

IBM Software Group

Plataforma de Desarrollo de Software de IBM

IBM SDP

Rational software



ON DEMAND BUSINESS™

Agenda

- Herramientas para el ciclo completo de vida del software
- Eclipse 3.0: Framework abierto y multiplataforma
- Mejores prácticas para el desarrollo de software
- Desarrollo y arquitectura basada en modelado. UML 2.0
- Desarrollo rápido de aplicaciones Java. J2EE y sus nuevas tecnologías
- Gestión de la configuración: control del cambio y de versiones
- Aseguramiento de la calidad del software

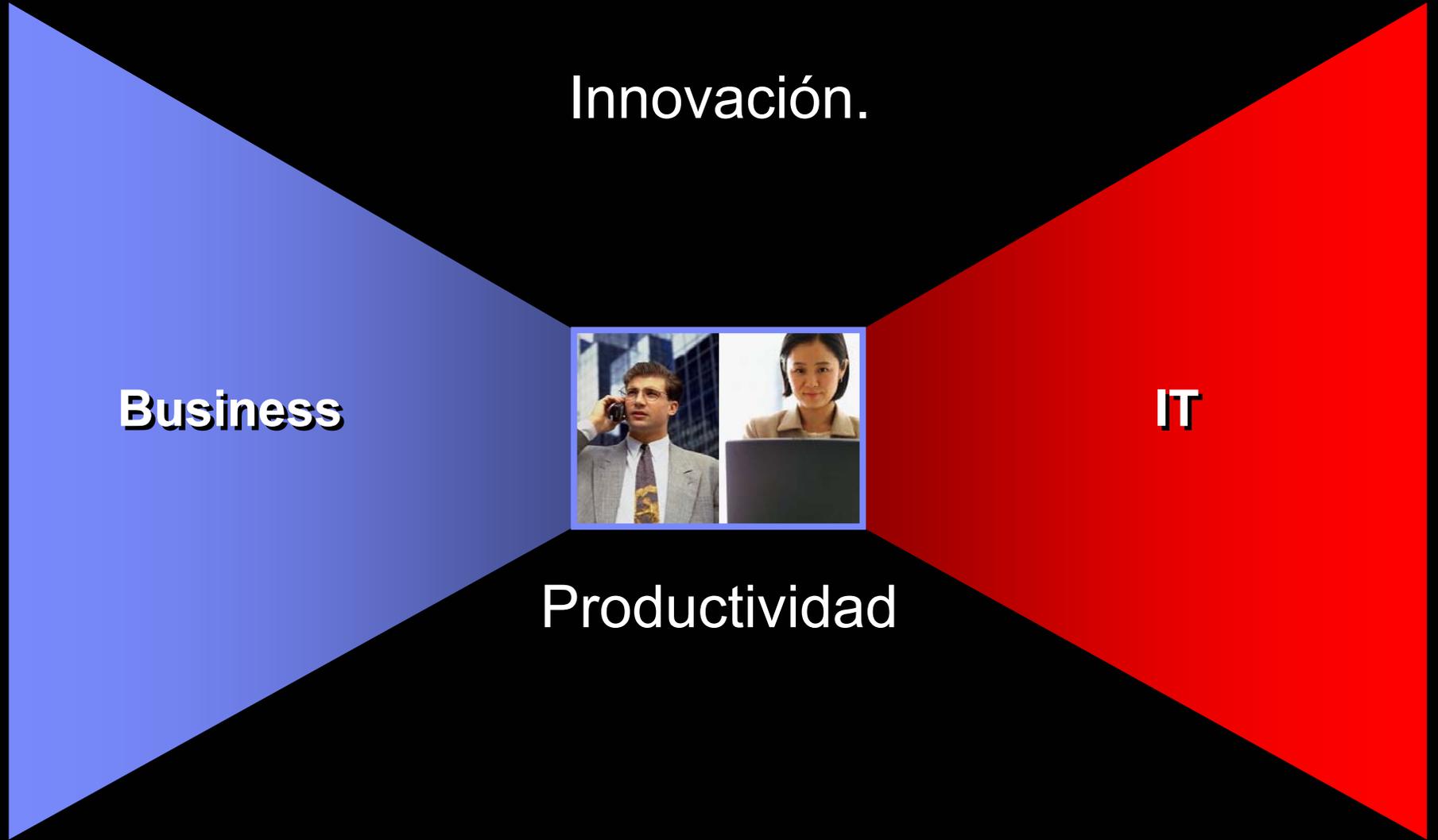


Agenda

- **Herramientas para el ciclo completo de vida del software**
- Eclipse 3.0: Framework abierto y multiplataforma
- Mejores prácticas para el desarrollo de software
- Desarrollo y arquitectura basada en modelado. UML 2.0
- Desarrollo rápido de aplicaciones Java. J2EE y sus nuevas tecnologías
- Gestión de la configuración: control del cambio y de versiones
- Aseguramiento de la calidad del software



Imperativos en el mundo actual



Ámbito del desarrollo de Software

Modernizar, Extender, Construir, Integrar, Desplegar

Business



IT

Modernizar

Existing Systems



Deploy

Construir

New Systems



Packaged Apps

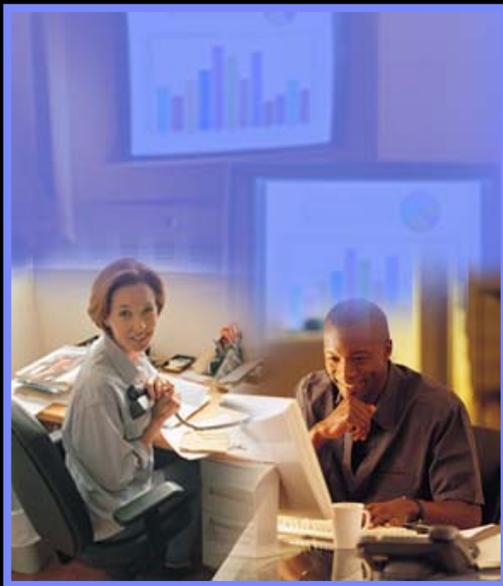
Extender

Arquitecturas basadas en componentes y servicios



El desarrollo de software clave en el éxito del negocio. Permite tener conocimiento de clientes, mercados y procesos.

Clientes



- Preferencias
- Historia
- Información demográfica

Mercado



- Competidores
- Proveedores
- Tendencias de Mercado

Procesos de Negocio

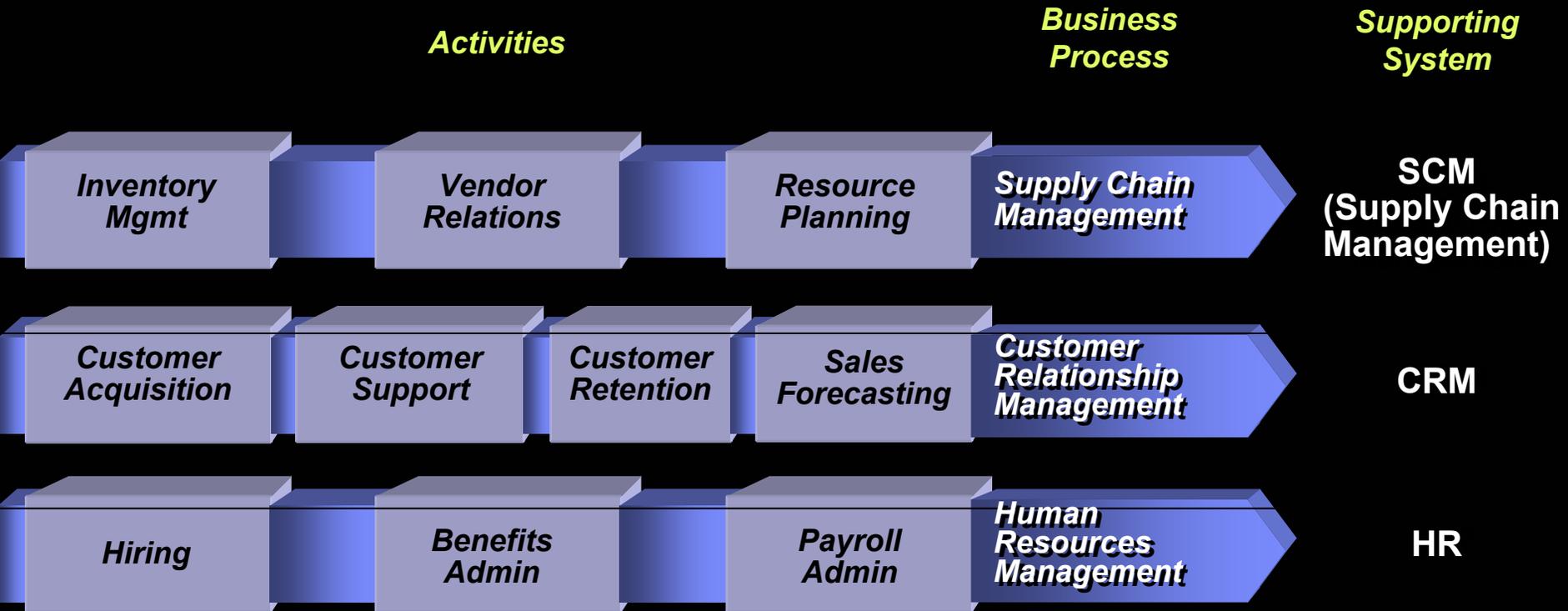


- Métodos y procedimientos
- Enlaces



Desarrollo de Software como proceso de negocio.

Procesos de Negocio: "Un conjunto de actividades cuya agrupación crea valor al cliente."



Desarrollo de Software como proceso de negocio.

Activities

Business Process

Supporting System

Requirements & Analysis

Design & Construction

Testing

Project & Change Mgmt

Software Development

SDP

Inventory Mgmt

Vendor Relations

Resource Planning

Supply Chain Management

**SCM
(Supply Chain Management)**

Customer Acquisition

Customer Support

Customer Retention

Sales Forecasting

Customer Relationship Management

CRM

Hiring

Benefits Admin

Payroll Admin

Human Resources Management

HR



The IBM Software Development Platform



Analista

Modelado y simulación de procesos de negocio



Arquitecto

Modelado de aplicaciones y de datos



Desarrollador

Construcción y generación de código



Probador

Diseño y ejecución de pruebas



The IBM Software Development Platform



Analista



Arquitecto



Desarrollador



Probador

Modelado y simulación de procesos de negocio

Modelado de aplicaciones y de datos

Construcción y generación de código

Diseño y ejecución de pruebas



Project Manager

- Proceso Común.
- Seguimiento de proyecto.
- Gestión de requisitos
- Gestión de Configuración.
- Control de versión.
- Verificación de la calidad.

TEAM UNIFYING PLATFORM



Agenda

- Herramientas para el ciclo completo de vida del software
- **Eclipse 3.0: Framework abierto y multiplataforma**
- Mejores prácticas para el desarrollo de software
- Desarrollo y arquitectura basada en modelado. UML 2.0
- Desarrollo rápido de aplicaciones Java. J2EE y sus nuevas tecnologías
- Gestión de la configuración: control del cambio y de versiones
- Aseguramiento de la calidad del software



Necesidades actuales de un IDE: ¡pasado, presente y futuro!

Web

HTML,
DHTML,
XML, JSP
Wireless

Java

J2EE, J2SE, J2ME
Conectividad JCA

Documentación

Diagramas, UML, mantenimiento
automático, código, datos, arquitectura...

Servicios Web

Creación (build+wrap),
Consumo, Composición y
coreografías visuales



Host

Desarrollo en Cobol
y PL/1

Desarrollo para dispositivos

Telemática,
Dispositivos
domésticos, J2ME...

