

Scrum y ecosistema software en I2E



Jose Luis Ramirez Terry
Emilio Bravo Garcia

Agenda

Quienes somos

¿Como trabajamos?

¿Qué utilizamos?

Preguntas

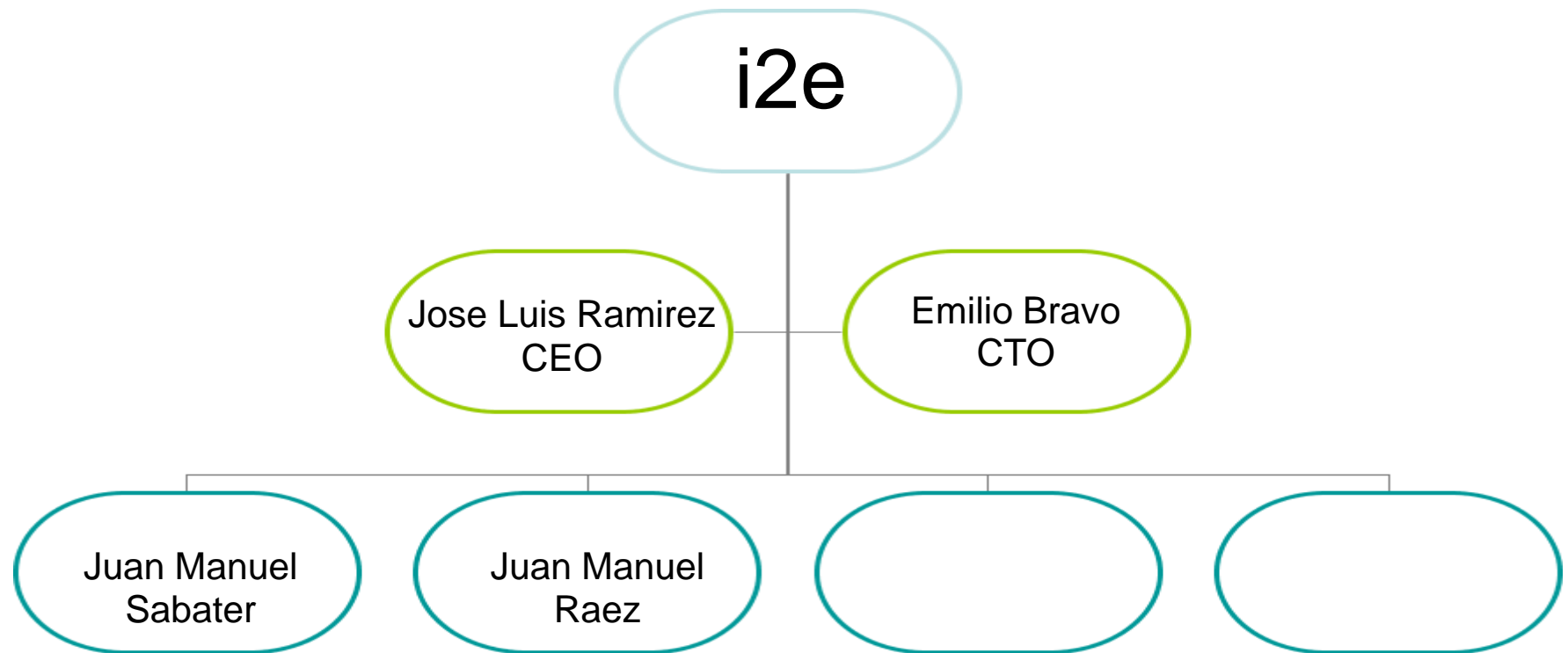
Quienes Somos

»» Empresa desarrollo de Software

Quienes Somos

- ▶ I2e es una empresa que desarrolla software a medida, gracias a la experiencia acumulada estamos trabajando en 2 productos que esperamos ofrecer en breve.
- ▶ Fundada el año 2007 (desde 2005 como autónomos) por nosotros 2
- ▶ José Luís Ramírez
- ▶ Emilio Bravo
- ▶ Fuimos alumnos del especialista en la 3ª edición (curso 04-05)
- ▶ Actualmente trabajamos 4 personas
- ▶ Más información en <http://www.i2e.com.es/>
- ▶ Blog: <http://www.i2e.com.es/blog/>
- ▶ Twitter: http://twitter.com/i2e_es

Estructura de i2e



También Buscamos la colaboración con otras empresas y profesionales

Que tecnologías utilizamos

- ▶ Sobre todo utilizamos las siguientes tecnologías
 - ▶ GWT • realización de RIA (Rich Internet Applications) utilizando Java, GWT compila el código Java a Javascript.
 - ▶ Eclipse RCP • para aplicaciones de escritorio.
 - ▶ Spring • limpieza de código, control de transacciones, inyección de dependencias, no obliga a la utilización de un servidor de aplicaciones completo.
 - ▶ Hibernate • implementación de JPA más extendida, actualmente es el framework ORM más completo.
 - ▶ Lucene • indexación de grandes volúmenes de texto (la mayoría de la información ahí fuera no esta estructurada).
 - ▶ Alguna tecnología para dispositivos móviles, Java ME?, Android?.

