

IV Jornadas de  
Ingeniería del Software  
y Bases de Datos  
(JISBD'99)



Editores:  
Pere Botella  
Juan Hernández  
Fèlix Saltor

Grupo de Paralelismo  
Departamento de Informática  
Escuela Politécnica  
Universidad de Extremadura

Cáceres,  
del 24 al 26 de noviembre de 1999

# IV Jornadas de Ingeniería del Software y Bases de Datos (JISBD'99)

Editores:

Pere Botella  
Juan Hernández  
Fèlix Saltor

Grupo de Paralelismo  
Departamento de Informática  
Escuela Politécnica  
Universidad de Extremadura

Cáceres, del 24 al 26 de noviembre de 1999

ACTAS DE LAS IV JORNADAS DE INGENIERÍA DEL SOFTWARE  
Y BASES DE DATOS (JISBD'99)

EDITORES:

Pere Botella  
Juan Hernández  
Fèlix Saltor

ORGANIZADAS POR:

Grupo de Paralelismo  
Departamento de Informática  
Escuela Politécnica  
Universidad de Extremadura

ENTIDADES COLABORADORAS:

ATI Asociación de Técnicos de Informática  
Caja Extremadura  
Diputación Provincial de Cáceres. Institución  
Cultural EL BROCENSE  
Dirección General de Enseñanza Superior e  
Investigación Científica  
Junta de Extremadura. Consejería de Educación,  
Ciencia y Tecnología  
Teleserver Extremadura

© Los autores

Primera edición, 1999

I.S.B.N.: 84-699-0956-8

Depósito legal: CC-238-1999

Imprime: Gráficas Morgado, S.L. - Cáceres.

Aldana, J.F.  
Alvarez, D.  
Aramburu,  
Bardanga, F.  
Blesa, P.  
Brosaboa, N.  
Calero, C.  
Cerverón,  
Cerral, A.  
Cuadra, D.  
Cueva, J.M.  
De Miguel,  
Diegado, C.  
Díaz, O.  
Duran, A.  
Duran, M.  
García, F.  
García, S.  
Garrv, E.  
Gómez, A.  
Gomis, O.  
Jume, A.  
Kaufman,  
Lain, C.  
Lopez, J.  
Lopez, A.  
Manzano,  
Marcos, I.  
Martin, C.  
Martinez,  
Martinez,  
Martinez,  
May, A.  
Miquel, I.  
Mojeno,  
Nieto, C.  
Ortega,  
Ortiz, F.  
Parama,  
Pazos,  
Pineda,  
Pérez,  
Pérez,  
Piatini,  
Places,  
Polo, M.  
Polo, A.  
Ruiz, E.  
Sañez,  
Sallas,  
Simos,  
Sinz, P.  
Tigens,  
Toro, F.  
Torres,  
Vassil,  
Vela, F.

## COMITÉ DE PROGRAMA DE INGENIERÍA DEL SOFTWARE

### PRESIDENTE

Pere Botella (Universitat Politècnica de Catalunya)

### MIEMBROS

Idoia Alarcón	(Universidad Autónoma de Madrid)
Asunción Barredo	(Universidad de Deusto)
José H. Canós	(Universitat Politècnica de València)
Buenaventura Clares	(Universidad de Granada)
Carlos Delgado Kloos	(Universidad Carlos III de Madrid)
Carmelo R. Fernández	(Euskal Herriko Unibertsitatea)
Xavier Franch	(Universitat Politècnica de Catalunya)
Juan Garbajosa	(Universidad Politécnica de Madrid)
Jesus García Molina	(Universidad de Murcia)
Juan Hernández	(Universidad de Extremadura)
Jon Iturrioz	(Euskal Herriko Unibertsitatea)
Natalia Juristo	(Universidad Politécnica de Madrid)
Marta López	(Universidade da Coruña)
José Manuel Marqués	(Universidad de Valladolid)
Mario Piattini	(Universidad de Castilla-La Mancha)
Isidro Ramos	(Universitat Politècnica de València)
Maria R. Sancho	(Universitat Politècnica de Catalunya)
Miguel Toro	(Universidad de Sevilla)
Ambrosio Toval	(Universidad de Murcia)
Edmundo Tovar	(Universidad Politécnica de Madrid)
José Maria Troya	(Universidad de Málaga)
Javier Tuya	(Universidad de Oviedo)
Antonio Vallecillo	(Universidad de Málaga)
Angel Velázquez	(Universidad Rey Juan Carlos)

## PRÓLOGO

El presente volumen contiene los trabajos presentados en las IV Jornadas de Ingeniería de Software y las IV Jornadas de Bases de Datos, *JISBD'99*, celebradas en Cáceres los días 24, 25 y 26 de noviembre de 1999.

Tanto las Jornadas de Ingeniería de Software como las Jornadas de Investigación y Docencia en Bases de Datos han constituido, en los últimos años, la referencia obligada para el intercambio de experiencias de los grupos españoles que desarrollamos nuestra investigación en estos campos. Asimismo, hay que destacar el notable incremento en la participación de grupos de investigación Iberoamericanos en esta cuarta edición.

Las IV Jornadas de Ingeniería de Software y Bases de Datos representan una edición especial, donde coinciden por primera vez estos dos eventos nacionales ya consolidados: de una parte, las Jornadas de Ingeniería de Software, que tuvieron lugar en Sevilla, San Sebastián y Murcia en los años 1996, 1997 y 1998, respectivamente; de otra, las Jornadas de Investigación y Docencia en Bases de Datos, celebradas con anterioridad en A Coruña, Madrid y Valencia en esos mismos años.

El objetivo principal de estas Jornadas es propiciar un foro de encuentro y discusión donde los grupos de investigación en las áreas de Ingeniería de Software y Bases de Datos, tanto universitarios como profesionales, puedan compartir experiencias, divulgar resultados, y donde se establezca un marco propicio de colaboración entre los distintos sectores.