Desarrollo rápido

Desarrollo visual con asistentes,
reutilización componentes, código 4GL
y plantillas de aplicaciones

Componentización en activos (assets)

Análisis de impacto



Eclipse: Plataforma universal para herramientas de desarrollo

- Arquitectura abierta basada en módulos
- Multiplataforma, multilenguaje, módulos abiertos y comerciales
- Los módulos básicos y gratuitos permiten compilar Java o C++
- Soportado por los principales fabricantes

Entorno desarrollo plugins

PDE

Compilador y herramientas Java

JDT

Plataforma eclipse

Platform

Máquina virtual Java2

Java VM



Gran acogida entre fabricantes y desarrolladores

- Entusiasmo entre los desarrolladores (datos año 2003)
 - ▶ 50.000 descargas en el primer fin de semana (eclipse 2.1)
 - ▶ 18 millones de descargas hasta la fecha (eclipse 2.1)
 - ▶ Número de fabricantes y categorías de plugins creciendo rápidamente
 - ▶ 644 plugins disponibles (sept. 2004)

- Proyecto activo
 - ▶ Versión 3.0 aparecida en 2004
 - ▶ Múltiples subproyectos iniciados

- Fuerte relación con la comunidad universitaria
 - ▶ Eclipse Innovation Grants: Premios a la innovación universitaria sobre eclipse (50 en 2002, 55 en 2003 y 78 en 2004).
 - ▶ Uno de estos fue ganado por un investigador de Univesidad de A Coruña con un plugin para facilitar el desarrollo de personas con discapacidades visuales



Existen plugins para todos los ámbitos

EclipsePlugins : plugin overview - Mozilla

File Edit View Go Bookmarks Tools Window Help

Back Forward Reload Stop

Home Bookmarks mozilla.org moz

eclipse plugins

- home
- what's new
- categories
- search
- promotion
- progr.
- contact
- links
- login / register

registered users: 4801
active visitors: 794
pages per minute: 91

RSS

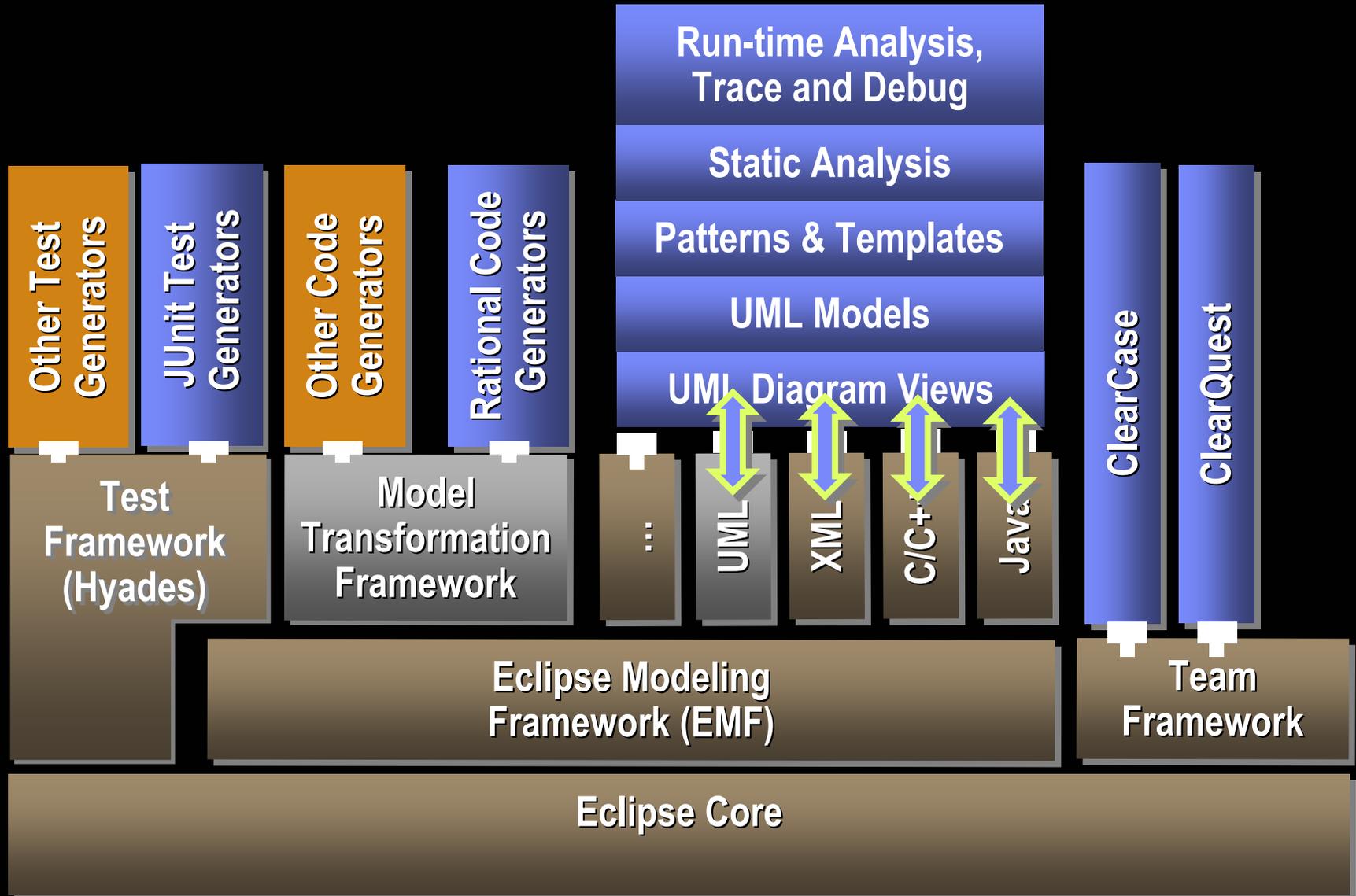
hosting provided by

Plugin categories

- All (644)
- Bug Tracker (5)
- Code Generation (9)
- Code mngt (71)
- Database (37)
- Deployment (17)
- Documentation (21)
- Editor (16)
- Entertainment (33)
- Graphics (5)
- J2EE development platform (32)
- Languages (37)
- Middleware (6)
- Mobile/PDA (10)
- Modelling (13)
- Network (8)
- News (5)
- Obsolete (1)
- Patterns (4)
- Profiling (10)
- Project management (4)
- Research (3)
- Rich Client (4)
- Safety-critical (2)
- SCM (20)
- Snippets (2)
- Snippets/Modelling (1)
- Source Code Analyzer (19)
- Team (5)
- Testing (22)
- Tomcat (5)
- Tools (49)
- Tutorial (26)
- UI (29)
- UI components (7)
- UML (16)
- Utility (14)
- Web (33)
- Web Service (15)
- XML (28)

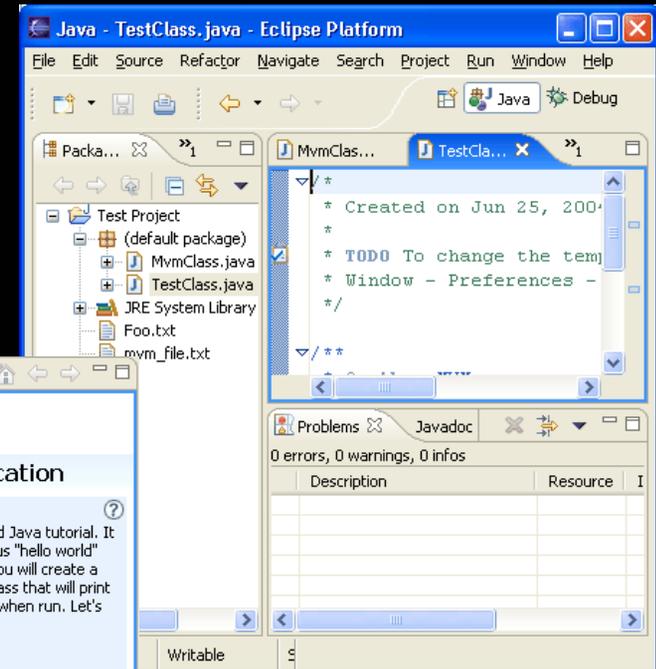
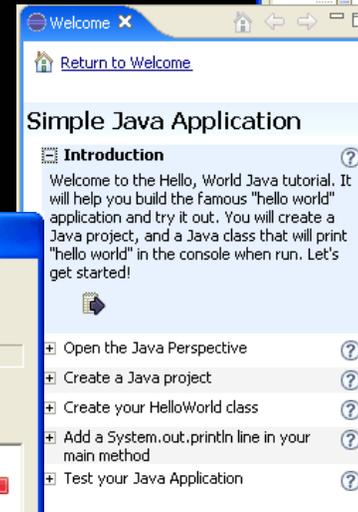
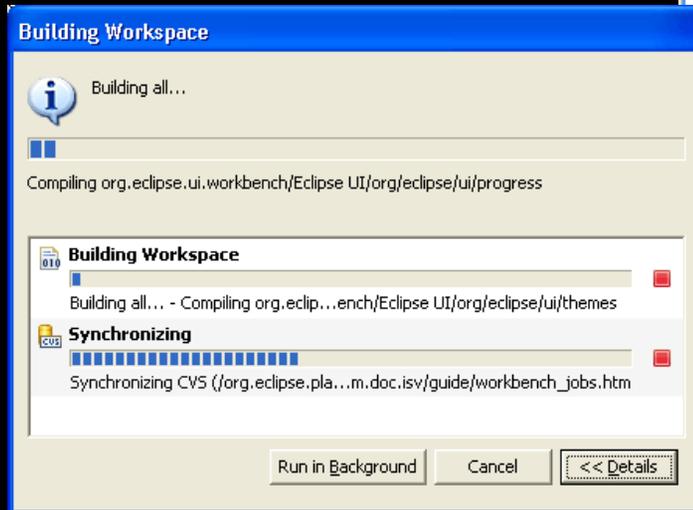


IBM SDP: Arquitectura abierta basada en Eclipse



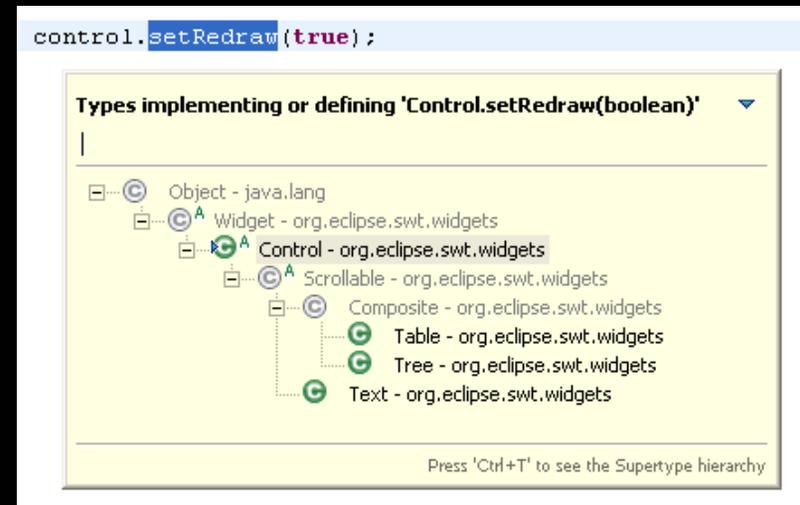
Eclipse 3.0: Mejoras I – plataforma, interfaz...

- Nuevo interfaz, más configurable y más rápido (mejor multitarea)
- Desarrollo basado en actividades
- Más control al realizar actividades
- Guías de uso
- Plataforma para Cliente Rico
- Y más...



Eclipse 3.0: Mejoras II – Equipo, Ant, Plugins y Java

- Trabajo en equipo: soporte mejorado a ClearCase y CVS
- Búsqueda y reemplazo
 - ▶ QuickDiff: Merge gráfico de ficheros
 - ▶ Localización en múltiples artefactos de desarrollo
 - ▶ Soporte a expresiones regulares
- Ayuda: creación en tiempo real de documentos de ayuda
- Actualizaciones: posibilidad que sean automáticas e invisibles
- Mejoras para Java
 - ▶ Mejor gestión importaciones (copy&paste)
 - ▶ Vista rápida de jerarquías de clases
 - ▶ Visualización rápida estructura clases
 - ▶ Más herramientas para Javadoc
 - ▶ Mejor refactorización: desde declaraciones, Refresca Javadocs, etc
 - ▶ Mejoras para pruebas con **jUnit**
 - ▶ Debug más flexible y potente, etc...