¿Como Trabajamos?

»» Utilizamos Scrum + AUP

Scrum

- ▶ Se trata de una metodología ágil para la Gestión de Proyectos, centrada en entregar en cada momento la funcionalidad con más valor para el cliente.
- ▶ El desarrollo se realiza de forma iterativa e incremental (en Sprints).
 - ▶ Al final de cada iteración/sprint se entrega una versión funcional del software a partir de la cual el usuario y el equipo obtienen el feedback necesario.
- ▶ <http://es.wikipedia.org/wiki/Scrum>



Roles Scrum



▶ **Product Owner**

- ▶ El Product Owner representa la voz del cliente. Se asegura de que el equipo Scrum trabaja de forma adecuada desde la perspectiva del negocio. El Product Owner escribe historias de usuario, las prioriza, y las coloca en el Product Backlog.

▶ **ScrumMaster (o Facilitador)**

- ▶ El Scrum es facilitado por un ScrumMaster, cuyo trabajo primario es eliminar los obstáculos que impiden que el equipo alcance el objetivo del sprint. El ScrumMaster se asegura de que el proceso Scrum se utiliza como es debido. El ScrumMaster es el que hace que las reglas se cumplan.

▶ **Equipo**

- ▶ El equipo tiene la responsabilidad de entregar el producto. Un pequeño equipo de 5 a 9 personas con las habilidades transversales necesarias para realizar el trabajo (diseñador, desarrollador, etc).

Roles Scrum

▶ Usuarios

- ▶ Es el destinatario final del producto.

▶ Stakeholders (Clientes, Proveedores, Inversores)

- ▶ Se refiere a la gente que hace posible el proyecto y para quienes el proyecto producirá el beneficio acordado que lo justifica. Sólo participan directamente durante las revisiones del sprint.

▶ Managers

- ▶ Es la gente que establece el ambiente para el desarrollo del producto.

Reuniones Scrum

▶ Daily Scrum

- ▶ Cada día de un sprint, se realiza la reunión sobre el estado de un proyecto. La reunión comienza puntualmente a su hora.

▶ Reunión de Planificación del Sprint (Sprint Planning Meeting)

- ▶ Al inicio del ciclo Sprint (cada 15 o 30 días), una "Reunión de Planificación del Sprint" se lleva a cabo.
- ▶ Seleccionar qué trabajo se hará.
- ▶ Preparar, con el equipo completo, el Sprint Backlog que detalla el tiempo que tomará hacer el trabajo.
- ▶ Identificar y comunicar cuánto del trabajo es probable que se realice durante el actual Sprint.
- ▶ Ocho horas como límite.



Reuniones Scrum

- ▶ **Reunión de Revisión del Sprint (Sprint Review Meeting)**
 - ▶ Revisar el trabajo que fue completado y no completado.
 - ▶ Presentar el trabajo completado a los interesados (alias “demo”).
 - ▶ El trabajo incompleto no puede ser demostrado.
 - ▶ Cuatro horas como límite.
- ▶ **Retrospectiva del Sprint (Sprint Retrospective)**
 - ▶ Después de cada sprint, se lleva a cabo una retrospectiva del sprint, en la cual todos los miembros del equipo dejan sus impresiones sobre el sprint recién superado. El propósito de la retrospectiva es realizar una mejora continua del proceso. Esta reunión tiene un tiempo fijo de cuatro horas.

Documentos Scrum

▶ **Product backlog**

- ▶ Contiene descripciones genéricas de todos los requerimientos, funcionalidades deseables, etc. priorizadas según su retorno sobre la inversión (ROI) . Es el que va a ser construido. Esta estimación ayuda al product owner a ajustar la línea temporal y, de manera limitada, la prioridad de las diferentes tareas.

▶ **Sprint backlog**

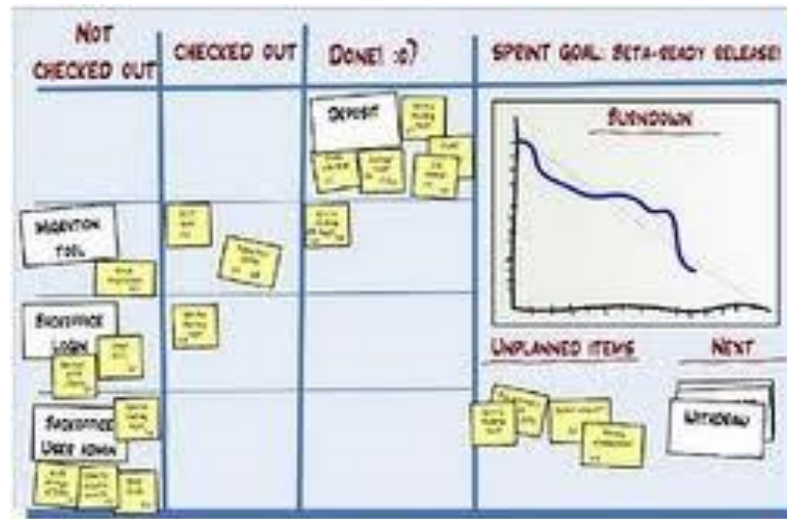
- ▶ El sprint backlog es un documento detallado donde se describe el cómo el equipo va a implementar los requisitos durante el siguiente sprint. Las tareas se dividen en horas con ninguna tarea de duración superior a 16 horas.



