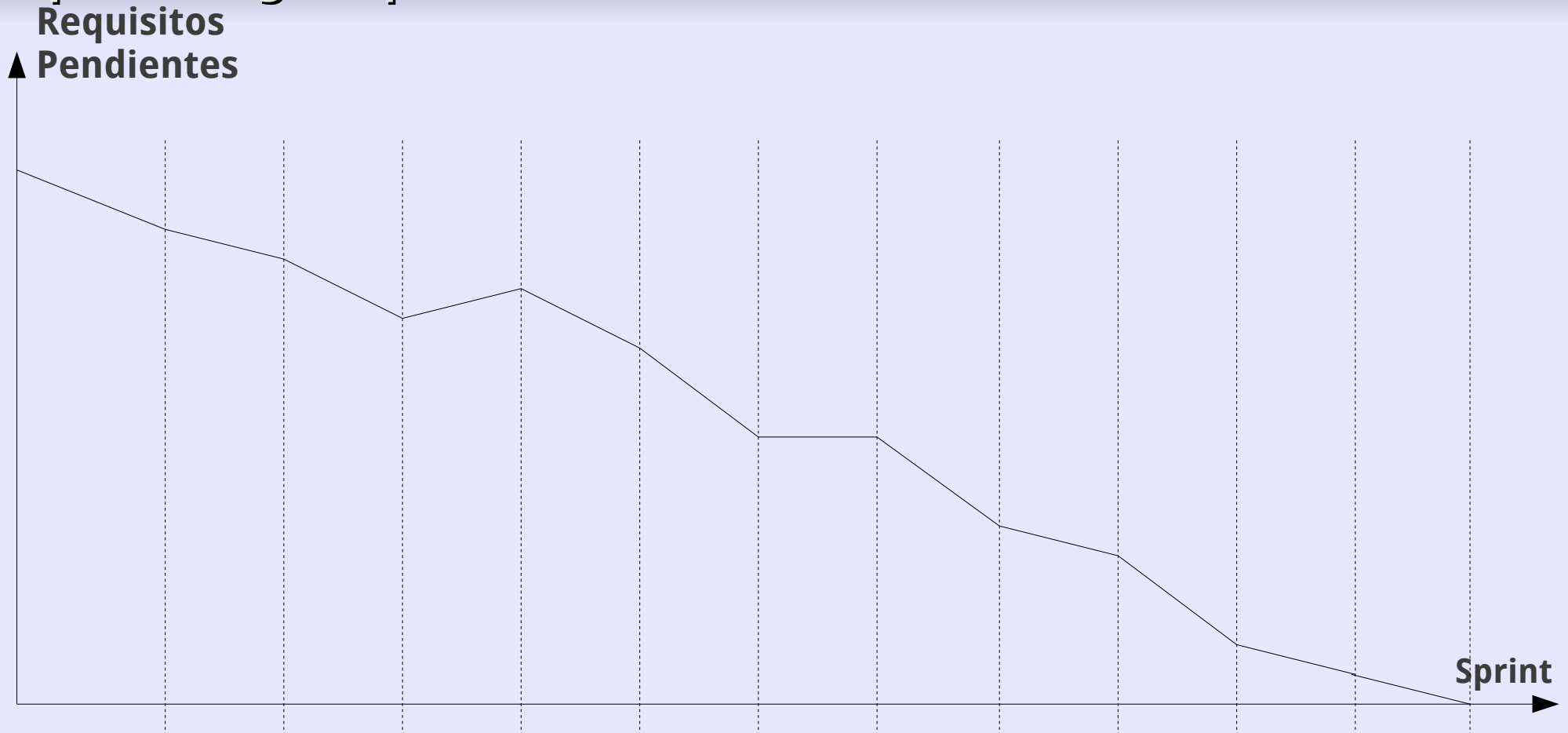
SCRUM – Documentos: *Sprint Backlog*

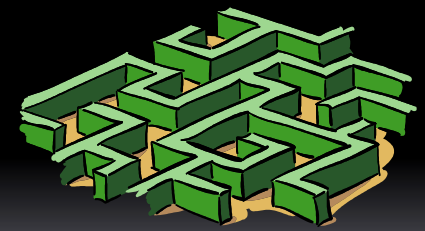
- Descripción de los requisitos del siguiente *sprint*.
- Suficiente detalle para ser implementados.
- Aceptado y comprometido por el equipo.
- Tareas valoradas en horas. Ninguna tarea ha de tener una duración superior a las 16 horas (2 jornadas).
- Tareas no están asignadas a miembros concretos.
- . . .





SCRUM - Documentos por ejemplo: *Burn Down Chart*

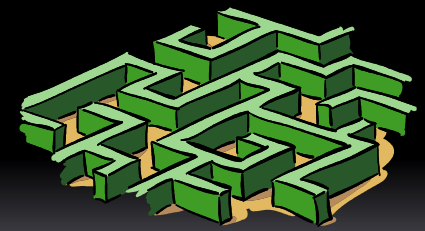




SCRUM y DRUPAL: ¿una pareja perfecta?

- Pareja perfecta para proyectos WEB . . .
- La altísima modularidad de DRUPAL facilita la construcción rápida de prototipos... (!) ¿Quién *conoce* todos los módulos ;-)?
- DRUPAL proporciona un alto rendimiento, tanto en la arquitectura interna del sitio (módulos) como en el aspecto estético (plantillas).
- DRUPAL permite respetar la regla básica de los métodos ágiles: SIMPLICIDAD.
- También ofrece una gran flexibilidad: siempre se puede adaptar un módulo existente o desarrollar uno nuevo.
- (!) El **Scrum Master** ha de *dominar* DRUPAL y todos conocemos la curva de aprendizaje de Drupal ;-)

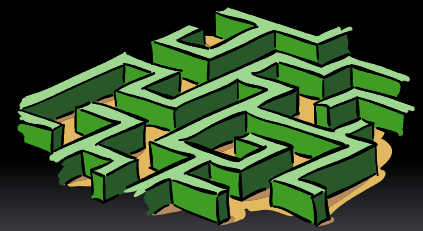




Fuentes de información sobre SCRUM

- <http://en.wikipedia.org/wiki/SCRUM>
- “**Scrum y XP desde las trincheras**”. Henrik Kniberg. Traducción al castellano en InfoQ.com y en proyectalis.com.
- “**Flexibilidad con scrum**”. Juan Palcios. En múltiples sitios de Internet.



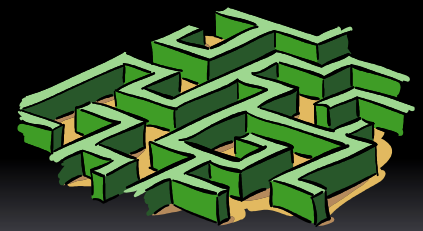


Anexo: Las 10 causas principales del fracaso de los proyectos:

- La escasa participación de los usuarios.
- Los requerimientos y especificaciones incompletas.
- Los cambios frecuentes en los requerimientos y las especificaciones.
- Falta de soporte ejecutivo.
- Incompetencia tecnológica.
- Falta de recursos operativos.
- Expectativas no realistas por todas las partes.
- Objetivos poco claros.
- Cronogramas irreales.
- Nuevas tecnologías poco conocidas o validadas.

Origen: <http://www.standishgroup.com>





Gracias por vuestra atención.



Preguntas y turno abierto . . .