Agenda

- Herramientas para el ciclo completo de vida del software
- Eclipse 3.0: Framework abierto y multiplataforma
- **Mejores prácticas para el desarrollo de software**
- Desarrollo y arquitectura basada en modelado. UML 2.0
- Desarrollo rápido de aplicaciones Java. J2EE y sus nuevas tecnologías
- Gestión de la configuración: control del cambio y de versiones
- Aseguramiento de la calidad del software



Mejores prácticas en desarrollo de software

Los 4 imperativos.



Desarrollo Iterativo

- ✓ Proceso Flexible
- ✓ Creación de sucesivas iteraciones de releases ejecutables.
- ✓ Confirmación de requisitos.



Beneficios Negocio

- Menos proyectos fallidos, reducción de costes.
- Integración de equipos distribuidos, subcontratas, outsourcing, etc.



Beneficios IT

- Minimizar el Riesgo
- Identificar la raíz de los problemas a tiempo.
- Dimensionamiento y métricas de los proyectos realista.

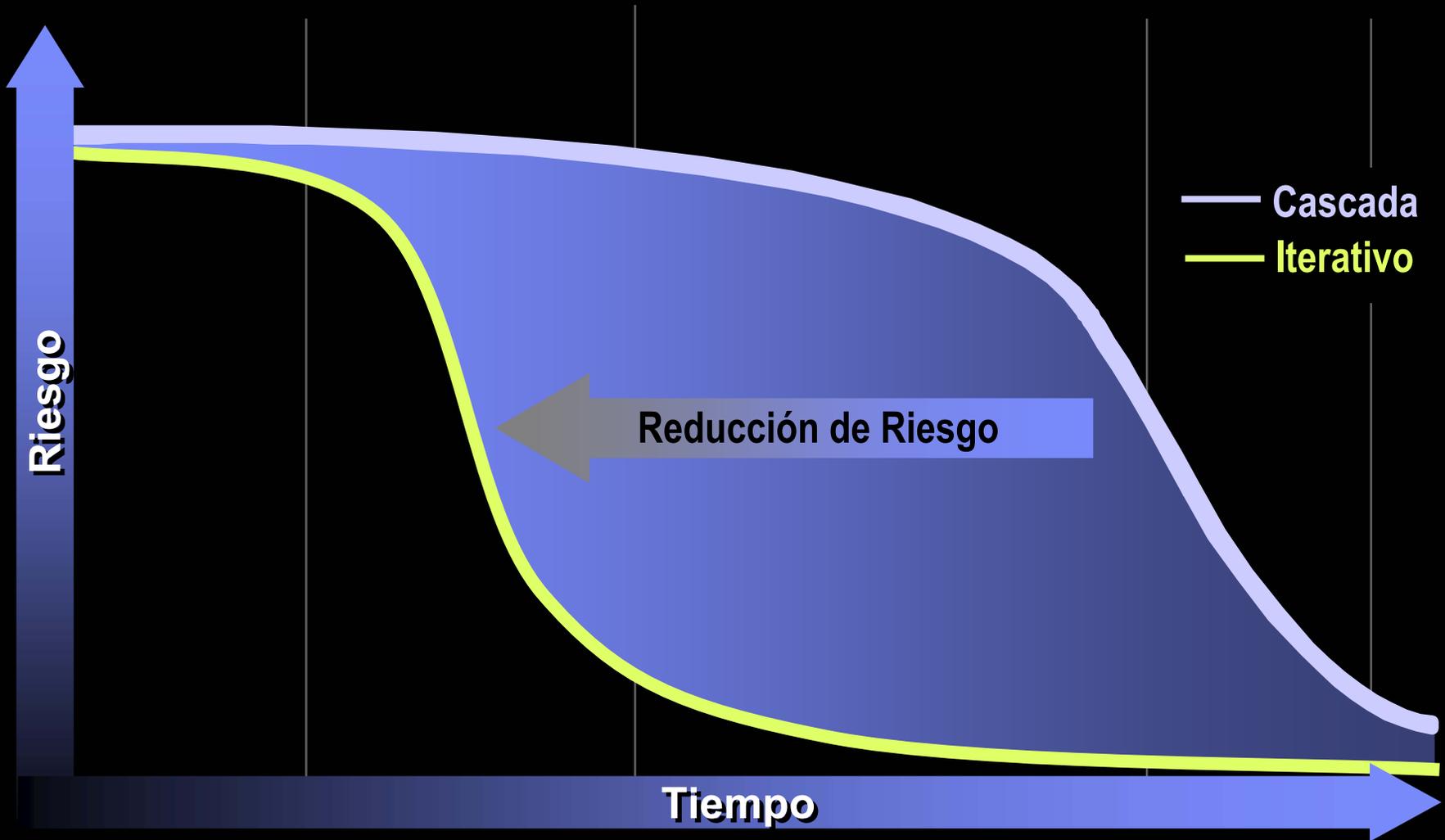


Desarrollo Iterativo:

- **Validar** la solución con los requisitos en cada iteración.
- **Congelar** gradualmente versiones de la arquitectura para facilitar la reutilización de componentes.
- Encontrar y **corregir** defectos pronto en el ciclo.
- **Entender** el impacto de cambio.
- Permite utilizar las mejores prácticas para planificar, ejecutar y medir los proyectos.

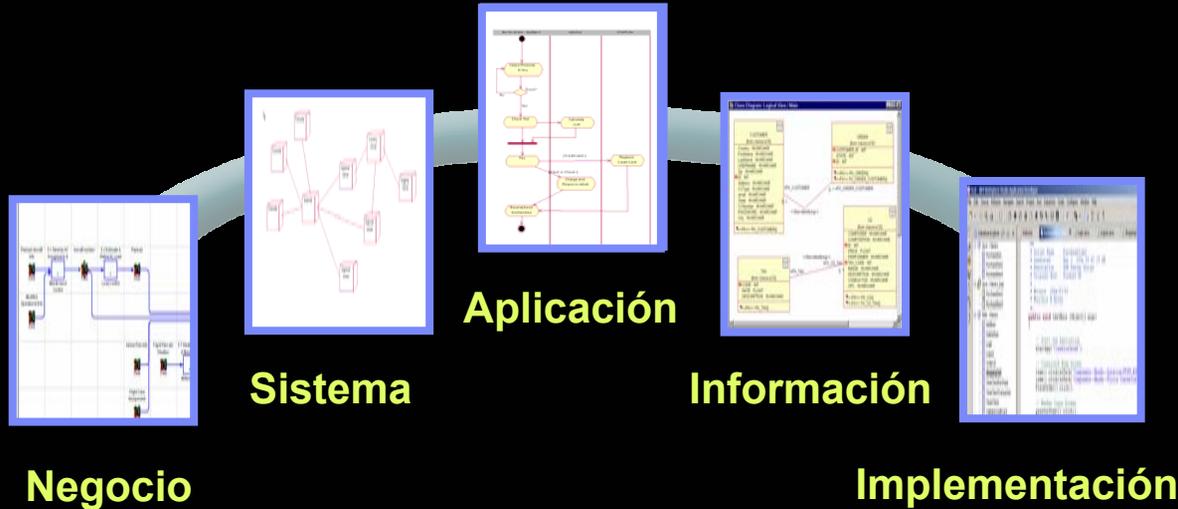


Minimizar el riesgo en las fases iniciales



Foco en la Arquitectura

- ✓ Creación de modelos a diferentes niveles de abstracción
- ✓ Implementación de arquitecturas basadas en componentes y orientadas a servicios
- ✓ Visualización de la arquitectura



Beneficios Negocio

- Mejor alineamiento de los proyectos con las necesidades del negocio
- Mayor nivel de reutilización incrementa la productividad
- Cambios más sencillos y rápidos



Beneficios IT

- Facilita una arquitectura y diseño basado en modelado
- Escala mejor que focalizándose en el código
- Facilita la trazabilidad en todo el ciclo de vida



Aseguramiento continuo de la calidad

- ✓ Construcción con calidad desde el principio
- ✓ Automatización de pruebas
- ✓ Comunicación entre análisis, desarrollo y pruebas



Beneficios Negocio

- Menos riesgo en las implantaciones
- Ciclos rápidos
- Aseguramiento de los niveles de satisfacción



Beneficios IT

- Trazabilidad completa
- Aceleración de la búsqueda y resolución de problemas
- Aceleración del ciclo de pruebas



Gestión del Cambio

- ✓ Gestión del Cambio
- ✓ Gestión de artefactos
- ✓ Automatización de flujos de trabajo

Software Configuration Management



Beneficios Negocio

- Rápida respuesta al cambio.
- Desarrollo en paralelo
- Reutilización y protección de artefactos.

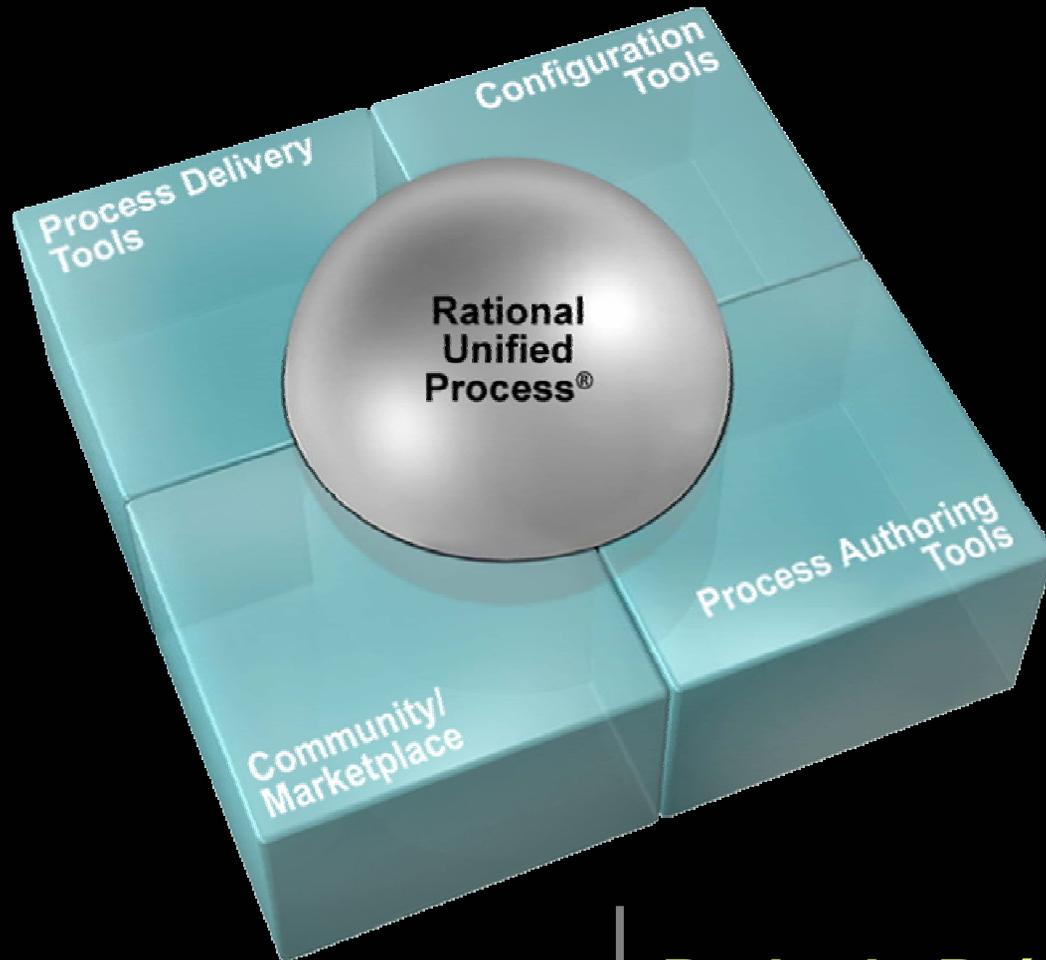


Beneficios IT

- Workflow flexibles
- Gestión de equipos distribuidos
- Trazabilidad en el ciclo de desarrollo



IBM Rational Unified Process Platform



Probado.Práctico.Flexible.



