

Tenth Mexican International Conference on

Computer Science

21

25

SEPT

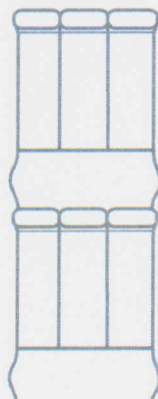
2009

MEXICO

CITY



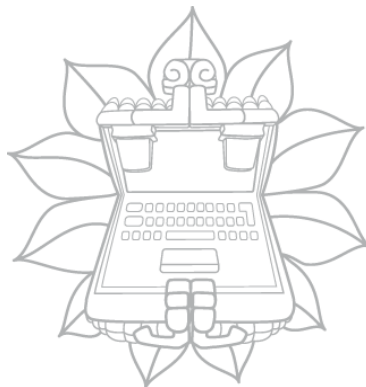
dia.ccm.itesm.mx/enc2009



Advances in Computer Science

*Joint Workshops of the Mexican International
Conference on Computer Science*

2009



Preface

ENC 2009 features a number of co-located workshops related to emergent topics in Computer Science, and held during the main conference. The aim of these workshops is to identify and integrate scientific communities interested in emerging areas in Computer Science, the Mexican Society in Computer Science (*Sociedad Mexicana de Ciencia de la Computación*) organizes the sixth meeting of joint workshops of the Mexican International Conference in Computer Science (ENC 2009): Advances in Computer Science.

This year, workshop proposals were evaluated by a scientific committee composed by members who are aware of new research and industrial trends in our country. The quality of the organizing group, the originality of the topics and the impact on the scientific committee were the most important criteria taken into consideration for accepting workshops.

In 2009, the accepted workshops represent the groups that are constantly working on the following topics:

- [Symposium on Software Engineering](#)
- [Symposium on Mobile Systems and Enterprise Computing](#)
- [Symposium on Clinical and Medical Computing](#)
- [PhD Workshop](#)

Advances in computer science is a meeting that supports communities with common interests to discuss and present advances in emerging topics and also to let other communities consolidate their scientific and technological impact in Mexico.

ENC 2009 Workshop Coordinators

Francisco Acosta

francisco.acosta@ujat.mx

Universidad Juárez Autónoma de Tabasco

Mexico

Rafael Lozano Espinosa

ralozano@itesm.mx

Tec. de Monterrey CCM

Mexico

Symposium on Software Engineering

The objective of the symposium is to integrate the academic community to discuss novel aspects on research and education related to software engineering; define the issues to be considered for defining education programs on the topic and recommend initiatives like the National programme of software engineering sponsored by the Mexican Economy Council.

Organizers

Blanca Gil Castellanos

Tec. de Monterrey CCM Mexico

José Reyes Juárez Ramírez

Universidad Autónoma de Baja California Mexico

Hanna Oktaba

Universidad Nacional Autónoma de México Mexico

Program Committee

Blanca Gil Castellanos

Tec. de Monterrey CCM Mexico

Aurora Vizcaino Barceló

Universidad de Castilla-La Mancha Spain

Guillermo Licea Sandoval

Universidad Autónoma de Baja California Mexico

Hanna Oktaba

Universidad Nacional Autónoma de México Mexico

José Reyes Juárez Ramírez

Universidad Autónoma de Baja California Mexico

Ma. Guadalupe Ibarguengoitia González

Universidad Nacional Autónoma de México Mexico

Accepted papers

Title Especialización de MoProSoft basada en el Método Ágil Scrum (MPS-Scrum)

Author(s) Magdalena Dávila Muñoz and Hanna Oktaba

Abstract Este artículo describe el proceso llamado MPS-Scrum resultado de la combinación de MoProSoft y Scrum. El propósito de MPS-Scrum es acelerar el desarrollo de software incorporando prácticas ágiles de Scrum en procesos de desarrollo de software establecidos bajo MoProSoft. MPS-Scrum describe una serie de prácticas tomadas de Scrum con roles y productos de trabajo de MoProSoft. MoProSoft es un modelo mexicano que reúne una serie de prácticas de gestión y de ingeniería para el desarrollo de software para guiar a las organizaciones en la mejora de sus