Las IV Jornadas se han organizado en doce sesiones técnicas paralelas, dos conferencias invitadas y dos mesas redondas.

Para las sesiones técnicas se recibieron un total de 61 trabajos en Ingeniería del Software y 26 trabajos en Bases de Datos. Cada artículo fue revisado por al menos dos miembros del Comité de Programa, quienes tras una ardua labor dado el número y calidad de las contribuciones recibidas, seleccionaron 23 trabajos en el área de Ingeniería de Software y 11 en el área de Bases de Datos. Adicionalmente, en cada una de las áreas se consideraron 8 y 5 trabajos para su presentación en la modalidad de artículos cortos, respectivamente. A mediados del mes de julio, los autores recibieron los comentarios y opiniones de los miembros de los respectivos comités relativos a sus trabajos, para incorporar en las versiones finales recogidas en este volumen, las modificaciones recomendadas por los revisores.

Las conferencias invitadas se han organizado teniendo en cuenta las inquietudes e intereses de ambos colectivos, y tuvimos el honor de contar con la presencia de personalidades prestigiosas en sus respectivas especialidades. El Dr. Mehmet Aksit, profesor de la Universidad de Twente (Holanda), director y líder del grupo TRESE (*Twente Research and Education on Software Engineering*), quien impartió la conferencia "Quality-Oriented Software Engineering". El Dr. Stephen Cannan, de *james martin + co* (Holanda) y miembro del *ISO Database Languages Working Group*, quien impartió la conferencia "The New SQL standard. Good, Bad or simply Ugly?"

Las mesas redondas trataron temas de interés comunes a ambos colectivos. De un lado "¿Para qué la tesis doctoral?", coordinada por el Dr. Oscar Díaz. De otro, "La Ingeniería de Software y las Bases de Datos en la Industria: cómo trasladar la investigación a la práctica", coordinada por el Dr. Mario Piattini. A todos los participantes en las mismas, quisiera desde aquí hacer constar también mi agradecimiento.

Estas jornadas no hubieran sido posibles sin el trabajo voluntario y desinteresado de muchas personas. A todos los miembros de los respectivos Comités de Programa y a los revisores externos, quisiera agradecerles el esfuerzo realizado en todo el proceso de revisión y selección de los trabajos. A sus dos Presidentes, Dr. Pere Botella y Dr. Fèlix Saltor, de la Universitat Politècnica de Catalunya, mi gratitud por el esfuerzo realizado y la colaboración prestada en todo momento.

Asimismo, quiero agradecer a todos los miembros del Comité Organizador el buen ánimo que han tenido y las horas empleadas que, a veces, sin tiempo disponible debido a otras ocupaciones no menos importantes, han dedicado en la realización de la multitud de tareas que conlleva la organización de este tipo de eventos.

Finalmente y en nombre de todo el Comité Organizador, quiero expresar nuestra gratitud a todos los autores que enviaron sus artículos y a todos los participantes en estas Jornadas, sin los cuales, esta cuarta edición no hubieran tenido lugar. Todo el Comité Organizador espera que este encuentro haya sido de vuestro interés.

<b>BD4: Métodos de acceso e indexación .....</b>	<b>97</b>
<i>Moderador: Manuel Barrena (Universidad de Extremadura)</i>	
Avances en el diseño del índice m-Q paginado .....	99
<i>M. Salas, A. Polo, A.I. López, J.C. Manzano</i>	
Chronology: Una aproximación al almacenamiento y recuperación de información de actualidad .....	111
<i>R. Berlanga, M.J. Aramburu, S. García, I. Sanz</i>	
Revisando las técnicas de optimización de consultas recursivas mediante la propagación de semicomposiciones .....	123
<i>J.F. Aldana, A.C. Gómez</i>	
<b>BD5: El colectivo de investigadores de bases de datos y sus inquietudes: su estructuración, investigación y docencia .....</b>	<b>135</b>
<i>Moderadores: Adoración de Miguel (Universidad Carlos III de Madrid) Fèlix Saltor (Universitat Politècnica de Catalunya)</i>	
<b>BD6: Diseño y depuración de bases de datos .....</b>	<b>137</b>
<i>Moderadora: Matilde Celma (Universitat Politècnica de València)</i>	
Caracterización formal de métricas para bases de datos relacionales .....	139
<i>C. Calero, M. Piattini</i>	
Depuración de disparadores en bases de datos activas .....	151
<i>O. Díaz, A. Jaime</i>	

### *Sesiones técnicas de Ingeniería del Software*

<b>IS0: Artículos cortos .....</b>	<b>163</b>
<i>Moderador: Pere Botella (Universitat Politècnica de Catalunya)</i>	
Especialización en el ámbito del modelado conceptual .....	165
<i>P. Sánchez, P. Letelier, J. A. Troyano, V. Pelechano, J. Torres</i>	
Modelado orientado a objetos basado en UML conectando a una herramienta de diseño detallado .....	167
<i>M. López, I.A. Parra, R. Santaolaya, O.G. Fragoso, B.J. Ferro</i>	