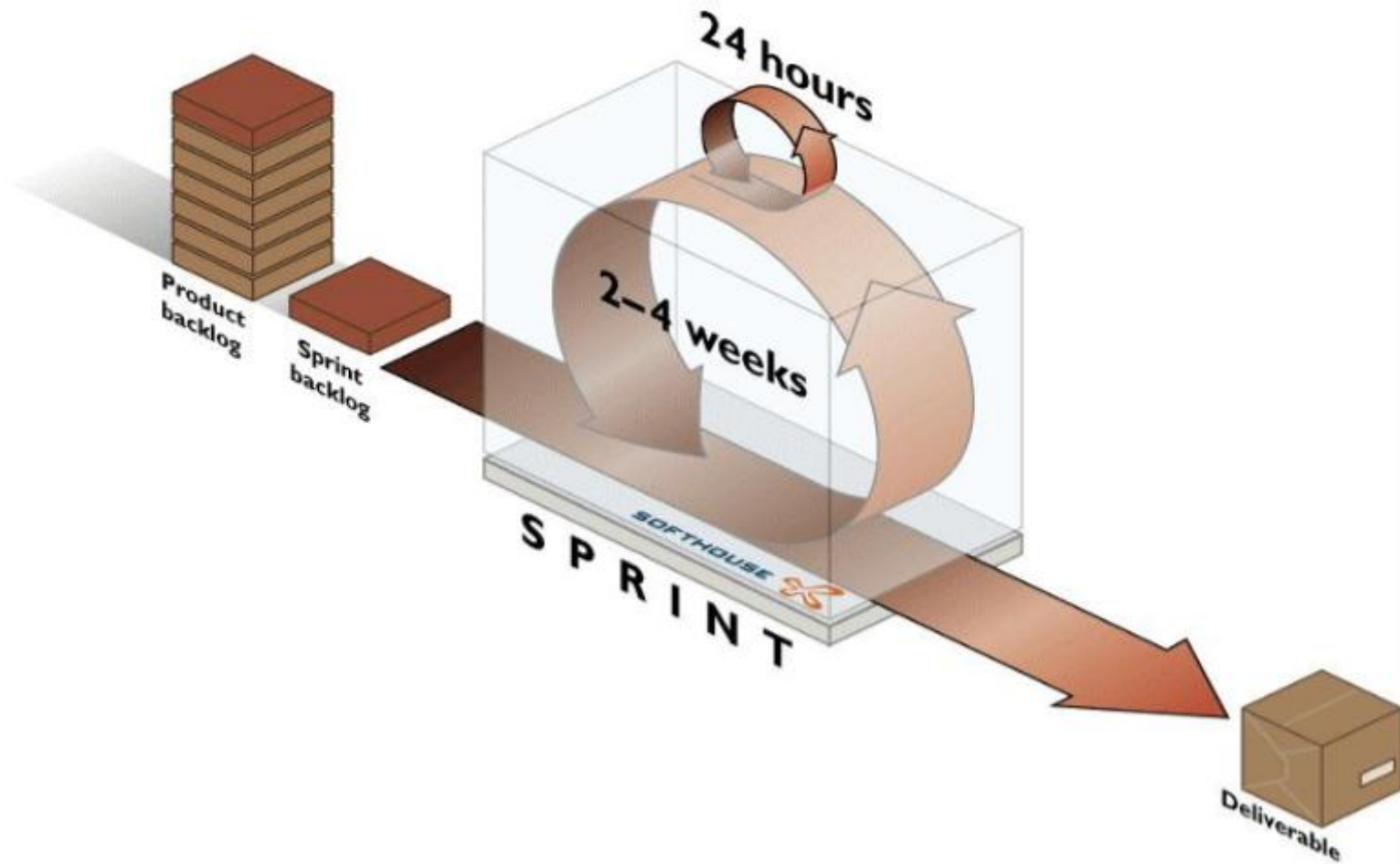
Documentos Scrum

► Burn down

- La burn down chart es una gráfica mostrada públicamente que mide la cantidad de requisitos en el Backlog del proyecto pendientes al comienzo de cada Sprint. Dibujando una línea que conecte los puntos de todos los Sprints completados, podremos ver el progreso del proyecto.