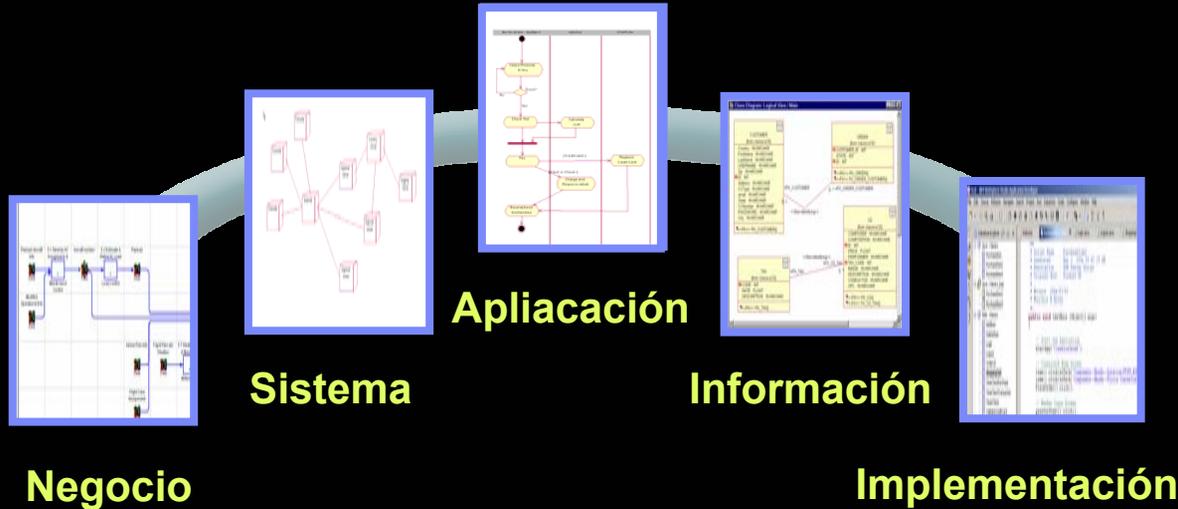
Agenda

- Herramientas para el ciclo completo de vida del software
- Eclipse 3.0: Framework abierto y multiplataforma
- Mejores prácticas para el desarrollo de software
- **Desarrollo y arquitectura basada en modelado. UML 2.0**
- Desarrollo rápido de aplicaciones Java. J2EE y sus nuevas tecnologías
- Gestión de la configuración: control del cambio y de versiones
- Aseguramiento de la calidad del software



Foco en la Arquitectura

- ✓ Creación de modelos a diferentes niveles de abstracción
- ✓ Implementación de arquitecturas basadas en componentes y orientadas a servicios
- ✓ Visualización de la arquitectura



Beneficios Negocio

- Mejor alineamiento de los proyectos con las necesidades del negocio
- Mayor nivel de reutilización incrementa la productividad
- Cambios más sencillos y rápidos



Beneficios IT

- Facilita una arquitectura y diseño basado en modelado
- Escala mejor que focalizándose en el código
- Facilita la trazabilidad en todo el ciclo de vida



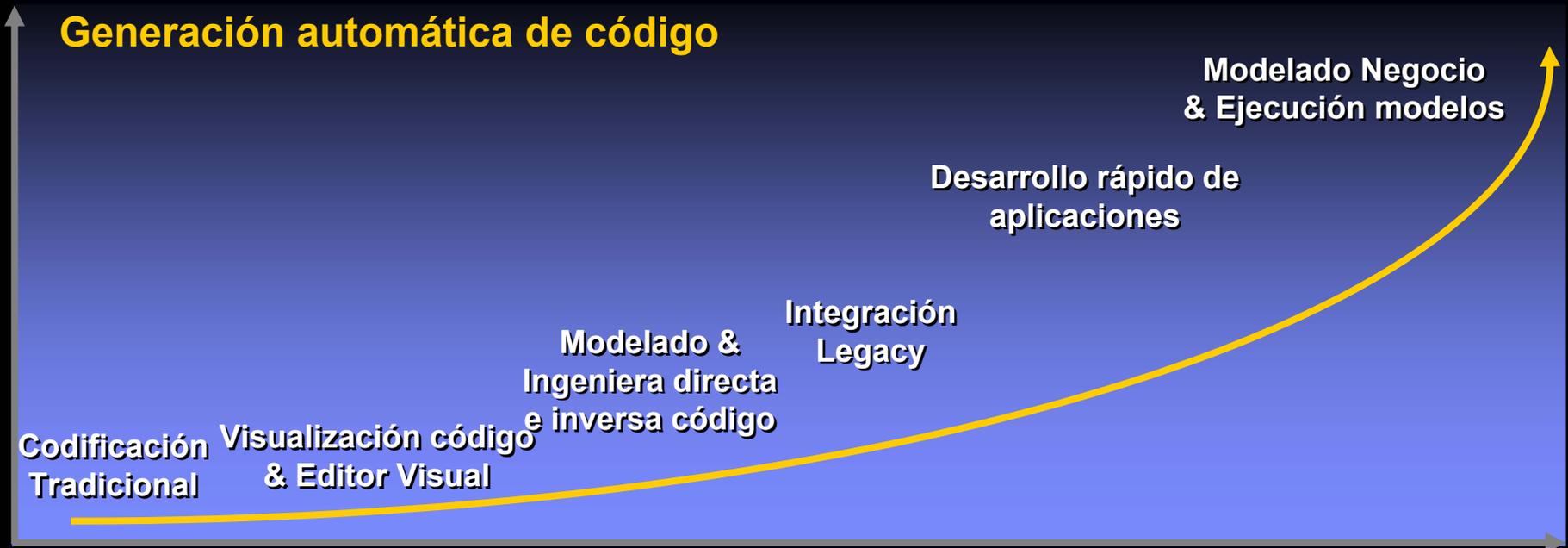
Foco en la Arquitectura: Asegurar el éxito

- ✓ Establecer un **proceso** para el desarrollo y mantenimiento de la arquitectura
- ✓ **Analizar** qué se necesita construir antes de construirlo
- ✓ Diseñar y comunicar tu arquitectura a través de **modelos**
- ✓ **Construir** sistemas que cumplen con la arquitectura definida
- ✓ **Validar** que la implementación está en concordancia con el diseño
- ✓ **Gestionar los cambios** en la arquitectura a medida que el software y los sistemas evolucionan



Espectro de Soluciones

Adopta el estilo correcto de desarrollo que mejor se adapte a tus necesidades



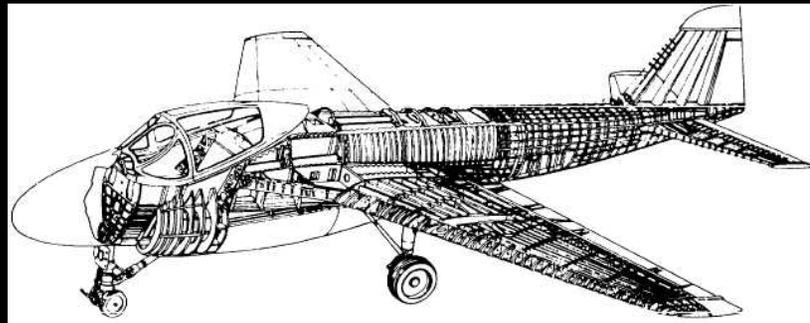
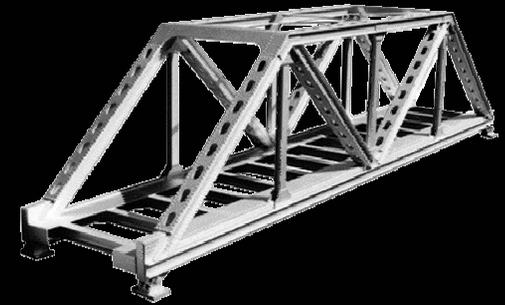
IBM tiene soluciones de implementación que se expanden a través :

- Lenguajes programación
- Sistemas operativos
- Niveles formación

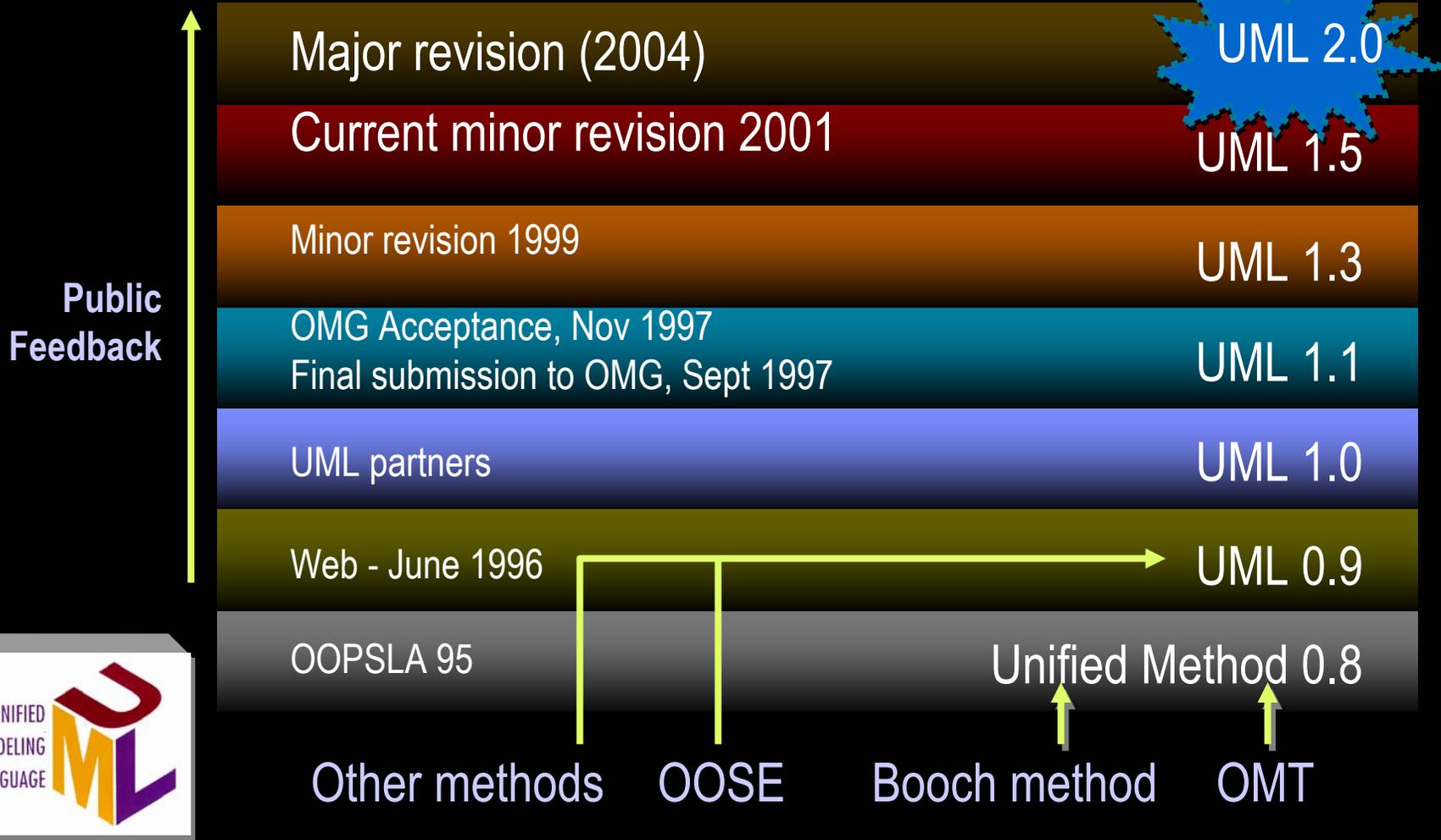


Beneficios del modelar visualmente

- Para gestionar la complejidad
- Para detectar errores y omisiones cuanto antes
- Para comunicarse con los usuarios
- Para dirigir la implementación
- Para comprender el impacto de un cambio
- Para asegurarse que los recursos se despliegan eficientemente