Verificación formal del subsistema de identificación de clientes en una tienda virtual .....	169
<i>J. García, J. Tuya, J.A. Corrales</i>	
WinHIPE: un entorno integrado de programación funcional .....	171
<i>J.A. Velázquez, A. Presa</i>	
Metodología de desarrollo de sistemas de información distribuidos basados en tecnología WEB .....	173
<i>P. Cuesta, J.B. García, J.M. López, R. González</i>	
Desarrollo de una arquitectura distribuida orientada a objeto para la monitorización automática de vehículos .....	175
<i>P.J. Álvarez, J. Guillo, D. Infante, F.J. Zarazaga, F.J. Nogueras, P.R. Muro-Medrano</i>	
Una crítica a la modelización conceptual .....	177
<i>O. Dieste, M. López</i>	
Instanciación de frameworks mediante SmartBooks .....	179
<i>A. Ortigosa, M. Campo, R. Moriyón</i>	
<b>IS1: Ingeniería de requisitos. Evolución y mantenimiento .....</b>	<b>181</b>
<i>Moderadora: Idoia Alarcón (Universidad Autónoma de Madrid)</i>	
Elicitación de requisitos de usuario mediante plantillas y patrones de requisitos .....	183
<i>A. Durán, B. Bernárdez, M. Toro, A. Ruiz</i>	
Una herramienta para la captura de requisitos de usuario .....	195
<i>M.A. Laguna, J.M. Marqués, F.J. García</i>	
La evolución de modelos en el desarrollo de software: los enfoques OASIS y MEDES .....	207
<i>J. Parets, J. Carsí, J.H. Canós, A. Anaya, M.V. Hurtado, M.C. Penadés, P. Paderewski, I. Ramos, M.J. Rodríguez</i>	
Elaboración de una metodología para el mantenimiento de software .....	219
<i>M. Polo, M. Piattini</i>	
<b>IS2: Patrones de diseño. Objetos activos .....</b>	<b>231</b>
<i>Moderador: Carlos Delgado-Kloos (Universidad Carlos III de Madrid)</i>	
Especialización dinámica y generación automática de código: un enfoque basado en patrones de diseño .....	233
<i>V. Pelechano, J. Sánchez, O. Pastor</i>	



	Una arquitectura para una herramienta de patrones de diseño .....	245
	<i>J. Sáez, J. García, P.J. Jiménez</i>	
	Un patrón de activación de objetos activos .....	257
	<i>P. Paderewski, J. Parets</i>	
	Simulación del comportamiento en entornos de cooperación de objetos activos mediante Maude .....	269
	<i>M. Sánchez, J.M. Murillo, A. Navasa, M.A. Pérez, A. Polo</i>	
<b>IS3:</b>	<b>UML: de la teoría a la práctica .....</b>	<b>281</b>
	<i>Moderador: Miguel Toro (Universidad de Sevilla)</i>	
	Modelado formal y ejecutabilidad de diagramas de clases de UML .....	283
	<i>J.L. Fernández, A. Toval</i>	
	Modelado basado en roles con UML .....	295
	<i>M.J. Ortín, J. García</i>	
	Integración de UML en la metodología MÉTRICA .....	307
	<i>J.R. Hilerá, J. Martínez, S. Otón, J.A. Gutiérrez</i>	
	CADomo: Un entorno de soporte a la construcción de aplicaciones de domótica .....	317
	<i>C. Ocampo, I. Dapena, P. Botella</i>	
<b>IS4:</b>	<b>Orientación a objetos. Auditorías y métricas .....</b>	<b>329</b>
	<i>Moderador: Ambrosio Toval (Universidad de Murcia)</i>	
	Metodología para la aplicación de prototipos rápidos orientados a objetos .....	331
	<i>R.M. Mato, P. Valdés, A. Tamayo</i>	
	Obtención automática de clases genéricas a través de una operación de parametrización .....	343
	<i>Y. Crespo, J.J. Rodríguez, F.J. García, J.M. Marqués</i>	
	Modelo de cualificación y auditorías de elementos reutilizables de un repositorio .....	355
	<i>E. Manso, F.J. García, M.P. Romay, J.M. Marqués</i>	
	Obtención directa de métricas de complejidad a partir del conocimiento empírico .....	367
	<i>D. Villén, M. Collado</i>	

<b>IS5: Procesos. Análisis de dominios .....</b>	<b>379</b>
<i>Moderador: Buenaventura Clares (Universidad de Granada)</i>	
Hacia una herramienta de soporte al proceso software basada en la tecnología de workflow .....	381
<i>M.C. Penadés, J.H. Canós, J.Á. Carsí</i>	
Contra el arraigo de la cascada .....	393
<i>I. Gutiérrez, N. Medinilla</i>	
Un proceso de estudio del dominio aplicable en Ingeniería del Conocimiento y en Ingeniería del Software .....	405
<i>S.T. Acuña, C.M. Lasserre, V.E. Quincoces</i>	
Valoración de métodos de análisis de dominios .....	417
<i>S. Vegas</i>	
<b>IS6: Distribución, objetos y componentes .....</b>	<b>429</b>
<i>Moderador: José María Troya (Universidad de Málaga)</i>	
Distribución de objetos transparente, dinámica e independiente de la plataforma basada en disfraces .....	431
<i>F. Sánchez, J. Hernández, J.M. Murillo, R. Rodríguez</i>	
Coordinación de componentes distribuidos: un enfoque generativo basado en arquitectura del software .....	443
<i>C. Canal, L. Fuentes, E. Pimentel, J.M. Troya</i>	
Aplicación del método de las jerarquías analíticas en la elección de componentes software .....	455
<i>A. Lozano, A. Gómez</i>	

# ELABORACIÓN DE UNA METODOLOGÍA PARA EL MANTENIMIENTO DE SOFTWARE<sup>1</sup>

Macario Polo Usaola y Mario Piattini Velthuis

Grupo ALARCOS  
Escuela Superior de Informática  
Universidad de Castilla-La Mancha  
Ronda de Calatrava, 5  
13071 Ciudad Real (ESPAÑA)  
{mpolo, mpiattin}@inf-cr.uclm.es  
Teléfono: + 34 926 29 53 00, ext. 3715  
Fax: + 34 926 29 53 54

**Resumen.** El mantenimiento es la etapa menos planificable y más costosa del ciclo de vida software, llegando a absorber hasta el 90% de los costes totales. Sin embargo, la mayoría de las organizaciones, que sí emplean metodologías para ejecutar sus nuevos desarrollos, no siguen ninguna para llevar a cabo el mantenimiento. Este hecho provoca una falta de definición del proceso, que añade dificultad a las sucesivas intervenciones de mantenimiento que el software, como producto en constante evolución, sufre.