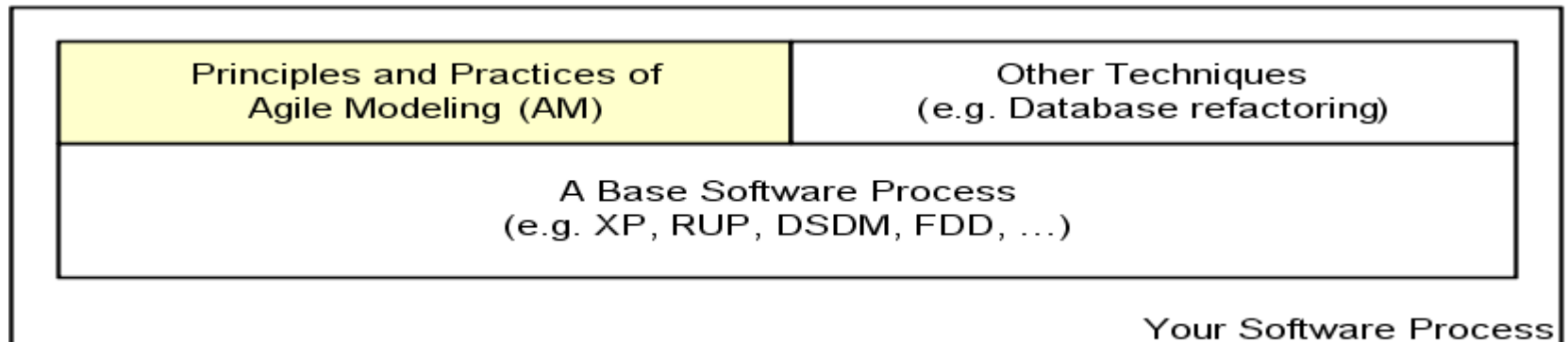


Proceso de desarrollo agil Scrum



Agile Unified Process (AUP)

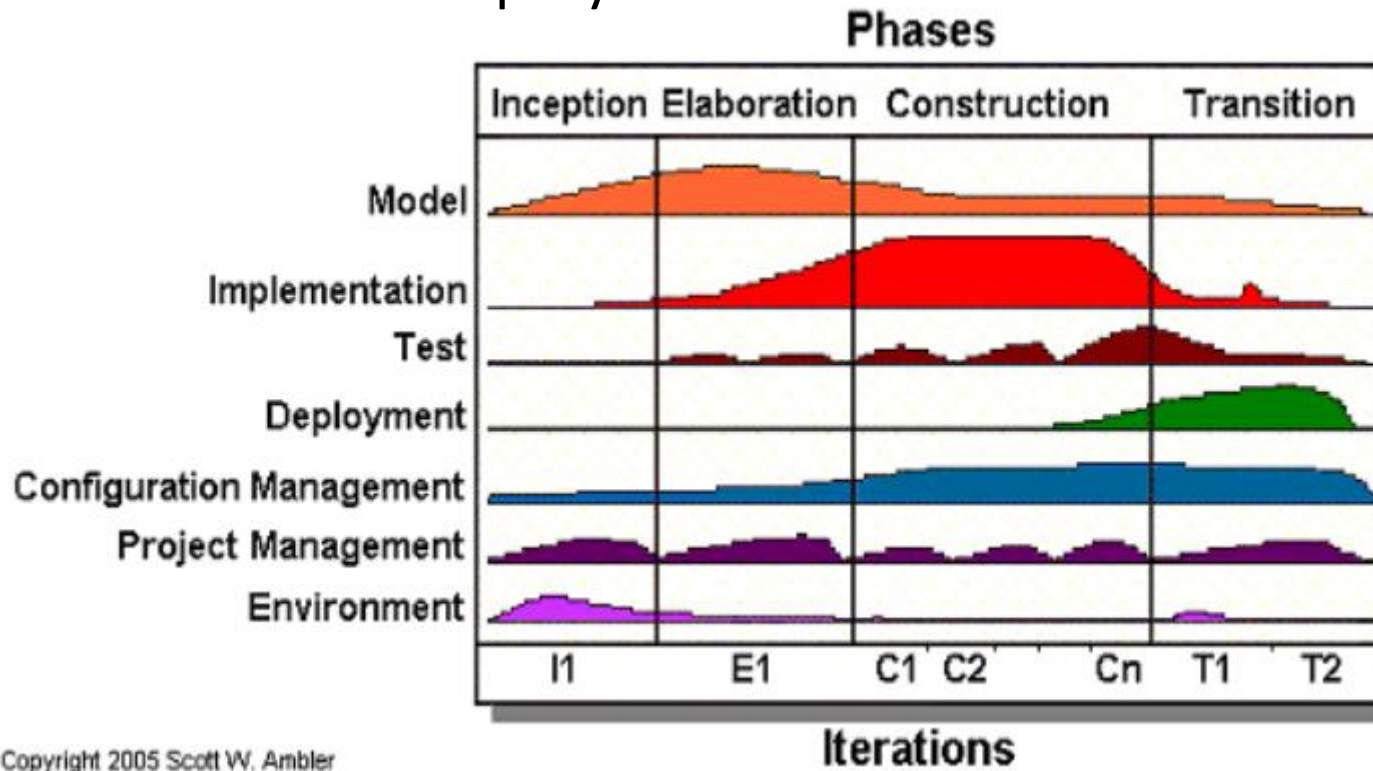
- ▶ Es una versión simplificada de RUP
- ▶ Aplica técnicas ágiles:
 - ▶ Desarrollo Dirigido por las Pruebas (TDD)
 - ▶ Modelado Ágil (AM)
 - ▶ Gestión del Cambio Ágil (ACM)
- ▶ Como proceso de desarrollo base utilizamos Scrum.