Ayoyando la Mejora de Procesos de Software a traves de Plantillas

Miguel Morales Trujillo,
Guadalupe E. Ibargengoita
*Posgrado en Ciencia e
Ingeniera de la Computacion,
Universidad Nacional
Autonoma de Mexico*
{migmor, gig}@
ciencias.unam.mx

Francisco J. Pino
*Grupo IDIS
Facultad de Ingeniera en
Electronica y
Telecomunicaciones,
Universidad del Cauca*
ffpino@unicauca.edu.co

Mario Piattini
*Grupo ALARCOS
Escuela Superior de
Informatica,
Universidad Castilla-La
Mancha*
Mario.Piattini@uclm.es

Resumen

Un obstaculo importante para una VSE al decidir implantar un modelo de mejora de procesos de software, es elegir el marco metodologico y despues enfrentarse al dilema de guiarlo y ajustarlo a sus necesidades especificas. Consideramos que el modelo de referencia puede ser adaptado a las necesidades de la VSE de una manera mas facil mediante la utilizacion de plantillas, las cuales pueden permitir la integracion adecuada de las actividades propuestas en el modelo y aquellas que la empresa ya vena realizando, resultando en la disminucion de la brecha entre la definicion de los modelos y su adopcion. Este documento presenta una alternativa centrada en plantillas de productos, basadas en tecnicas descritas en RUP y PMBoK, que pretenden facilitar la adopcion de los procesos de los marcos metodologicos para VSEs, definidos en COMPETISOFT y la futura norma ISO/IEC 29110-5-1. El objetivo es intentar incorporar las mejoras en los procesos, mediante el uso de plantillas de productos, que sean lo mas manejables, faciles y entendibles posible, tomando en cuenta las limitantes intrinsecas de las pequenas empresas.

Palabras clave: COMPETISOFT, RUP, PMBoK, SPI, VSE, plantilla, unidad atomica.

1. Introduccion

La mejora de procesos de software (SPI por sus siglas en ingles), es un esfuerzo controlado y guiado que tiene como objetivo mejorar el conjunto de procesos de una empresa desarrolladora de software, en particular su capacidad y su madurez. Toda iniciativa para implantar un SPI en una organizacion

debe ser guiada por un marco metodologico que presente las actividades necesarias para obtener esa mejora dentro de la organizacion.

Las organizaciones que se catalogan como pequenas empresas (VSEs por sus siglas en ingles), tienen menos de 25 empleados, son fundamentales para muchas economas, en pases como Estados Unidos, Brasil, Canada, China, India, Finlandia, Irlanda y Hungra las VSEs representan cerca del 85% de todas las organizaciones dedicadas al desarrollo de software [1] [2].

Para que las VSEs obtengan una mejora continua, es necesario que adopten modelos de referencia de mejora de procesos. Sin embargo es muy frecuente que las VSEs consideren que la implantacion de estos modelos es demasiado costosa, requiere demasiado tiempo y que sobretodo se piensa que dichos modelos estan dirigidos a organizaciones grandes, lo que resulta un obstaculo para comenzar la mejora en la VSEs [6] [7].

Actualmente, en un esfuerzo para que las VSEs eviten este obstaculo y se inicien en un proceso de mejora, se han desarrollado modelos de referencia de procesos dirigidos especificamente a ellas como son COMPETISOFT [3] [4] e ISO/IEC PDTR 29110 [5]. Estos modelos simplifican las actividades requeridas para llevar a cabo la mejora. Sin embargo estos modelos en general no describen como realizar dichas actividades, dejando a la organizacion a un paso de concretar el SPI.