En este trabajo presentamos la visión del proceso de mantenimiento que ofrece MANTEMA, una metodología para mantenimiento de software que integra todas las actividades relacionadas con este proceso. El objetivo de MANTEMA es convertir el mantenimiento de software en un proceso controlable y mensurable mediante la identificación y definición clara de todos los objetos (software, documentos, personas, tareas...) que intervienen en el mantenimiento.

MANTEMA ha sido desarrollada en un trabajo de colaboración entre nuestra universidad y Atos ODS, una multinacional francesa entre cuyas principales actividades de negocio se encuentra la externalización de mantenimiento de software de grandes empresas.

**Palabras clave:** Mantenimiento, Ciclo de vida, Métricas.

---

<sup>1</sup> Este trabajo se encuentra parcialmente financiado por los proyectos MÁNTICA (CICYT IFD-097-0168) y MANTEMA (Atos ODS, S.A. y Universidad de Castilla-La Mancha; ATYCA, Dirección General de Tecnología y Seguridad Industrial del Ministerio de Industria y Energía).

## 1 Introducción.

El mantenimiento es la etapa más costosa del ciclo de vida software, absorbiendo en algunos casos entre el 67 y el 90% de los costes totales [14]. A pesar de estas circunstancias, la mayoría de las organizaciones no siguen ninguna metodología para el mantenimiento de su software, aunque sí que emplean alguna para los nuevos desarrollos [13].

Estas circunstancias hacen del mantenimiento un proceso descontrolado, de difícil seguimiento y poco o nada mensurable. Es necesario, por tanto, dotar a las organizaciones de técnicas metodológicas que permitan convertir el mantenimiento en un proceso plenamente controlable. Además, la mejora del proceso de mantenimiento resulta de gran interés, debido al gran número de sistemas heredados (*legacy*) que se utilizan en la actualidad [2].

Para la definición de una metodología de mantenimiento resulta interesante tomar como referencia marcos más globales, como ISO/IEC 12207 [9] o IEEE 1074 [8], ya que así obtendremos un proceso de mantenimiento perfectamente integrado con el resto de procesos.

En este trabajo presentamos MANTEMA, una metodología para el mantenimiento de software desarrollada entre nuestra universidad y Atos ODS, una organización multinacional entre cuyas principales actividades de negocio se encuentra la externalización del mantenimiento de software. Atos ODS está aplicando la metodología a diversos proyectos de grandes clientes.

MANTEMA está construida tomando el proceso de mantenimiento definido en ISO/IEC 12207 como punto de partida. La conveniencia de utilizar este estándar internacional se encuentra justificada en [16].

En MANTEMA se identifican todos los elementos que intervienen en el proceso de mantenimiento y se definen los contextos en que son utilizados. De este modo se consigue una metodología integral que aporta soluciones al mantenimiento convirtiéndolo en un proceso controlado, estructurado y totalmente mensurable.

Este trabajo está organizado del siguiente modo: en la sección 2 realizamos una introducción a la metodología MANTEMA, en la que explicamos las ideas iniciales que nos han llevado a la metodología; en la sección 3 profundizamos en los aspectos más destacables de la metodología; en la sección 4 listamos las métricas definidas para el proceso de mantenimiento. Por último, en la sección 5, exponemos nuestras conclusiones y nuestras líneas de trabajo actuales.

## 2 Planteamiento de la metodología MANTEMA.

MANTEMA aborda por entero el problema del mantenimiento, intentando disminuir los costes de todas sus actividades. Para conseguirlo, una de las primeras tareas que debe completarse es la definición clara del conjunto de actividades que se deben ejecutar a lo largo del proceso de mantenimiento.

En este sentido es útil seguir las recomendaciones de ISO/IEC 12207 [9], IEEE 1219 [7] o [18], que abogan por ejecutar de forma independiente las peticiones de modificación, en función del tipo de mantenimiento a que cada una se refiera. Ini-

cialmente, en MANTEMA definíamos cinco conjuntos de actividades, correspondientes a los siguientes cinco tipos de mantenimiento [16]:

1. Correctivo *urgente*, que tiene lugar cuando existe un error en el software que bloquea el funcionamiento normal de la organización o de la aplicación, siendo crítico el tiempo de solución.
2. Correctivo *no urgente*, que ocurre cuando existe un error en el software que no es crítico, pero que tal vez impida el funcionamiento de la aplicación o el normal funcionamiento de la empresa en un periodo de tiempo relativamente corto.
3. Perfectivo, que tiene lugar cuando se van a añadir nuevas características o funcionalidades al software en explotación.
4. Adaptativo, que se aplica cuando el software en explotación va a cambiarse para que continúe funcionando correctamente en un entorno cambiante.
5. Preventivo, que es aplicado cuando se desea mejorar las características internas de un producto para hacerlo, por ejemplo, más fácilmente mantenible.

Durante la construcción de la metodología se siguió el método científico Action-Research [4], muy utilizado en proyectos que requieren la interacción de dos o más organizaciones. Fruto de la retroalimentación producida por este método, las dos organizaciones interesadas observamos que la aplicación de las primeras versiones de la metodología a casos reales de Atos ODS demostró que, a pesar de que las intervenciones de correctivo no urgente, perfectivo, preventivo y adaptativo poseen características propias que las diferencian unas de otras, las líneas de diseño de sus respectivos conjuntos de actividades eran bastante similares. Igualmente, la ejecución de las intervenciones de estos cuatro tipos seguía también patrones de comportamiento bastante parecidos, de manera que se decidió integrar estos cuatro tipos de mantenimiento en uno solo (que denominamos "planificable"), pasando a llamar "mantenimiento no planificable" al que hasta entonces había sido "correctivo urgente".