Copyright 2001-2005 Scott W. Ambler

Agile UP

- De AUP nos quedamos con las disciplinas, algunos de los roles y si fuera necesario algún artefacto para la documentación del proyecto.



Copyright 2005 Scott W. Ambler

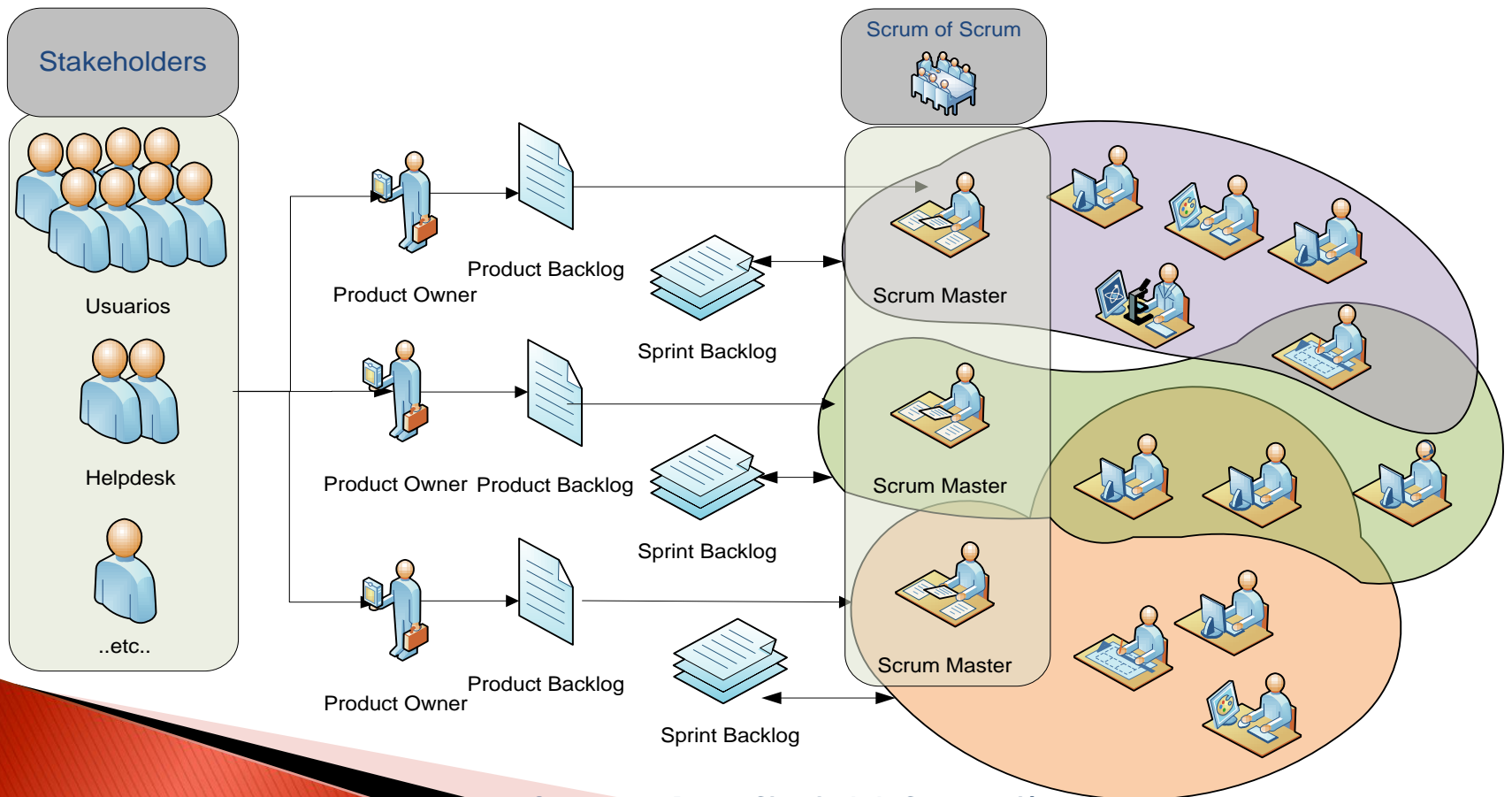
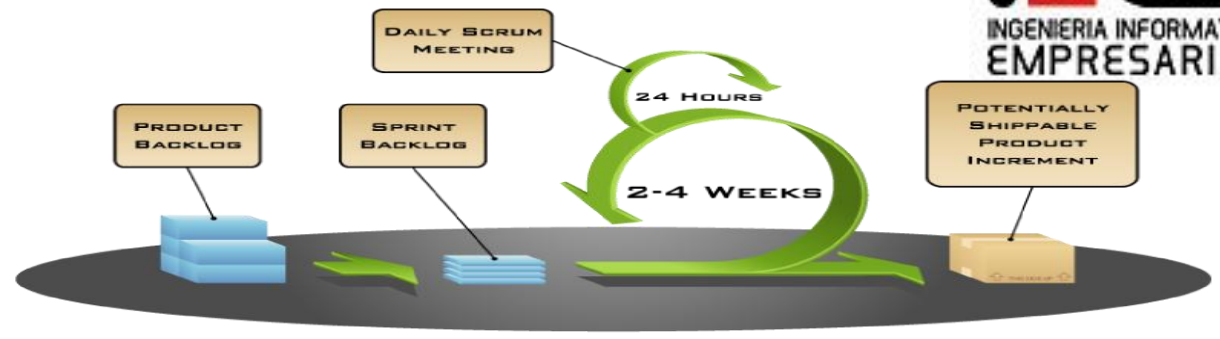
Scrum en I2E