El paso de plasmar los resultados obtenidos al ejecutar las actividades descritas no es trivial. En este punto la organizacion enfrenta un inconveniente de diferente ndole, la presencia de la problematica de definir como representar los resultados de las actividades en un producto tangible. Una estrategia para guiar la implementacion de procesos de software

según los modelos de referencia de procesos [13], dentro de una organización es el uso de plantillas para los productos que faciliten la ejecución de las actividades. Como un apoyo a esta estrategia, este trabajo presenta la propuesta de una guía basada en plantillas para apoyar la implantación del marco metodológico de COMPETISOFT como modelo de procesos para la organización.

El uso de las plantillas ayuda a reducir el desconocimiento que tienen las organizaciones de los modelos, ya que representan los resultados de las actividades que deben realizar, disminuyendo la necesidad de recurrir a otro material de apoyo o fuente de conocimiento. Es decir, la plantilla actuará como guía explícita y recipiente de los resultados de cada actividad, siendo ésta un producto que cumple con los estándares requeridos.

Este documento está organizado de la siguiente forma, en la sección 2 se presentan los antecedentes, en la sección 3 se muestran las plantillas propuestas y sus características, en la parte 4 dos ejemplos de su uso y en la 5 las conclusiones y el trabajo futuro.

2. Antecedentes

Existen algunos modelos de referencia disponibles para VSEs, como son COMPETISOFT o ISO/IEC PDTR 29110, en estos modelos se definen grupos de procesos, y éstos a su vez agrupan actividades que tienen que realizar las organizaciones para conseguir una mejora. Sin embargo estos modelos se limitan a describir las actividades de cada grupo de procesos y los productos que se generan con su realización. Nuestra propuesta es encuadrar al marco metodológico de COMPETISOFT en un desarrollo que se base en plantillas, en las que se plasmarán los resultados de las actividades para generar productos estandarizados que cumpla con los requerimientos adecuados para incrementar el nivel de madurez de la organización y su capacidad de procesos.

COMPETISOFT surge con el objetivo de incrementar el nivel de competitividad de las empresas pequeñas de software Iberoamericanas, difundiendo un marco común que pudiese ajustarse a las necesidades específicas de cada una de ellas, con el fin de establecer un mecanismo de evaluación y certificación dirigido a las VSEs.

Buscando que el modelo sea el que se adapte a la organización, en COMPETISOFT se integran de manera consistente elementos básicos e indispensables para llevar a cabo una implantación adecuada del marco metodológico. El modelo está dividido en tres capas, Alta Dirección, Gerencia y Operación, esta estructura acumula los principios básicos que

componen a una VSE. Por una parte la Alta Dirección es la encargada de la planificación estratégica y la revisión continua del todo en la organización; la Gerencia es la encargada de proveer los recursos, procesos y proyectos, además de vigilar el cumplimiento de los objetivos de la organización; y como tercer bloque se tiene la capa de Operación, donde se concentran los procesos de *Desarrollo y Mantenimiento de Software* así como la *Administración del Proyecto*.

La capa de *Operación* busca que cada uno de los proyectos de desarrollo de software que tiene la organización cumpla con los compromisos establecidos en tiempo, costo y calidad [17]. Los procesos que establecen las prácticas requeridas son justamente la *Administración del Proyecto*, para terminar en tiempo y costo, el *Desarrollo de Software* para cumplir con los requisitos y eliminar defectos en fases tempranas a través de las verificaciones y validaciones y el *Mantenimiento de Software* que debe realizarse después de que se ha liberado el producto. Debido a la importancia crítica que tiene la capa de Operación para la organización, este trabajo se concentra en esos grupos de procesos para la realización de la SPI basada en plantillas.

Teniendo definido el alcance de las plantillas se debió establecer una metodología para regir su estructura y desarrollo. En este caso las plantillas están basadas en el Proceso Unificado de Rational (RUP por sus siglas en inglés) [8] [9] y el Cuerpo de Conocimiento de la Administración de Proyectos (PMBOK por sus siglas en inglés) [10], ya que son estándares aplicables a cualquier desarrollo de software.

La elección de RUP como regidor de las plantillas se debe en gran parte a que, al igual que COMPETISOFT, propone un desarrollo iterativo e incremental, es altamente adaptable a las necesidades específicas de cada organización y motiva el cumplimiento de los criterios de calidad. Agregado a esto, la aceptación que tiene como estándar entre la industria lo hace una excelente alternativa para los objetivos de las plantillas.