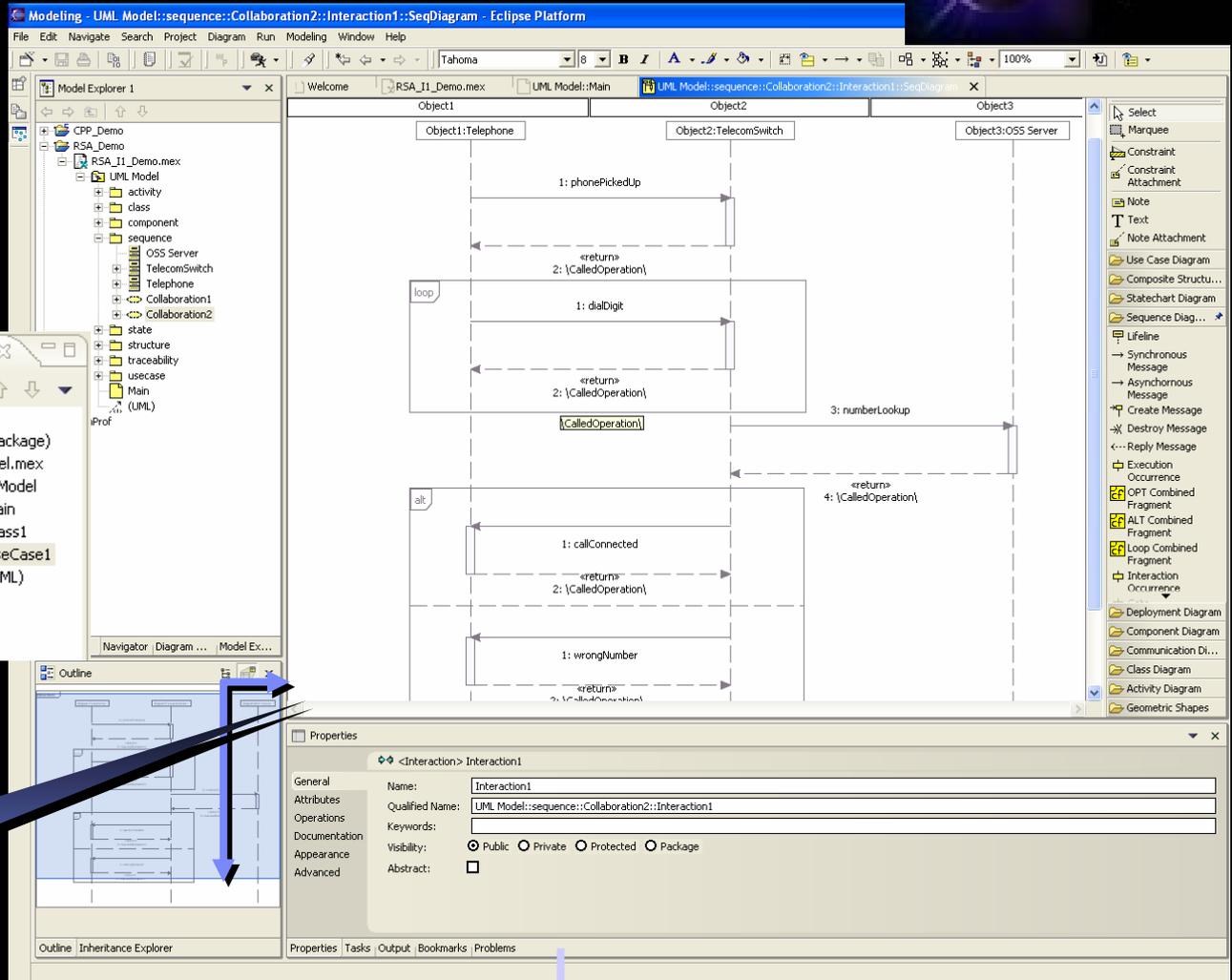
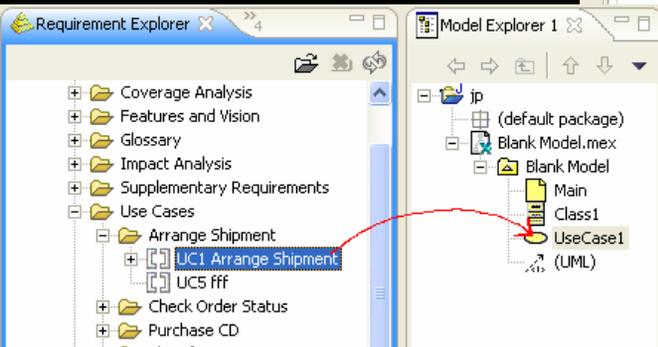


UML: *El lenguaje para el desarrollo SW*



Requisitos, modelado y construcción UML en eclipse

Perspectiva Modelado



Sincronización Manual / Automática

IBM Rose XDE Developer Plus

Agenda

- Herramientas para el ciclo completo de vida del software
- Eclipse 3.0: Framework abierto y multiplataforma
- Mejores prácticas para el desarrollo de software
- Desarrollo y arquitectura basada en modelado. UML 2.0
- **Desarrollo rápido de aplicaciones Java. J2EE y sus nuevas tecnologías**
- Gestión de la configuración: control del cambio y de versiones
- Aseguramiento de la calidad del software



WebSphere Studio Site Developer y App. Developer

- Objetivo: dar al desarrollador **herramientas que aumenten su productividad** sobre del desarrollo web y java tradicional (entre 3x y 10x), automatizando tareas de infraestructura y generando **aplicaciones mejor mantenibles**.

Application Developer

Site Developer

- Usuario: Desarrollador web
- Herramientas para:
 - Desarrollo web
 - Java Server Faces / SDO
 - Desarrollo Struts
 - HTML/JSP/Servlets
 - Servicios web
 - XML
 - Editor visual para Java
- Servidores: IBM WAS, BEA WebLogic

- Usuario: Desarrollador J2EE
- Herramientas para:
 - J2EE
 - Visualización UML
 - Rendimiento y trazas
 - Pruebas componentes
 - ClearCase LT
- Servidores: IBM WAS, BEA WebLogic

Servidores: IBM WAS, BEA WebLogic

WebSphere Studio Workbench

- Versión de eclipse soportada por IBM

- Base para herramientas de IBM

- Plataforma universal para herramientas
- Proyecto de código abierto

Eclipse

- Provides frameworks for tool builders to focus on tool building



Herramientas integrales para el diseño e implementación de un portal web

- Planear, diseñar y mantener visualmente la estructura del sitio web
 - Gestión de plantillas para mantener la consistencia de los diferentes estilos gráficos y poder cambiarlos de manera rápida y sencilla.
 - Generación de elementos de navegación (no hay links rotos).

Estructura del sitio web

Aplicamos estilo arrastrando a la página

The screenshot shows the 'Project Navigator' and 'Navigation Detail' views. The tree structure includes: Home, Company, News (with sub-categories A, B, C and sub-items B-1, B-2, B-3), and Products (with sub-categories A, B, C and sub-items A-1, B-1, C-1, A-2, B-2, C-2, A-3, B-3, C-3). A blue arrow points from the 'Page Template' gallery to the 'Navigation Detail' view, which shows a preview of a page with a template applied.

Barra de navegación generada

The screenshot shows a generated navigation bar with links for Home, Company, News, and Products. A red circle highlights the 'Investor Relations' link in the dropdown menu.

Investor Relations

Place Investor Relations content here.

The screenshot shows a web browser displaying the 'Investor Relations' page. The navigation bar at the top includes Home, Company, News, and Products. The main content area is currently empty, with the text 'Place Investor Relations content here.' below the heading.

Información detallada de páginas

The screenshot shows a detailed view of page information, including a table with columns for File Name, Page Title, Page, Last Modified, Date, and Size. The table lists various pages such as Home, Company, News, and Products, along with their respective page titles and last modified dates.

Información detallada de páginas

The screenshot shows a detailed view of page information, including a table with columns for File Name, Page Title, Page, Last Modified, Date, and Size. The table lists various pages such as Home, Company, News, and Products, along with their respective page titles and last modified dates.

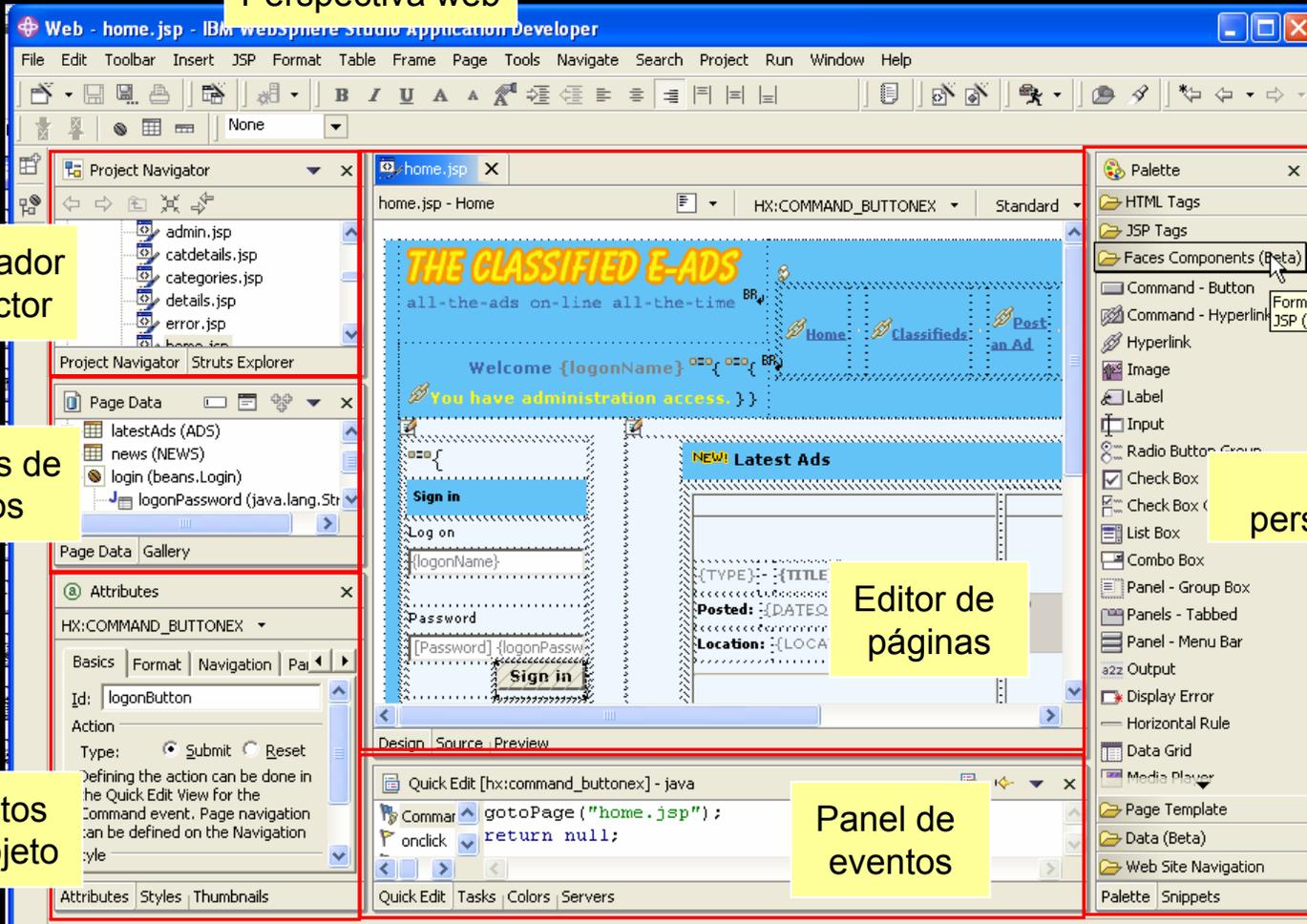
Desarrollo rápido de web en Java con Java Server Faces