- ▶ Utilizamos como proceso de desarrollo Scrum + AUP.
- ▶ Iteraciones de 3 semanas en cada proyecto.
- ▶ Intentamos que todos los integrantes del equipo intervienen en cada decisión del proyecto.

Ejemplo de una iteración (Sprint) en el proceso de desarrollo (Scrum + AUP)

- 1. Estimar:** se estiman las pilas de producto y extraer nuevas pilas.
- 2. Planificar:** se planifica el sprint en función del tamaño y la velocidad (recursos disponibles), se decide cuantos items del producto se puede realizar.
- 3. Modelar:** se realiza una sesión de modelado.
- 4. Desarrollar:** Desarrollo del producto según lo planificado.
- 5. Demostrar:** al final de la iteración se realiza una demo con todo lo que se ha realizado.
- 6. Realimentar:** se obtiene el feedback de la demo para realizar un nuevo sprint.

Scrum en I2E



¿Qué utilizamos?

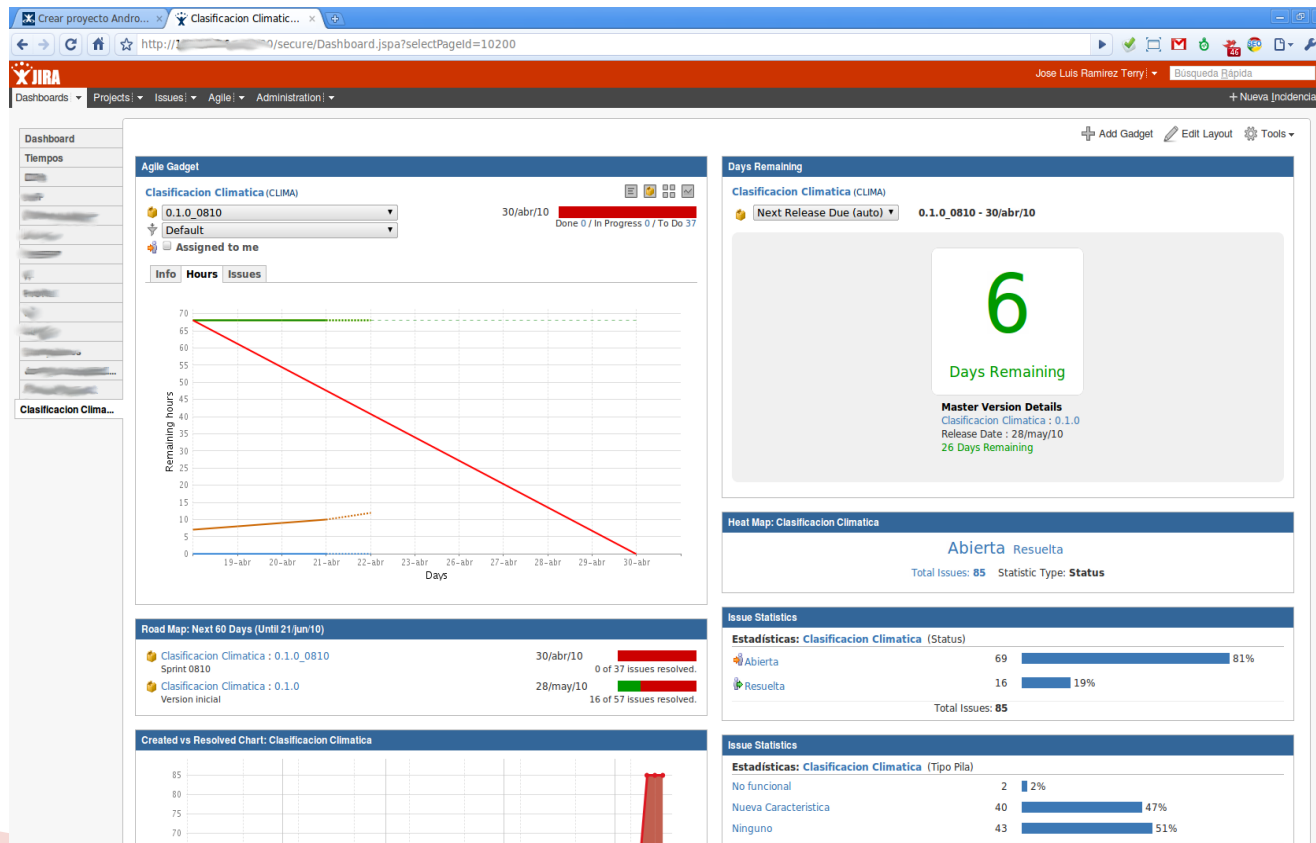
»» Ecosistema de Software