Por su parte, PMBoK, es un estándar definido para la Administración de Proyectos, dividido en nueve áreas del conocimiento, en donde se engloban las mejores prácticas y se estandariza la información referente a la gestión de proyectos de manera consistente y aplicando ese conocimiento a la ingeniería de software.

En la siguiente sección se presentarán las características y el desarrollo de las plantillas, que pretenden ser una guía explícita de cómo concretar la realización de las actividades en una mejora de proceso para una organización.

3. Las Plantillas y sus características

La utilización de plantillas como guías explícitas de la mejora de procesos espera lograr un aporte a la organización, al ser una herramienta [18] [19] que disminuirá la brecha entre la organización y el uso de los modelos de referencia. Para desarrollar las plantillas se comenzó con los procesos de *Administración del Proyecto*, posteriormente con los de *Desarrollo y Mantenimiento*. Para cada uno de los procesos establecidos se identificaron aquellas actividades que generaban un producto y aquellas que no lo hacían.

Para aquellas actividades que tienen un producto asociado, la plantilla respectiva ofrece una técnica para la realización de la actividad. Por ejemplo, para la captura de requisitos se propone su representación mediante el *Diagrama de casos de uso*, para la creación del prototipo de interfaz de usuario se propone un *taller de requerimientos* y para el plan de pruebas de integración se propone hacerlo por medio de un *grafo de dependencias*.

Para llevar a buen término las actividades propuestas en cada proceso y solventar sus objetivos, se optó por recopilarlas del PMBoK, para la *Administración del Proyecto*, y de RUP para el *Desarrollo y Mantenimiento de Software*. En la *Administración del Proyecto* [11] presenta un mapeo conjunto entre RUP y el PMBoK, en el que se concluye que no hay contradicciones entre los dos estándares por lo que las prácticas del RUP y las de PMBoK pueden coexistir sin anomalías. Lo que nos permite establecer una traza entre PMBoK, RUP y COMPETISOFT.

Por otro lado, las principales razones por las que se eligió a RUP en la parte de *Desarrollo y Mantenimiento de Software* se deben a que estos procesos están dirigidos por casos de uso y centrados en la arquitectura. Además de acuerdo al estudio realizado en [12] se concluye que el nivel de cobertura entre estos dos procesos es altamente alcanzado de acuerdo a la calificación de ISO/IEC 15504 [15]. La norma ISO/IEC 15504 provee un modelo para la evaluación de procesos. Mediante un conjunto de requerimientos mínimos que deben cumplir los procesos se determina el nivel de cobertura.

Sin embargo, se encontró que existían productos propuestos en COMPETISOFT que RUP no describe, por ejemplo el *Reporte de Pruebas de Integración* para el que COMPETISOFT destina una fase completa del *Desarrollo*, por lo que en casos similares se optó por utilizar otros estándares de la industria como base para adaptarlos a los objetivos de este trabajo.

Entre las principales características de las plantillas se puede observar la modularidad con la que cuentan las entidades que componen su estructura. Se pueden diferenciar dos tipos, las entidades *generales*, que son comunes a todas las plantillas y están enfocadas a describir aspectos como la *Introducción*, *Descripción general* del documento, *Justificación de decisiones* e *Información de apoyo*. Y por otro lado tenemos a las entidades del *proceso* que son las que contendrán todo aquello que generen las actividades descritas en el modelo de referencia.

Cada entidad es tratada como una *unidad atómica* de la plantilla, y cada *unidad atómica* contiene una descripción de la técnica propuesta para generar su contenido. Con esto se intenta proveer una guía explícita a través del proceso, es decir, describir con más detalle la manera de generar el producto para que cumpla con lo requerido por las actividades del proceso. En el Apéndice se presenta una plantilla completa. La descomposición en *unidades atómicas* facilita la trazabilidad de los elementos a lo largo del ciclo de vida del software. Ejemplo de esto es la traza entre los requisitos solicitados por el cliente, los casos de uso y las clases del análisis que pretenden cubrir estos requisitos, siendo rastreables estas correspondencias en cualquier instante del desarrollo.