En MANTEMA completamos la estructura del proceso de mantenimiento identificando y definiendo dos conjuntos adicionales de actividades:

- En el primero, "Actividades y tareas iniciales comunes", englobamos todas las actividades que deben ser ejecutadas al comenzar el proceso de mantenimiento y que serán anteriores a la ejecución de cualquier intervención.
- El segundo, "Actividades y tareas finales comunes", agrupa las actividades que serán realizadas al finalizar el proceso de mantenimiento, y que serán ejecutadas con posterioridad a las intervenciones.

Con este planteamiento, el proceso de mantenimiento definido por MANTEMA puede entenderse como el grafo polietápico que mostramos en la figura 1.

Como ya hemos dicho, cada nodo del grafo mostrado en la Figura 1 está formado por un conjunto de actividades. Siguiendo la terminología utilizada en ISO/IEC 12207 [9], cada actividad está formada, a su vez, por un conjunto de tareas.

La ya mencionada identificación de todos los objetos relacionados con el proceso de mantenimiento es realizada "a nivel de tarea", ya que, para cada una de éstas, definimos:

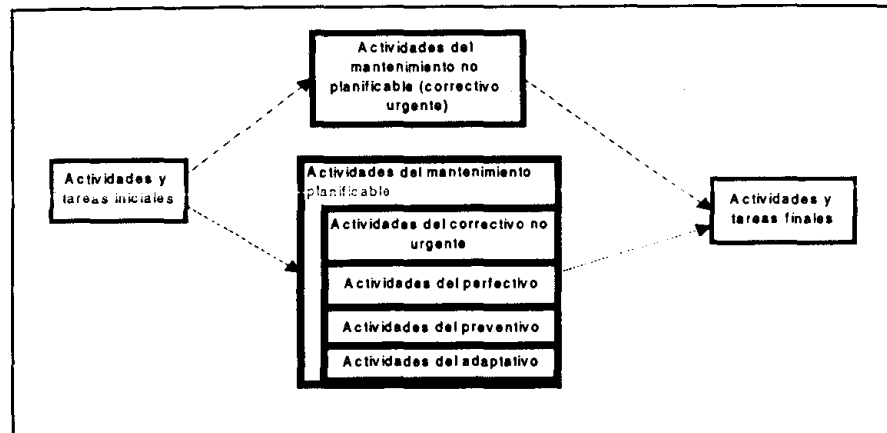


Figura 1. El proceso de mantenimiento en MANTEMA.

- Entradas, que son los objetos necesarios para la correcta ejecución de la tarea. Estos objetos podrán ser programas, documentos, etc., y podrán ser tomados bien de tareas anteriores, bien del entorno del proceso.
- Salidas, que son los objetos que se generan tras la ejecución de la tarea. Las salidas podrán ir dirigidas a otras tareas posteriores o bien al entorno.
- Técnicas, que son las técnicas que pueden utilizarse para ejecutar la tarea.
- Interfaces con otros procesos, que se establecerán durante la ejecución de la tarea con el resto de procesos definidos por la organización para el ciclo de vida software (por ejemplo, con el proceso de "Gestión de la configuración").
- Responsables, que son las personas encargadas de la ejecución de la tarea.
- La determinación de los responsables de cada tarea es un tema que resulta de gran interés, ya que la estructura y otras propiedades de los grupos de trabajo tiene una gran influencia sobre su rendimiento [11]. Así, por ejemplo, Brooks afirma que la productividad del grupo de trabajo decrece conforme aumenta su número de miembros [3]. La reducción de cualquier clase de coste en cualquier proceso del ciclo de vida software será siempre bienvenida, pero especialmente en el proceso de mantenimiento. Aunque resulta imposible establecer una estructura de equipo de validez universal, en MANTEMA sí que damos unas directrices generales que debe tener la organización que presta el servicio de mantenimiento.

Debido a la creciente importancia que están adquiriendo las actividades de *outsourcing* de mantenimiento de software [1], entre el conjunto de actividades del proceso de mantenimiento de MANTEMA se encuentran algunas dedicadas al *outsourcing*, y en las cuales se habrá identificado como *responsable* a la organización que presta este servicio.

Con estas consideraciones, en MANTEMA definimos las tres siguientes organizaciones como posibles intervinientes en el proceso de mantenimiento (aunque habrá situaciones en que dos o incluso tres organizaciones de las listadas a continuación coincidan en una sola):

- 1) Cliente, que es la organización que solicita el servicio de mantenimiento de sus software.

- 2) Organización de mantenimiento, que es la organización que suministra el servicio de mantenimiento. Como hemos mencionado, en determinadas situaciones podrá coincidir con la organización del Cliente.
- 3) Usuario, que es la organización que utiliza el software mantenido.

Además, en [17] identificamos una serie de perfiles para cada una de las tres organizaciones arriba mencionadas.

### 3 Estructura detallada de MANTEMA.

En esta sección presentamos el Proceso de Mantenimiento que define la metodología MANTEMA.

#### 3.1 Actividades y tareas iniciales comunes.

Este conjunto de actividades está representado por el nodo inicial del grafo mostrado en la Figura 1. Consta de tres actividades. En la ejecución de las dos primeras (Estudio inicial y Planificación del proceso) se prepara el proceso de mantenimiento, mientras que la tercera está dedicada a la recepción y clasificación de las peticiones de modificación.

Durante la primera actividad (*Estudio inicial*), la organización de mantenimiento recoge información acerca del software que se va a mantener, con el objeto de realizar la planificación del proceso de mantenimiento y, en su caso, de presentar una propuesta de mantenimiento a la organización Cliente. La información recogida durante esta actividad incluye:

- Detalles del entorno hardware (marca y modelo del hardware central, relación de terminales, relación de estaciones de trabajo, tipos de redes, tipos de conexiones con el hardware central)
- Características generales del software (entorno de desarrollo, tipo de sistema de ficheros, tipos de bases de datos)
- Detalles de las aplicaciones (breve descripción, unidad funcional usuaria, aplicaciones relacionadas, nº de pantallas, nº de listados, etc.)
- Aspectos destacables de los procesos de desarrollo y mantenimiento (metodologías y técnicas de desarrollo utilizadas, procedimientos de gestión de calidad, de auditoría, de resolución de problemas, etc.)
- Análisis de riesgos, que se realiza conforme a un completo cuestionario de riesgos elaborado a partir de Eurométodo [5] y la experiencia real.