Ecosistema de software

- ▶ **MagicDraw**
 - ▶ Para la realización de los diagramas UML.
 - ▶ AndroMDA para la generación de código.
- ▶ **Eclipse + plugins**
 - ▶ Para el desarrollo de código.
- ▶ **Jira + GreenHopper y Confluence**
 - ▶ Para la gestión y documentación de los proyectos.
- ▶ **Hudson**
 - ▶ Integración Continua.
- ▶ **Subversion**
 - ▶ Control de versiones

Jira + GreenHopper

- ▶ Herramienta que facilita la gestión de proyectos siguiendo una metodología scrum



Jira + GreenHopper



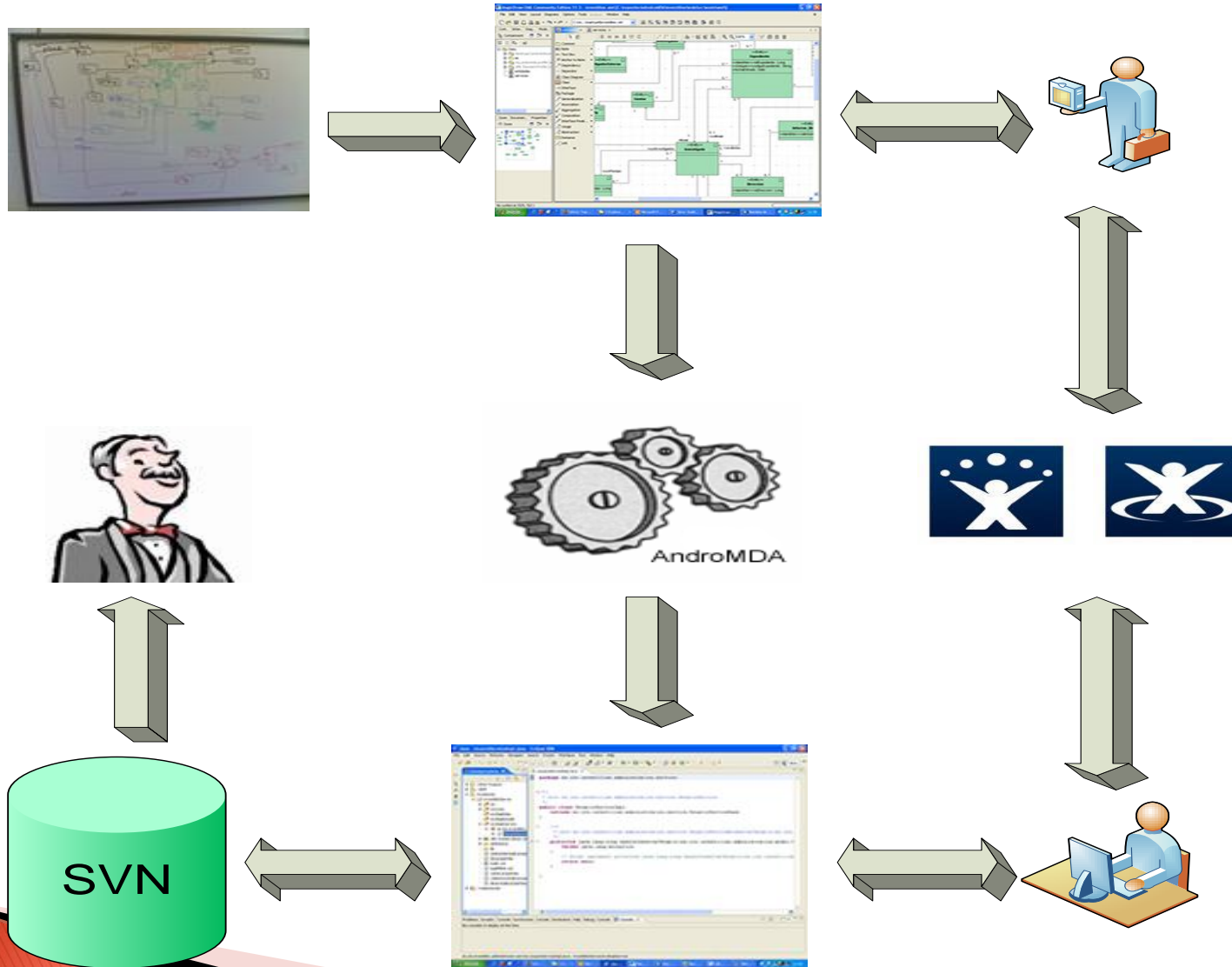
The screenshot shows the Jira GreenHopper interface for a project named "Donde Repostar". The top navigation bar includes "Dashboards", "Projects", "Issues", "Ágil", "Tempo", and "Administration". The user "Emilio Bravo Garcia" is logged in, and a search bar is visible. The main content area displays a list of issues for version "0.2.0".