La elección de los componentes de cada entidad del proceso estuvo dirigida por los requerimientos de COMPETISOFT siempre vigilando que su diligencia fuese sencilla y autocontenido en la plantilla. Ejemplo de esto es, que en la fase de pruebas, la *unidad atómica Estrategia de las pruebas* está destinada a realizar pruebas no contempladas en la metodología pero si requeridas para la organización. Como complemento se provee a cada plantilla de un *Diagrama de unidades atómicas* que muestra la forma de uso de cada plantilla.

Entre las principales características de las plantillas está la modularidad con la que cuentan las entidades del proceso que componen su estructura que permiten adecuarlas a varios modelos de referencia.

3.1 Plantillas de Administración del Proyecto

El primer subgrupo de procesos concierne a la *Administración del Proyecto* (AP), y es el encargado de establecer y llevar a cabo sistemáticamente las actividades que permitan cumplir con los objetivos del proyecto en tiempo y costo.

Las actividades de AP, generaron un total de cuatro plantillas, tres de ellas son productos que se generan varias veces a lo largo del proceso: la *Matriz de Trazabilidad*, el *Documento de Aceptación* y el *Reporte de Seguimiento*. Por esta razón se presenta una estandarización de esos documentos en la que se

incluyen los elementos específicos de cada actividad que, de acuerdo a PMBoK se consideran buenas prácticas. La cuarta plantilla, *Plan del Proyecto*, describe la información relativa al proyecto en el que se incluyen y documentan técnicas de diferentes áreas del conocimiento del PMBoK. El llenado de las plantillas de AP, buscó ser lo más intuitivo posible, debido a que son productos que están presentes varias veces a lo largo del proceso. La Figura 1 muestra la *unidad atómica Estrategia de control de versiones*, perteneciente a la plantilla *Plan del Proyecto*, en la que se define, utilizando técnicas del PMBoK, la manera en que se gestionarán los cambios durante el desarrollo, manteniendo la integridad de las líneas base, y la *Forma de solicitud de cambio*, necesaria para conservar un registro de cambio del proyecto.



AP Plan del proyecto

16. Estrategia de control de versiones

[Método para administrar la configuración del sistema, se debe incluir un mecanismo para solicitar cambios así como la estructura y reglas de uso del repositorio.]

Estrategia del control de versiones															
Identificación de la configuración	[establecer lo que será almacenado en el repositorio, suministrar la línea base para clasificar productos y documentos]														
Estado de configuración	[información de los productos que están en el repositorio]														
Auditor	[establecer quién verificará que se ha cumplido con los requerimientos funcionales]														
Frecuencia de auditorías	[establecer el plazo en que se efectuarán las auditorías]														
Cambios solicitados	<table border="1"> <thead> <tr> <th>Nombre del producto</th> <th>Descripción</th> <th>Impacto</th> <th>Solicitante</th> <th>Estado</th> </tr> </thead> <tbody> <tr> <td> </td> <td> </td> <td> </td> <td> </td> <td> </td> </tr> </tbody> </table>					Nombre del producto	Descripción	Impacto	Solicitante	Estado					
	Nombre del producto	Descripción	Impacto	Solicitante	Estado										
[listado de los cambios solicitados a lo largo del proceso]															
Políticas	[acuerdos para el uso del repositorio, establecer quién pueda almacenar, modificar y acceder al repositorio]														

Forma de Solicitud de Cambio	
Proyecto:	Fecha:
Información del elemento	
Nombre:	
Dirección de respaldo:	
Información del cambio	

Figura 1. Segmento de la plantilla AP Plan del Proyecto.