Una vez que la información anterior está recopilada, las dos siguientes tareas están dedicadas a la preparación de la propuesta de mantenimiento y a la redacción y firma del contrato de prestación del servicio, ambas muy importantes en el caso de que exista una relación de *outsourcing* entre la organización de Mantenimiento y el Cliente.

Concluida la primera actividad, durante la segunda de este nodo inicial se realiza la *Planificación del proceso* de mantenimiento. Durante esta actividad, la organización de mantenimiento adquiere conocimiento de la aplicación. En los casos en que existe *outsourcing*, la organización de mantenimiento actúa durante esta tarea, de habitual-

mente dos meses de duración, simplemente observando cómo ejecuta sus tareas el actual equipo de mantenimiento. Tras este periodo "mudo", la organización de mantenimiento debe entregar al cliente documentación completa del software que incluya informes de auditoría, posibles mejoras, etc., además de ir construyendo documentación de consumo interno que incluya valores de métricas, tablas de referencias cruzadas, diccionarios de datos, etc.

Dentro todavía de la segunda actividad, se definen los procedimientos que deberán seguirse para presentar las peticiones de modificación, se implementa el proceso de gestión de configuración (en caso de que se carezca de uno) y, deseablemente, se preparan los entornos en que se realizarán las pruebas.

A partir del momento en que queda terminada la segunda actividad, la organización de mantenimiento está preparada para ejecutar las acciones necesarias para servir las peticiones de modificación. Entraríamos, por tanto, en un conjunto de actividades y tareas cíclico, en el sentido de que serán ejecutadas para cada Petición de Modificación que se reciba.

Precisamente la tercera y última actividad de este conjunto inicial de actividades y tareas está dedicada a la recepción de la Petición de Modificación y a su clasificación según uno de los dos tipos de mantenimiento que distinguimos en la sección 2 (aunque, para las peticiones de planificable, será necesario seguir precisando si se tratan de correctivo no urgente, perfectivo, preventivo o adaptativo, ya que algunas las formas de actuar serán ligeramente diferentes en cada caso).

### 3.2 Actividades y tareas del mantenimiento no planificable (correctivo urgente).

Detallamos en las dos tablas siguientes la serie de actividades y tareas que deben seguir las intervenciones de mantenimiento que, en la tercera actividad del nodo inicial del proceso, se hayan clasificado como correspondientes a mantenimiento "no planificable" (correctivo urgente).

La documentación de MANTEMA [12] detalla cada uno de los cuatro nodos de la figura 1 con una tabla que tiene exactamente la misma estructura que las mostradas a continuación. Sin embargo, las limitaciones de espacio nos impiden adjuntarlas todas.

Tabla 1. Estructura del mantenimiento no planificable (continúa).

	Análisis del error	Intervención correctiva urgente (continúa)	
	NP1.1 Investigar y analizar causas	NP2.1 Realizar acciones correctivas	NP2.2 Cumplimentar documentación
Entradas	Producto software en explotación con error bloqueante o crítico Petición de modificación	Conjunto de elementos software a corregir	Elementos software antiguos (con errores visibles) Elementos software corregidos
Salidas	Conjunto de elementos software a corregir	Conjunto de elementos software corregidos	Documentación de las acciones correctivas realizadas
Técnicas		Codificación	
Responsable	Equipo de mantenimiento Usuario	Equipo de mantenimiento	Equipo de mantenimiento
Interfaces con otros procesos		Aseguramiento de la calidad Gestión de la configuración	



Tabla 2. Estructura del mantenimiento no planificable (continuación).

	Intervención correctiva urgente (continuación) NP2.3 Ejecutar pruebas unitarias	Cierre intervención NP3.1 Pasar a producción
Entradas	Elementos software corregidos Casos de prueba	Elementos software corregidos y probados
Salidas	Elementos software corregidos y probados Documentación con las pruebas unitarias realizadas	Producto software en explotación corregido
Técnicas	Técnicas de prueba del software	
Responsable	Equipo de mantenimiento	Equipo de mantenimiento Usuario
Interfaces con otros procesos	Aseguramiento de la calidad	Gestión de la configuración

### 3.3 Actividades y tareas finales comunes.

Este conjunto de actividades está representado por el último nodo del grafo de la figura 1 y consta de cinco actividades:

- 1) Registro de la intervención, tras la cual la intervención (incluyendo toda su documentación asociada) queda registrada según los procedimientos que se establecieron en la actividad "Planificación del proceso" del conjunto de actividades y tareas iniciales comunes.
- 2) Actualización de la base de datos histórica, que consiste en almacenar (si no se ha hecho ya) los valores de las diferentes métricas que deben recogerse en cada tarea y que listamos en la sección 5.
- 3) Migración, que será realizada conforme a la actividad de Migración definida en ISO/IEC 12207 cuando el software deba ser cambiado de entorno operativo [9].
- 4) Retirada, que será también realizada conforme a ISO/IEC 12207.
- 5) Fin de la externalización, que ocurrirá si ha existido relación de outsourcing entre la Organización de Mantenimiento y el Cliente.

Como se observa, una gran parte de los elementos que las tareas utilizan como entradas o salidas son documentos que deben ser generados durante el proceso. En nuestra metodología definimos plantillas con los contenidos de todos o casi todos los documentos generables durante el proceso. En la figura 2 mostramos una lista de los más relevantes.