Perspectiva web

Navegador proyector

Objetos de datos

Atributos del objeto



Paleta personalizable

Editor de páginas

Panel de eventos

Comparaciones entre tecnologías JSP

	Java Server Faces	JSP	JSP y Struts
Componentes	<ul style="list-style-type: none"> ▪ Componentes UI ricos de datos con eventos ▪ Componentes particularizados 	<ul style="list-style-type: none"> ▪ Standard Tags (JSTL) no son UI y básicas ▪ Componentes particularizados via librerías de tags 	<ul style="list-style-type: none"> ▪ Librería de tags específica Struts ▪ Componentes UI básicos
Independencia de dispositivo	<ul style="list-style-type: none"> ▪ Render Kits proporcionan independencia dispositivo 	<ul style="list-style-type: none"> ▪ Ninguna 	<ul style="list-style-type: none"> ▪ Ninguna
Gestión de validaciones y errores	<ul style="list-style-type: none"> ▪ Framework de validación 	<ul style="list-style-type: none"> ▪ Ninguna 	<ul style="list-style-type: none"> ▪ Framework de validación
Scripting	<ul style="list-style-type: none"> ▪ Se pueden añadir scripts a eventos ▪ Los scripts pueden acceder a todos los componentes 	<ul style="list-style-type: none"> ▪ Incrustados en código de la página 	<ul style="list-style-type: none"> ▪ Scripts incluidos en las clases "action" de Java ▪ Los scripts tienen acceso a datos del formulario pero no a sus componentes
Navegación de páginas	<ul style="list-style-type: none"> ▪ Si 	<ul style="list-style-type: none"> ▪ Ninguna 	<ul style="list-style-type: none"> ▪ Si
Gestión de sesión	<ul style="list-style-type: none"> ▪ Automática 	<ul style="list-style-type: none"> ▪ Manual 	<ul style="list-style-type: none"> ▪ Manual



Otras herramientas de productividad

- Desarrollo automatizado de **Web Services** a partir de aplicaciones y datos existentes.
 - ▶ P.e. bases de datos, beans de java, XML, interfaces java a aplicaciones existentes
 - ▶ Generación automática de infraestructura: UDDI, WSDL...
- Desarrollo de aplicaciones basadas en **Struts**
 - ▶ Editor visual “arrastrar y soltar” de navegación y de configuración, etc.
- Asistencia al trabajo con **XML**
 - ▶ Generación de XML desde java o bases de datos
 - ▶ Asistencia a la creación de XSL, DTD, Schema y otros artefactos relacionados



Productividad para J2EE en WS Application Developer

- Perspectiva J2EE para trabajar con artefactos de esta plataforma
 - ▶ Edición visual de las configuraciones J2EE

- Herramientas para automatizar el desarrollo de EJBs (1.1, 2.0)
 - ▶ Crear asistidamente EJBs desde recursos existentes (tablas,java...)
 - ▶ Generar código para acceder a EJBs desde otros código Java
 - ▶ Mapeo a bases de datos: top-down, up-down, meet-in-the-middle

- Automatización de las pruebas unitarias EJB (Universal Test Client)

- Asistencia a trabajar con proyectos java externos a aplicacion J2EE

- Depuración en caliente
 - ▶ Examinar y modificar código y datos ejecutándose en un servidor remoto sin pararlo



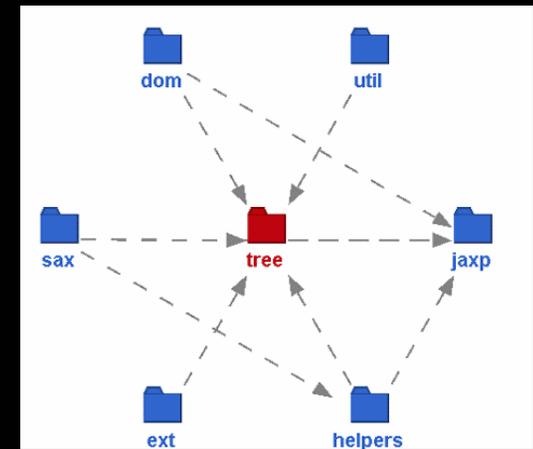
Nuevas tecnologías para la calidad del código estático

The screenshot shows the IBM WebSphere Studio Application Developer interface. The main editor displays a Java class with an `equals` method. The `Explanations` pane on the right provides details about errors in the implementation, such as 'equals(java.lang.Object) boolean is incorrectly implemented'. The `Tasks` pane at the bottom lists 14 items, including 'Missing cleanup method', 'Referenced methods: 50 Unreference methods: 5 Synthesized methods: 1', and several instances of 'equals(java.lang.Object) boolean is incorrectly implemented'.

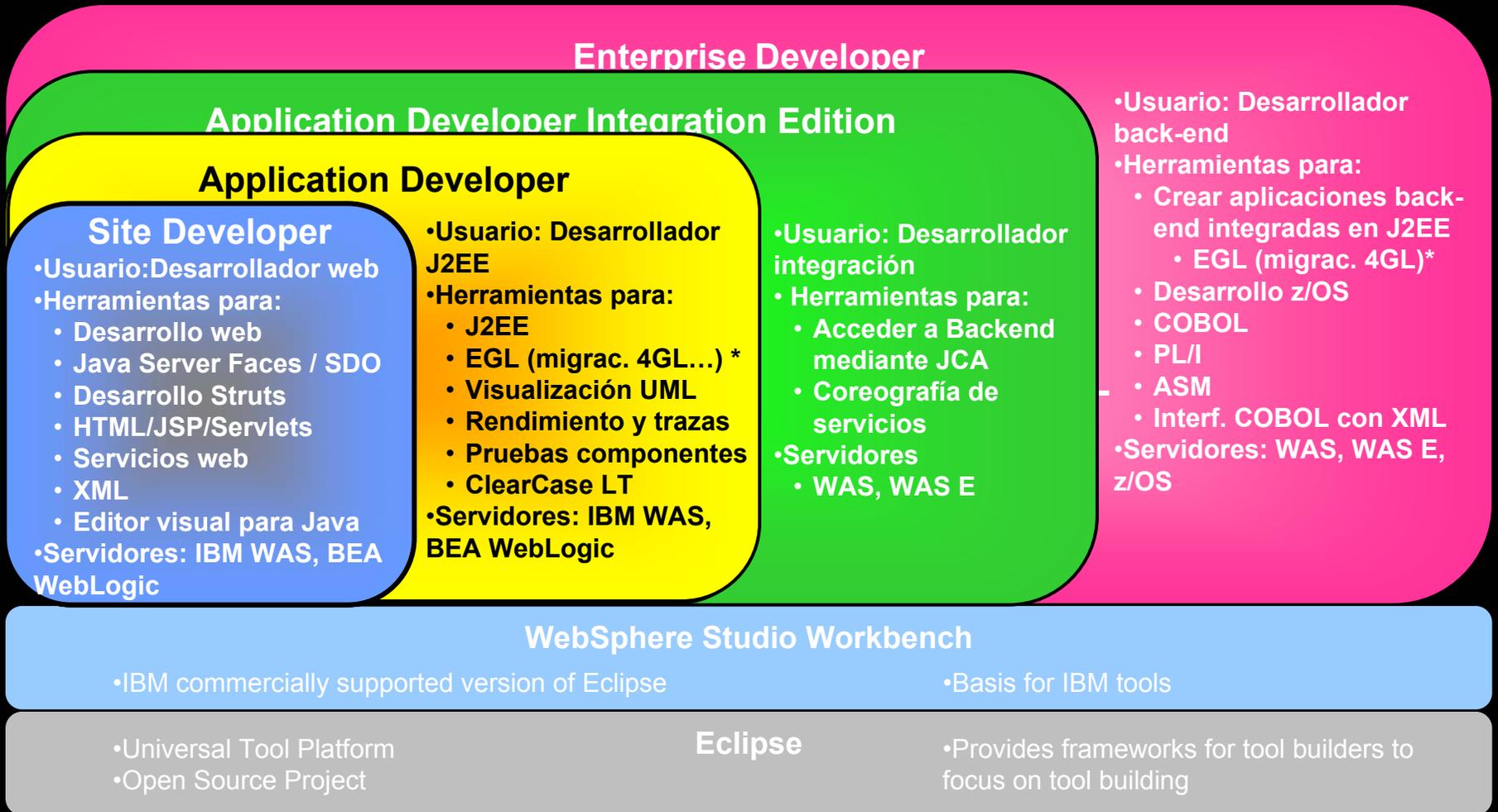
The screenshot shows the 'What If' analysis tool. It displays a dependency graph with nodes and edges, and a list of 'Potentially Affected Objects' on the right. The list includes classes like `InputSource`, `DefaultHandler`, `DefaultValidationErr...`, `XmlDocument$1`, `Parser2$NullHandler`, `Parser2`, `Parser2$DocLocator`, `ValidatingParser`, `ValidatingParser$1`, `ValidatingParser$E...`, `ValidatingParser$M...`, `ValidatingParser$C...`, `XMLReaderImpl`, `XMLReaderAdapter`, `DocumentBuilderL...`, `EntityResolver`, `XMLReader`, `SAXParserImpl`, `SAXParserFactoryI...`, `ParserAdapter`, `ParserAdapter$Attr...`, `XMLFilterImpl`, `XMLFilter`, `XMLReaderAdapter`, `XMLReaderFactory...`, `XMLReaderFactory`, `Parser`, and `ParserFactory`.

Analizador de código estático (prev.5.1)

Análisis estructural para Java (prev.5.1)



Familia Studio: Escalable a sus necesidades



* Próximamente



Agenda

- Herramientas para el ciclo completo de vida del software
- Eclipse 3.0: Framework abierto y multiplataforma
- Mejores prácticas para el desarrollo de software
- Desarrollo y arquitectura basada en modelado. UML 2.0
- Desarrollo rápido de aplicaciones Java. J2EE y sus nuevas tecnologías
- **Gestión de la configuración: control del cambio y de versiones**
- Aseguramiento de la calidad del software



Gestión del Cambio y de los Activos

- ✓ Gestión del Cambio
- ✓ Gestión de los Activos
- ✓ Automatizar y acelerar el workflow

Gestión de la Configuración del Software (SCM)



Beneficios de Negocio

- Respuesta rápida al cambio
- Equipos virtuales y desarrollo en paralelo,
- Reutilización y protección de activos

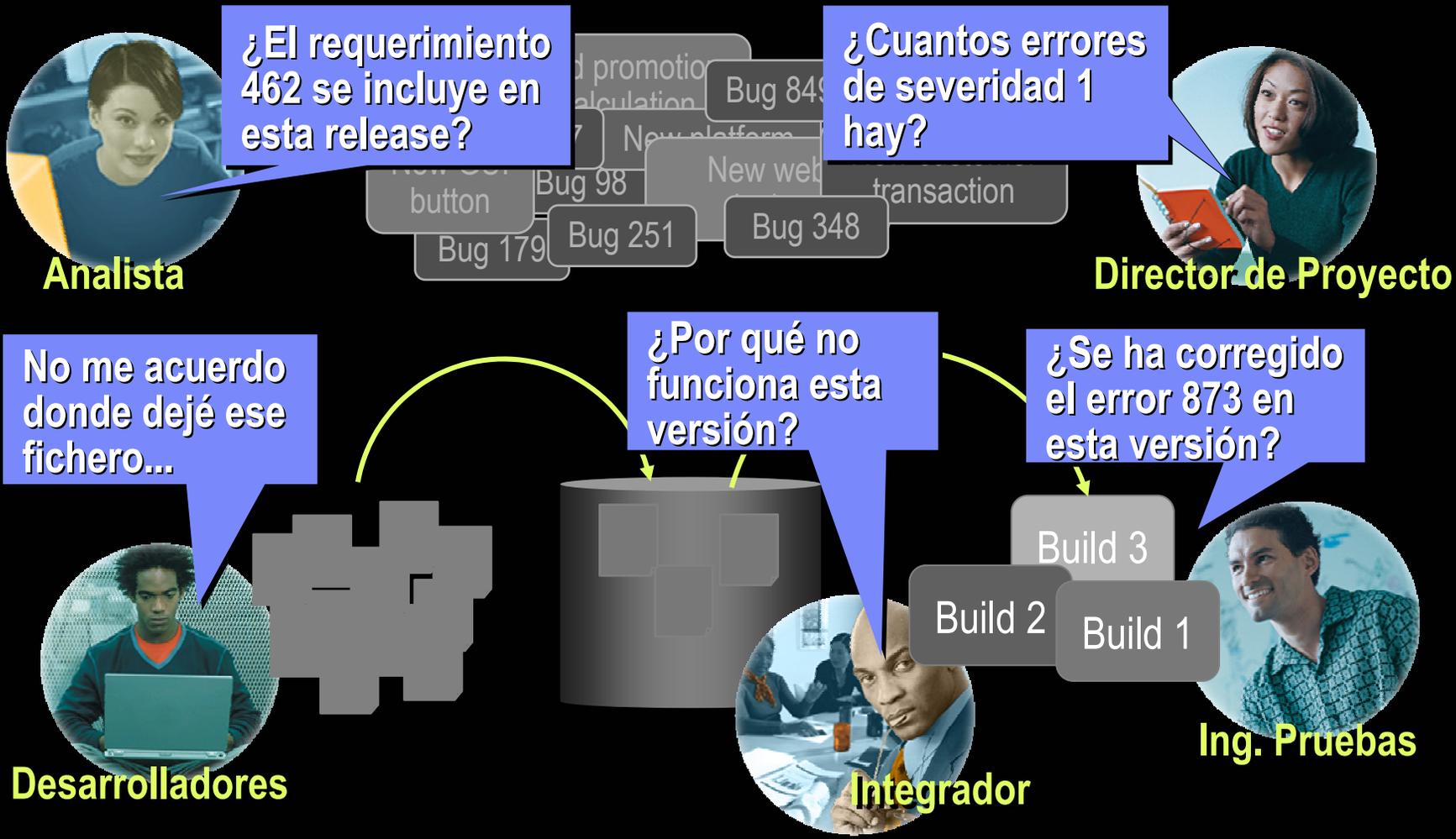


Beneficios IT

- Workflow flexible y soporte de procesos
- Gestión de equipos distribuidos
- Seguimiento a lo largo de todo el ciclo de vida