3.2 Plantillas de Desarrollo de Software

Los productos requeridos por el grupo de procesos de *Desarrollo de Software* (DSW), se engloban en una plantilla por cada fase propuesta por COMPETISOFT. Por lo que se obtienen siete elementos, comenzando por la *Especificación de Requisitos* en la cuál se incluyen las plantillas específicas para modelar los requerimientos mediante la técnica de casos de uso,

siendo la parte más importante del documento. A partir de la captura de cada requisito se comienza con la trazabilidad a lo largo del proceso. Dentro del mismo documento se deben incluir el *Plan de prueba* y el *Prototipo de la interfaz de usuario* para cubrir cada caso de uso requerido.

También se desarrollaron las plantillas de *Análisis, Diseño, Construcción, Integración, Pruebas y Cierre*. Como ejemplo de éstas plantillas en la Figura 2, se observan los elementos que componen a la *Vista Lógica* (punto 4.), en donde se debe ofrecer una descripción (4.1.) de la vista así como la definición de la arquitectura (4.2.) por medio de paquetes. La *Vista de Despliegue* (6.), es donde se propone su presentación por medio de un diagrama de despliegue (6.2.). Por último la *Vista de Datos* (7.), incluye las indicaciones para definir los aspectos relevantes de la persistencia de los datos (7.1.) y su representación mediante un diagrama (7.1.1.). Todo lo anterior está basado y adaptado a partir de las técnicas propuestas por RUP en la Fase de Diseño.



DSW Diseño

4. Vista Lógica

4.1 Descripción

[Esta sección describe de manera general la descomposición del modelo mediante capas o jerarquía de paquetes.]

4.2 Paquetes de la arquitectura

[Por cada paquete se debe incluir un nombre, una breve descripción de éste y un diagrama con las clases más importantes que contiene.]

4.2.1 <Nombre del Paquete 1>

4.2.1.1 Descripción breve del paquete

[La descripción debe expresar brevemente el propósito del paquete.]

4.3 Clases del Diseño

[Para cada clase del paquete se debe incluir un detallado de las clases identificadas, una descripción de las responsabilidades de la clase así como sus atributos y operaciones.]

4.3.1 <Diagrama de clases detallado>

5. Vista del Proceso

5.1 Descripción

[Esta sección describe la descomposición del sistema en procesos y grupos de procesos. Los organiza de acuerdo a las relaciones de interacción que existen entre ellos. Describe los principales

Figura 2. Segmento de la plantilla DSW Diseño.

Cabe resaltar que las *unidades atómicas* de la plantilla pueden ser reordenadas en subconjuntos para satisfacer las necesidades de otros modelos.

3.3 Plantillas de Mantenimiento de Software

El *Mantenimiento de Software* (MSW) propuesto en COMPETISOFT es un proceso ágil [16]. Para este proceso se abstraieron dos elementos necesarios para llevarlo a cabo, la plantilla de *Petición de Modificación* en la que se solicita la participación del equipo de mantenimiento, y la plantilla de *Intervención*, en la

cual se describe en detalle lo necesario para cumplir con la petición de modificación.

La estructura de estas plantillas, al igual que las plantillas de AP, pretende que sean de fácil diligencia, debido a que, por ejemplo, la *Petición de Modificación* puede ser diligenciada por el usuario del producto de software y no por un miembro de la organización desarrolladora (ver Figura 3). En esta figura se observa que cada elemento de esta plantilla busca ser fácil de abstraer y que la información que refleje sea de utilidad para el equipo de mantenimiento. Las guías explícitas de cada *unidad atómica* son preguntas directas, al poseedor de la plantilla, cuya respuesta debe ser a ser inmediata, clara y suficiente.