1. Cuestionario inicial	10. Alternativas de implementación
2. Propuesta de mantenimiento	11. Acciones perfectivas realizadas
3. Contrato de mantenimiento	12. Lista de elementos software y propiedades mejorables
4. Tabla de factores de riesgo	13. Acciones preventivas realizadas
5. Resumen técnico	14. Plan de migración
6. Petición de modificación	15. Notificación de futura migración
7. Acciones correctivas realizadas	16. Medidas del producto
8. Pruebas unitarias realizadas	17. Plan de mantenimiento del periodo
9. Diagnóstico y posibles soluciones	

Figura 2. Algunos de los documentos definidos en MANTEMA

#### 4 Métricas.

En MANTEMA definimos una larga serie de métricas que deben ser recogidas con cada intervención de mantenimiento y que se deben incorporar, para su posterior estudio, a una base de datos histórica.

Es interesante recoger métricas tanto de producto como de proceso. Entre las primeras, aparte de las clásicas, más conocidas y más útiles en el proceso de mantenimiento (complejidad ciclomática de McCabe [10], métricas de Halstead [6], etc.), utilizamos también otras para bases de datos que hemos elaborado durante la construcción de la metodología, debido a la creciente importancia que éstas ejercen sobre la complejidad de los sistemas de información de las organizaciones [15].

A continuación, listamos las métricas que deben ser recogidas en cada fase del proceso de mantenimiento definido, lo cual permite mantener puntualmente el control de cada intervención y realizar estimaciones no sólo de esfuerzo, sino también de recursos y su distribución, localización de puntos débiles, cálculo de tendencias, etc.

##### Métricas para las actividades y tareas iniciales comunes (nodo inicial de la figura 1).

- 1) Número de programas
- 2) Complejidad ciclomática de cada programa
- 3) Tamaño de cada programa
- 4) Edad de cada programa
- 5) Legibilidad
- 6) Métricas para bases de datos
- 7) Número de peticiones de modificación recibidas (NPMR)
- 8) Número de MR aceptadas
- 9) Número de MR rechazadas
- 10) Número de MR debidas a cambios legales
- 11) Número de MR debidas a mejoras del proceso

- 12) Número de MR debidas a cambios en los procesos de negocio
- 13) Número de MR por cada tipo de mantenimiento (correctivos urgente y no urgente, perfectivo, preventivo y adaptativo)
- 14) Número de MR por área o dominio funcional
- 15) Numero de MR distribuidas erróneamente
- 16) Número de MR rechazadas por área o dominio funcional

**Métricas para las intervenciones de mantenimiento (nodos centrales de la figura 1).**

- 17) Tiempo de respuesta
- 18) Número medio de anomalías por mes en:
  - Cada aplicación
  - Cada base de datos
  - Todas las aplicaciones que gestionan la misma base de datos
  - Cada equipo de usuarios
- 19) Número y porcentaje de errores cuya responsabilidad recae sobre el equipo de mantenimiento
- 20) Tiempo dedicado a cada tarea de la intervención
- 21) Número de errores detectados en las pruebas unitarias
- 22) Número de errores detectados en las pruebas de integración
- 23) Número de MR replanificadas por periodo, periodo y área, periodo y dominio funcional
- 24) Respeto a la planificación
- 25) Grado de flexibilidad sin replanificar:  
$$\text{Grado de flexibilidad} = \frac{\text{Máximo número de personas} - \text{Mínimo número de personas}}{\text{Número de meses}}$$
- 26) Grado de replanificación (número de horas replanificadas en cada periodo)
- 27) Evolución de la complejidad, tamaño, etc.

**Métricas para las actividades y tareas finales comunes (nodo final de la figura 1).**  
A excepción del tiempo dedicado a cada una de las tareas, no se definen otras métricas para estas actividades.

## **5. MANTOOL: una herramienta para la gestión del proceso.**

La mayoría de las herramientas CASE utilizables durante el proceso de mantenimiento de software son herramientas *verticales*, en el sentido de que automatizan únicamente determinadas tareas del mantenimiento (estimación de costes, pruebas, reestructuración), existiendo pocas herramientas *horizontales*, susceptibles de ser

utilizadas a lo largo de todo el proceso. Esta opinión es corroborada por [14], para quien "la ausencia de un proceso de mantenimiento definido dificulta la utilización de herramientas CASE para la gestión del proceso, y las herramientas disponibles sólo son aplicables a actividades concretas del mantenimiento".

La rigurosa definición que hace MANTEMA del proceso de mantenimiento, estructurándolo a nivel de actividad y tarea, y detallando los diferentes elementos que componen éstas, favorece la traducción de la metodología a un entorno de gestión automático que, tal vez, pueda verse complementado en ciertos momentos por componentes verticales. Con esta idea hemos desarrollado MANTOOL, una herramienta para la gestión automática del proceso de mantenimiento conforme a MANTEMA.

En MANTOOL, cada petición de modificación es clasificada según uno de los tipos de mantenimiento que hemos mencionado. Una vez introducida la información de la petición de modificación, se sigue la evolución de ésta en una pantalla como la que mostramos en la figura 3.

Cada solapa de la ventana anterior representa cada una de las tareas que deben ejecutarse para cada tipo de mantenimiento y, cada elemento de cada solapa, se corresponde con alguno de los objetos que definimos para cada tarea. Mediante el botón "Actualizar componentes" de la figura se accede a uno de los componentes *verticales* que hemos desarrollado, y que realiza diversas medidas sobre el código fuente que compone la aplicación. De este modo se facilita la labor de control de las aplicaciones que se están manteniendo, ya que en cada momento puede conocerse el estado exacto de una aplicación, la incidencia que las diferentes modificaciones han producido sobre sus atributos de calidad, la evolución de éstos a lo largo del tiempo, etc.