Gestión del Cambio en el proceso de desarrollo de SW



Dificultades del Desarrollador:



Desarrollador

- Acceso a la versión adecuada del producto.
- Desarrollar y probar en un entorno estable.
- Sincronización con el resto del equipo.
- Cómo resolver defectos sin afectar al desarrollo



Dificultades del Desarrollador:



Desarrollador

- Acceso a la versión adecuada del producto.
- Desarrollar y probar en un entorno estable.
- Sincronización con el resto del equipo.
- Cómo resolver defectos sin afectar al desarrollo

Dificultades del Jefe de Proyecto:



Jefe de Proyecto

- Evaluar el estado del proyecto.
- Optimizar la comunicación entre los miembros del Proyecto.
- Gestionar varios Proyectos en paralelo.
- Establecer Procesos de control y seguimiento del Proyecto.



Dificultades del Desarrollador:



Desarrollador

- Acceso a la versión adecuada del producto.
- Desarrollar y probar en un entorno estable.
- Sincronización con el resto del equipo.
- Cómo resolver defectos sin afectar al desarrollo

Dificultades del Jefe de Proyecto:



Jefe de Proyecto

- Evaluar el estado del proyecto.
- Optimizar la comunicación entre los miembros del Proyecto.
- Gestionar varios Proyectos en paralelo.
- Establecer Procesos de control y seguimiento del Proyecto.

Dificultades del Integrador:



Integrador

- Gran cantidad de versiones de archivos.
- Dependencias entre componentes.
- Pases a producción consistentes.



Gestión del Cambio y de Activos: enfoque

- ✓ Establecer un **proceso** que permita que los equipos trabajen rápida y consistentemente
- ✓ **Capturar**, versionar y relacionar información desde los modelos, al código, a las pruebas
- ✓ **Analizar** el impacto de los cambios a lo largo del ciclo de vida de la aplicación
- ✓ Permitir que los equipos **construyan** y **reutilicen activos** en cualquier momento y lugar
- ✓ **Medir** el progreso del proyecto utilizando los datos actualizados del desarrollo

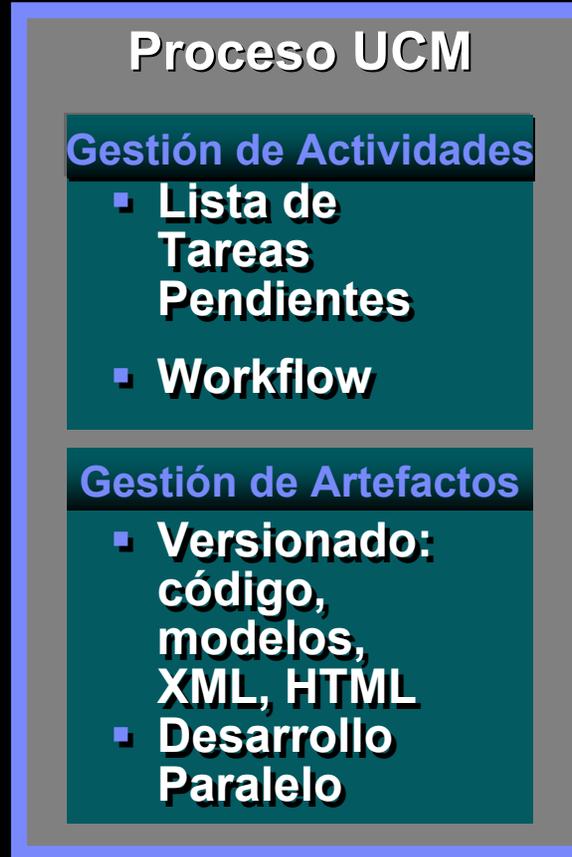


Unificar la Gestión de Artefactos y Actividades

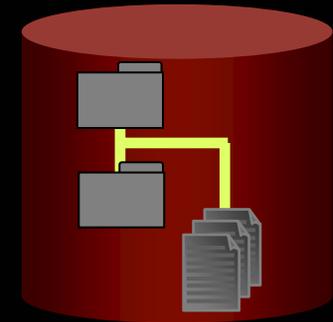
Asegurar resultados predecibles

Procesos basados en las Mejores Practicas, Out of the Box

- Décadas de experiencia en Gestión de Configuración
- Cientos de implantaciones exitosas
- Fácil de adoptar
- Optimizado para su aceptación por equipos



Gestión de Objetos a Alto-nivel

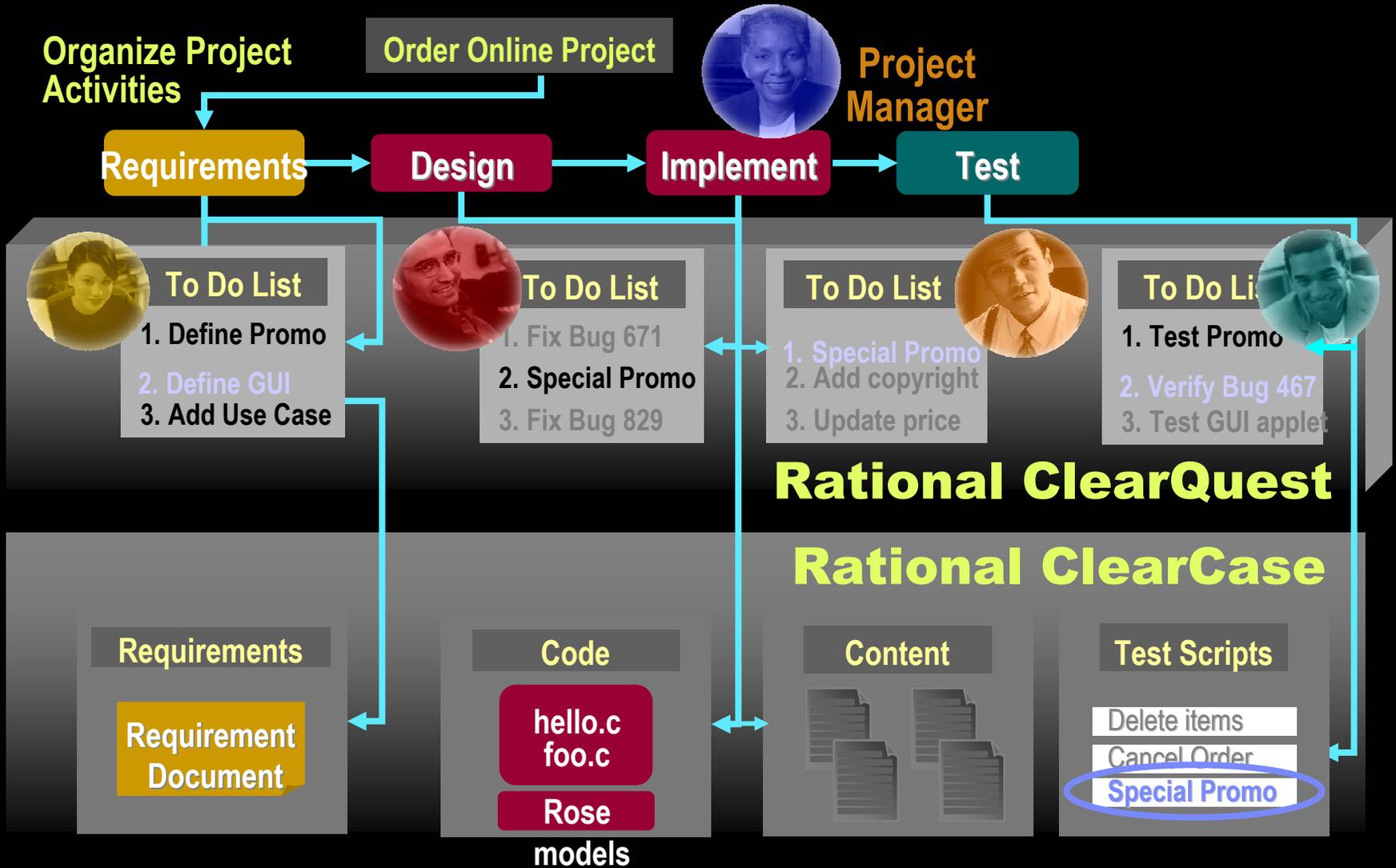


- ✓ Componentes
- ✓ Subsistemas
- ✓ Configuraciones

IBM Rational ClearCase e IBM Rational ClearQuest



Control de los Cambios del Software

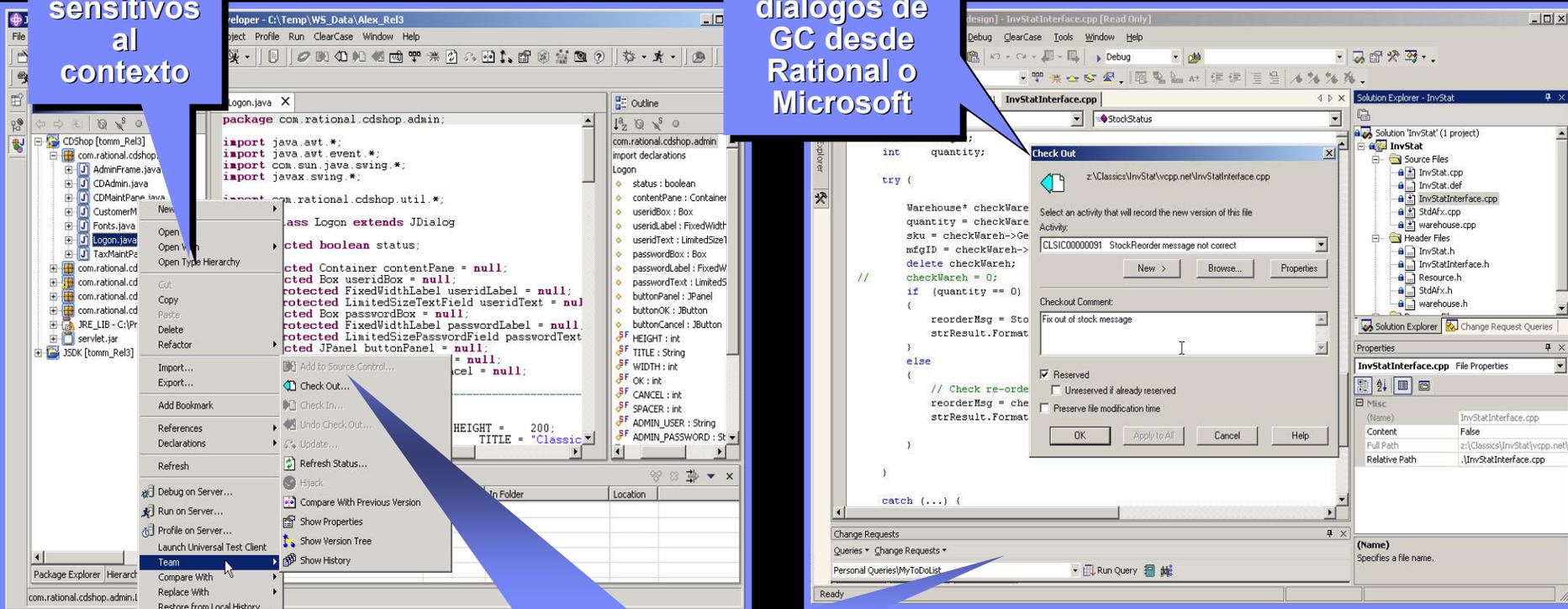


Gestión del Cambio dentro del Entorno de Desarrollo

Integración transparente en el entorno de desarrollo

Menús sensitivos al contexto

Elección de diálogos de GC desde Rational o Microsoft



Gestión de Configuración integrada con IBM WebSphere Studio y Microsoft Visual Studio .NET