ID	Summary	Version	Type	Assignee
DHR-68	Pequeños retoques esteitcos y otros	0.2.0	Web	Jose Luis Ramirez Terry
DHR-74	Desactivar las gasolineras que lleven mas de 15 dias sin actualizar.	0.2.0	ETL	Emilio Bravo Garcia
DHR-26	Descargar la informacion extra de las gasolineras.	0.2.0	Crawler	Emilio Bravo Garcia
DHR-31	RSS para obetner la informacion de las gasolineras	0.2.0	Servidor	Emilio Bravo Garcia
DHR-32	Obtener las noticias de algunos medios filtrando el contenido por palabras claves	0.2.0	Servidor	Emilio Bravo Garcia
DHR-33	Consultar en tiempo real las incidencias de la DGT	0.2.0	Crawler	Emilio Bravo Garcia
DHR-34	Crawler para descargar los coches	0.2.0	Crawler	Emilio Bravo Garcia
DHR-37	busqueda de coches	0.2.0	Desconocida	Emilio Bravo Garcia
DHR-38	Busqueda de radares	0.2.0	Desconocida	Emilio Bravo Garcia
DHR-41	Los usuarios pueden actualizar los datos de las gasolineras.	0.2.0	Servidor	Emilio Bravo Garcia

Summary statistics for version 0.2.0:

- Padre: Ninguno
- Fecha de inicio: Ninguna
- Fecha de Fin: Ninguna
- Fecha de Publicación: Ninguna
- Bug: 10
- Pila Características: 9
- Pila Producto: 21
- Tarea Bug: 1
- Tarea Pila: 7
- No. Total de Incidencias: 48
- Por Hacer: 38
- En Progreso: 2
- Hecho: 8
- Sin Resolver: 40
- Resuelto: 8
- Estimación del Tiempo: 1 día, 2 horas, 50

Proceso de desarrollo en el ecosistema



Herramientas Complementarias

- ▶ **Cobertura de los Test**
 - ▶ Utilizamos la herramienta Cobertura (<http://cobertura.sourceforge.net/>) para obtener el % de código cubierto por los test automáticos.
 - ▶ Con los informes obtenidos podemos observar que partes del código están sin probar.
- ▶ **Análisis Estático del Código**
 - ▶ Utilizamos Herramientas como FindBugs (<http://findbugs.sourceforge.net/>) o PDM para realizar un análisis estático del código y encontrar errores de implementación que reducen la calidad del código.

Integraciones Continuas

- ▶ Con la integración continua automatizamos:
 - ▶ la compilación de la aplicación
 - ▶ los test
 - ▶ el análisis estático del código
 - ▶ la generación del desplegable.
- ▶ También controlamos la evolución de la aplicación de una forma automatizada.
- ▶ Utilizamos Hudson (<https://hudson.dev.java.net/>) como servidor de integración. Lo utilizamos por su versatilidad y la integración con Jira (<http://www.atlassian.com/software/jira/>).

Ecosistema

Las Herramientas del ecosistema nos ayudan a controlar el desarrollo y a ser mas productivos. Un buen ecosistema no tiene precio.

The screenshot displays a multi-windowed development environment. On the left, an IDE (Eclipse) shows Java code for a service class. In the center, the Hudson web interface provides a dashboard for 'Project Libreria J2EE Tipo freeStyle', including a 'Build History' table and a 'File Coverage Summary' table. On the right, a series of trend charts (Test Result, FindBugs, PMD, Code Coverage) show performance metrics over time. At the bottom right, a table summarizes coverage data for various files.

File	Classes	Methods	Lines	Conditionals
libreria.server.service	67% (2/3)	67% (2/3)	44% (8/24)	18% (11/61)
libreria.server.modelo	100% (4/4)	57% (4/7)	49% (21/43)	32% (36/111)
libreria.utilidades	0% (0/2)	0% (0/2)	0% (0/24)	0% (0/31)
libreria.servicio	100% (1/1)	100% (1/1)	100% (1/1)	75% (3/4)
libreria.servicio	0% (0/0)	0% (0/0)	0% (0/76)	0% (0/22)
libreria.server.modelo	0% (0/1)	0% (0/1)	0% (0/8)	0% (0/9)

Para Saber Más

▶ Bibliografía

- ▶ Scrum y XP desde las trincheras.
- ▶ Flexibilidad con Scrum.
- ▶ Continuous Integration Improving Software Quality and Reducing Risk.
- ▶ Beyond Software Architecture: Creating and Sustaining Winning Solutions.

▶ Enlaces

- ▶ <http://www.navegapolis.net/>
- ▶ <http://www.presionblogosferica.com/>
- ▶ <http://www.mountangoatsoftware.com/>
- ▶ <http://www.andromda.org/>

¿Preguntas?



Gracias por su atención

- ▶ <http://www.i2e.es>
- ▶ Twitter: i2e_es
- ▶ joseluis@i2e.es
- ▶ emilio@i2e.es
- ▶ i2e@i2e.es