The screenshot shows a software window titled "Pantalla de seguimiento e introducción de información de peticiones". At the top, there is a process flow diagram with five steps: "Completar documentación", "Ejecutar pruebas unitarias", and "Pasar a producción". Below the diagram is a form with several sections:

- Información general:** Includes fields for "Investiga y analiza causas", "Tiempo y acciones correctivas", and "Completar documentación".
- Resumen de datos:** Contains fields for "ID", "Tiempo", "PF añadido", "PF modificado", and "PF borrado".
- Métricas:** Contains fields for "Alcance de pruebas unitarias", "LDC añadido", "LDC modificado", and "LDC borrado".

There are also buttons for "Actualizar información" and "Actualizar componentes".

Figura 3. Pantalla de seguimiento e introducción de información de peticiones.

### 6. Conclusiones.

En este artículo hemos presentado MANTEMA, una metodología integral para el mantenimiento de software. MANTEMA ha sido fruto del trabajo conjunto de nuestra universidad y Atos ODS.

MANTEMA constituye una guía de referencia completa para el mantenimiento de software, ya que define de manera muy estricta la secuencia de tareas que deben ser ejecutadas para cada tipo de intervención, además de tener considerados todos los aspectos del proceso de Mantenimiento, incluyendo métricas para el control del proceso.

Merced a la definición rigurosa del proceso hemos construido MANTOOL, que tiene todavía una larga lista de peticiones de perfección. En la figura 4 mostramos una imagen de su módulo *Tutor* que, además, nos sirve para hacer una somera ilustración de la estructura del mantenimiento planificable.

Por último, cabe destacar que estamos trabajando para adaptar la metodología a entornos avanzados (lenguajes de programación orientación a objetos, bases de datos objeto-relacional y 4GL).

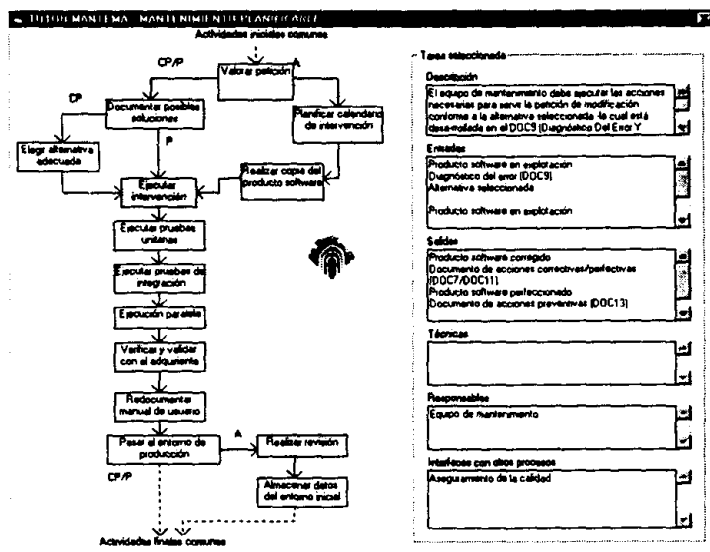


Figura 4. Estructura del mantenimiento planificable.

### 6. Referencias.

- 1 ACM (1996). *Communications of the ACM* (contiene un monográfico dedicado a *outsourcing*), vol. 39, no 7.
- 2 Basili, V., Briand, L., Condon, S., Kim, Y., Melo, W. y Valett, J.D. (1996) *Understanding and predicting the process of software maintenance releases*. Proceedings of the International Conference on Software Engineering, IEEE.

3. Brooks, F. P. (1995) *The Mythical Man-Month*. Essays on Software Engineering Anniversary Edition. Addison-Wesley, USA.
4. Checkland, P. (1981), *Systems Thinking, Systems Practice*. Wiley, Chichester, Reino Unido.
5. Euromethod (1996). *Euromethod ver. 1*. Euromethod project.
6. Halstead, M. (1977) *Elements of software science*. North-Holland.
7. IEEE (1992). *IEEE Std 1219. Standard for Software Maintenance*. IEEE Computer Society Press.
8. IEEE (1995). *IEEE Std 1074: Standard for Developing Software Life Cycle Processes*. IEEE Computer Society Press.
9. ISO/IEC 12207 (1995). Information Technology. Software life cycle processes.
10. McCabe, T. (1976) *A Software Complexity Measure*. IEEE Transactions on Software Engineering, vol. 2, nº 4.
11. McConnell, S. (1996), *Desarrollo y gestión de proyectos informáticos*. McGraw-Hill, España.
12. MMET-1.2 (1999). *Proyecto MANTEMA: documento MMET-1.2 (informe interno)*. Universidad de Castilla-La Mancha y Atos ODS, S.A.
13. Piattini, M.G., Ruiz, F., Polo, M., Villalba, J., Batanchury, T. y Martínez, M.A. (1998), *Mantenimiento del software: conceptos, métodos, herramientas y outsourcing*. Ed. RAMA. Madrid, Spain.
14. Pigoski, T. M. (1996), *Practical Software Maintenance. Best Practices for Managing Your Investment*. Ed. John Wiley & Sons. Estados Unidos.
15. Polo, M., Calero, C, Ruiz, F. y Piattini, M. (1998), *Métricas de calidad y complejidad para bases de datos*. Actas de las III Jornadas de Ingeniería del Software. Murcia.
16. Polo, M., Piattini, M.G., Ruiz, F. y Calero, C. (1999a), *MANTEMA: a Complete Rigorous Methodology for Supporting Maintenance based on The ISO/IEC 12207 Standard*. Proceedings of the Third European Conference on Software Maintenance and Reengineering. IEEE, Amsterdam (Holanda).
17. Polo, M., Piattini, M., Ruiz, F. y Calero, C. (1999b). *Roles in the maintenance process*. Enviado a Software Engineering Notes.
18. Pressman, R.S. (1993), *Ingeniería del software: un enfoque práctico*. McGraw-Hill.