IBM Rational ClearCase

Agenda

- Herramientas para el ciclo completo de vida del software
- Eclipse 3.0: Framework abierto y multiplataforma
- Mejores prácticas para el desarrollo de software
- Desarrollo y arquitectura basada en modelado. UML 2.0
- Desarrollo rápido de aplicaciones Java. J2EE y sus nuevas tecnologías
- Gestión de la configuración: control del cambio y de versiones
- **Aseguramiento de la calidad del software**



Aseguramiento continuo de la calidad

- ✓ Construcción con calidad desde el principio
- ✓ Automatización de pruebas
- ✓ Comunicación entre análisis, desarrollo y pruebas



Beneficios Negocio

- Menos riesgo en las implantaciones
- Ciclos rápidos
- Aseguramiento de los niveles de satisfacción



Beneficios IT

- Trazabilidad completa
- Aceleración de la búsqueda y resolución de problemas
- Aceleración del ciclo de pruebas



Aseguramiento continuo de la calidad: Asegurar el éxito

- ✓ Hacer que la calidad sea uno de las metas del **proceso de la organización**
- ✓ **Validar** que los requisitos están alineados con los clientes y sus necesidades
- ✓ **Construir con calidad** para conseguir funcionalidad, rendimiento, escalabilidad, mantenimiento, extensibilidad, seguridad
- ✓ **Asegurar** la calidad pronto y constantemente
- ✓ **Gestionar el cambio** para asegurar un correcto despliegue
- ✓ **Monitorizar** para capturar, aislar y reaccionar ante incumplimientos de nivel de servicio



Organizar las actividades de pruebas



**Gestión de toda la información de pruebas
y accesible a todo el equipo**



Asegurar la calidad mientras se codifica

Encuentra los errores antes de que rompan tu aplicación

- Herramientas de análisis que ayudan a detectar errores en la etapa de codificación
- Realizar análisis de memoria dentro del propio IDE
- Grabar los detalles de la ejecución de la aplicación en tiempo real
- Analizar aplicaciones con o sin debugger

The screenshot displays the IBM Rational IDE interface. On the left, a code editor shows a Java class with methods `B` and `R_f2()`. The `B` class contains a `for` loop. The `R_f2()` method calls `C.C1()`. On the right, a sequence diagram illustrates the execution flow between objects `APIA`, `APIB`, and `APIC`. The diagram shows messages like `APIA.A_f2()`, `APIB.B_f1()`, and `APIC.C_f1()` with associated time stamps and object lifelines.

Generación automática diagramas secuencia

Tiempos, cobertura, threds y análisis de memoria

IBM Rational PurifyPlus / IBM Rose XDE Developer Plus

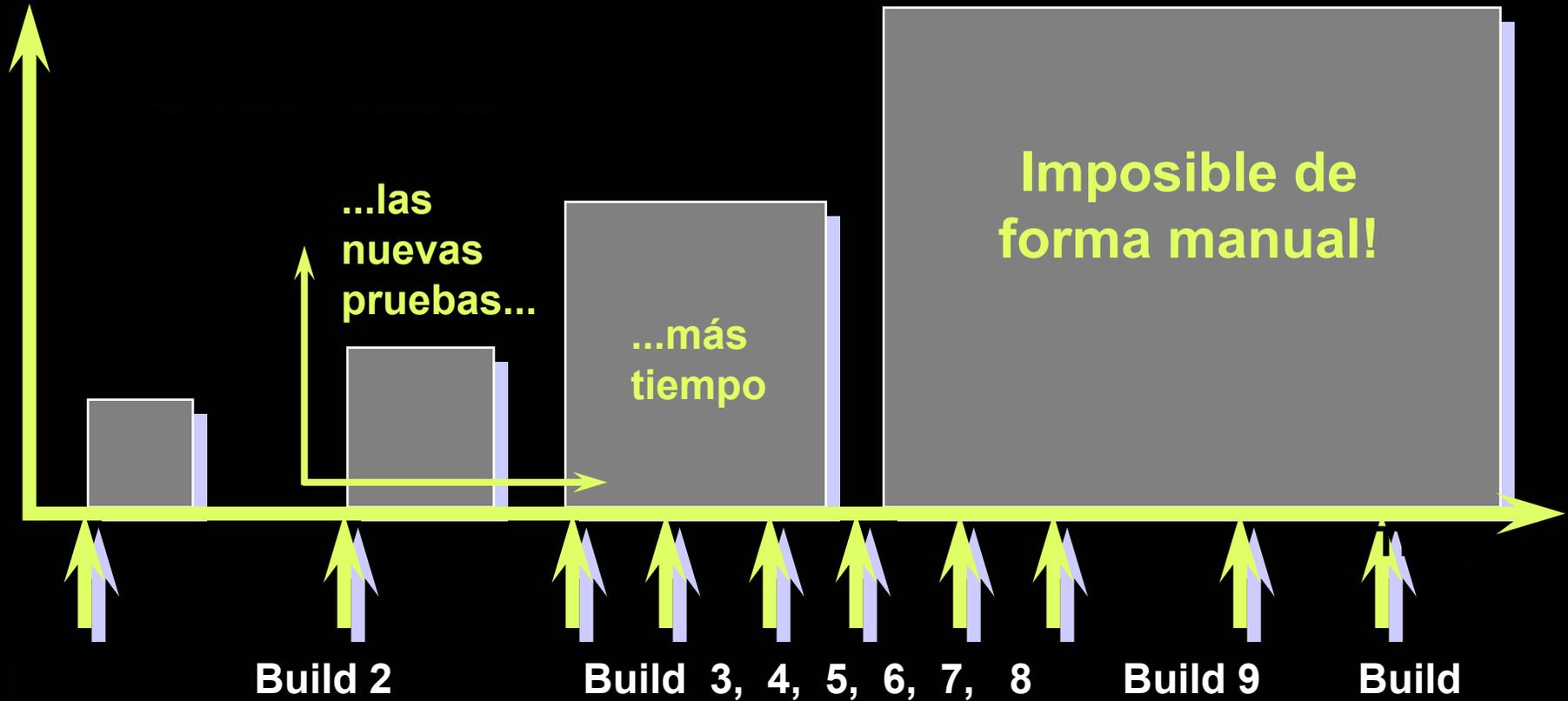


Aseguramiento de la calidad: Automatización de pruebas

Probar en cada iteración



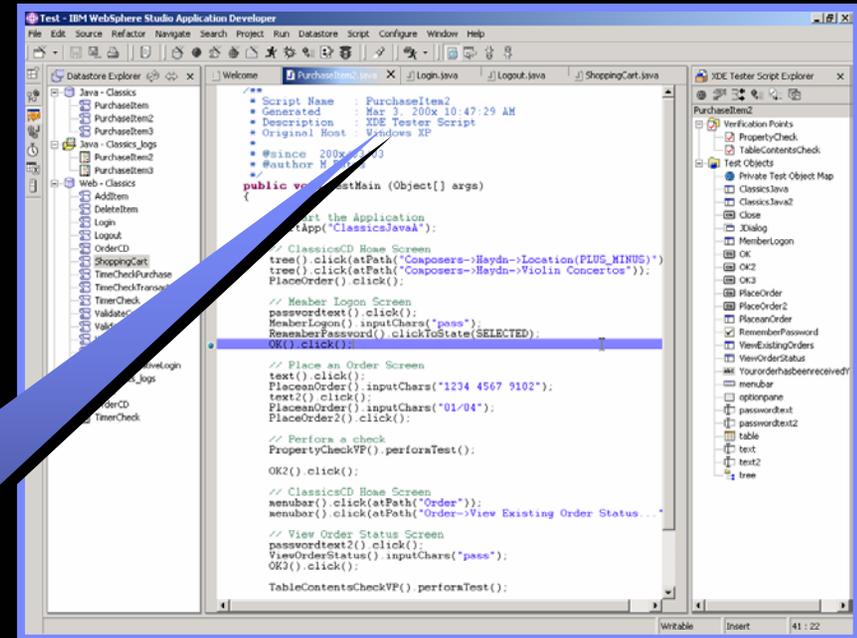
Aseguramiento de la calidad: Automatización de pruebas



Aseguramiento de la calidad: Automatización de pruebas

- Automatización de pruebas funcionales y de carga en diferentes entornos y tecnologías:
 - ▶ Java
 - ▶ Web
 - ▶ .NET
 - ▶ Terminales 3270 y 5250

Pruebas integradas
en el entorno de
desarrollo



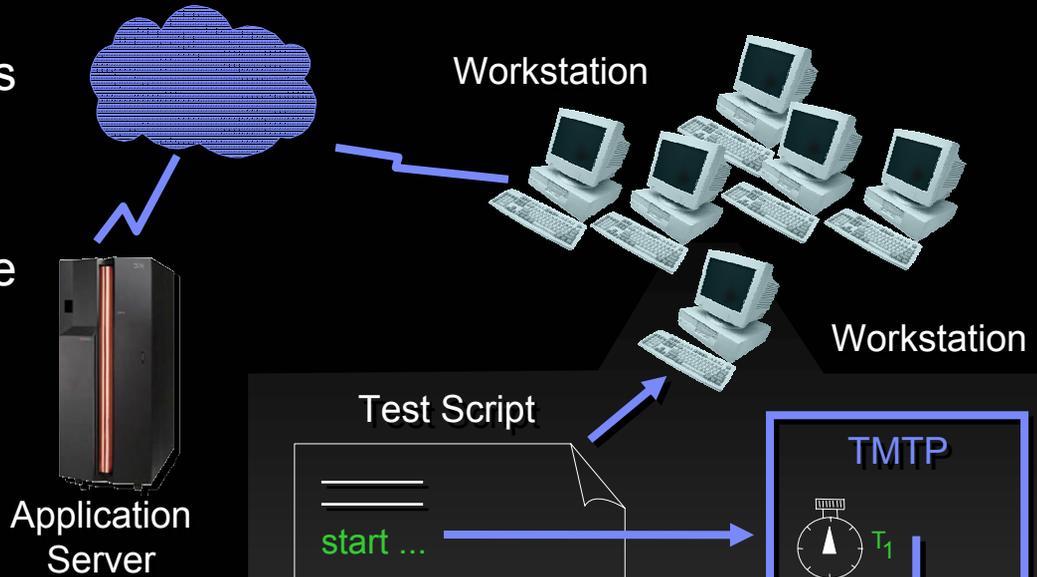
IBM Rational Functional Tester



Asegurar la calidad después del despliegue

Monitorizar el comportamiento del sistema

- Monitorizar la disponibilidad de las transacciones en intervalos regulares
- Medir los tiempos de respuesta de muchos usuarios concurrentes
- Identificar y aislar problemas de rendimiento y escalabilidad



Reutilización de scripts de pruebas para la monitorización

Integración con Tivoli Monitoring for Transaction Performance



Apéndice

- Mejora de procesos:
 - ¿Cómo implementar metodologías y herramientas en una organización? ¡¡ Paso a paso!!



Alcance de las mejoras: metodología, herramientas y organización