MSW Petición de modificación

Petición de modificación

Prioridad	[Normal / Urgente]
Descripción de el error	[En este apartado se deberán responder las preguntas del estilo: ¿A quién se le presentó el error? ¿Cuándo sucedió el error? ¿Cómo ocurrió el error? Dichas preguntas deberán ser respondidas por el usuario del sistema a quién se le presentó el error. El objetivo de esta sección es establecer las acciones que dispararon el error.]
Entorno de ocurrencia	[En este apartado se debe incluir un listado de las características presentes en el entorno del sistema en el momento en que se presentó el error.]
Aspectos afectados	[En esta sección se deberán listar los aspectos que se ven afectados por el defecto descrito o la ocurrencia del error.]
Solicitado por	[nombre de la persona que realiza la petición, en caso de ser distinta al usuario al que le ocurrió será necesario incluir el nombre de éste]
Diagnóstico y posibles soluciones	[En esta sección se deben proponer posibles soluciones al error encontrado, así como el diagnóstico del error, es decir la naturaleza del error mediante la observación de las señales o causas de éste.]
Información de apoyo	[La información de apoyo complementa aspectos no incluidos en esta plantilla y que se considera oportuno incluir.]

Figura 3. Plantilla MSW Petición de Modificación.

3.4 Diagrama de Unidades Atómicas

Para mostrar de una manera gráfica la aplicación de las plantillas durante la ejecución del proceso, se procedió a realizar diagramas en los que se representan tres elementos: roles involucrados, listado de actividades descritas en el modelo y *unidades atómicas* de la plantilla que son utilizados. En la Figura 4 presentamos el *diagrama de unidades atómicas* asociado a la plantilla del *Plan del Proyecto*. En la parte izquierda de la figura, se muestra el grafo de las actividades descritas en el proceso de AP del modelo de referencia de procesos de COMPETISOFT (secuencia A. 2.). Por medio de carriles se diferencian los roles involucrados (columnas) en cada una de las actividades. Dentro de cada carril se señala la *unidad atómica* de la plantilla mediante la cual se refleja la

realización de la tarea o actividad del proceso correspondiente. Para finalmente concretar un mapeo claro y consistente con el fin de cubrir las necesidades específicas de las actividades según el modelo de referencia de procesos y las plantillas.

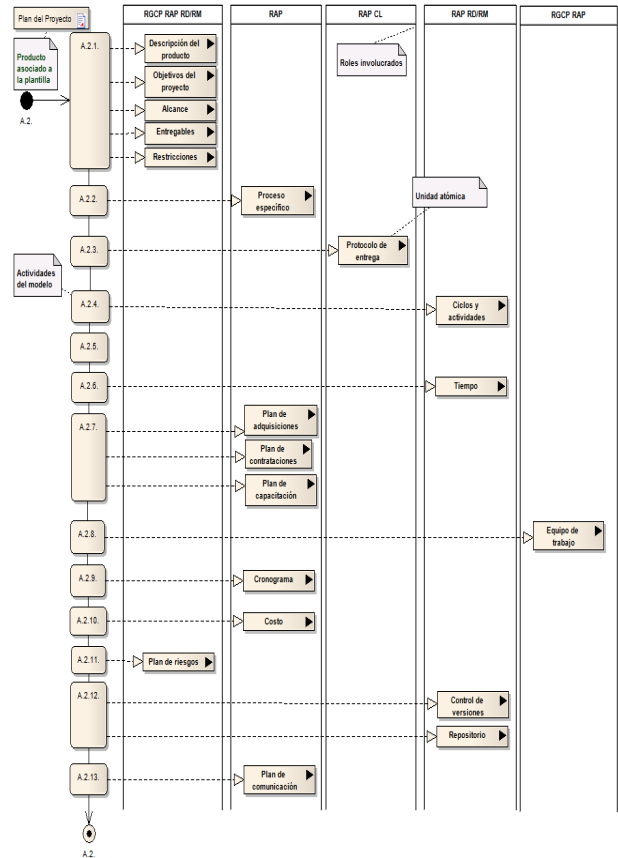


Figura 4. Diagrama de unidades atómicas.

3.5 Adaptación de las plantillas a ISO/IEC 29110-5-1

Aprovechando la modularidad de las plantillas, se eligió un conjunto de *unidades atómicas* para adaptarlas a la ISO/IEC 29110-5-1, con el fin de construir algunas plantillas, las cuáles son parte de cada uno de los *paquetes de despliegue* [14] que propone esta futura norma.

En la Figura 5, se muestra el contenido resultante de conjuntar las *unidades atómicas* adecuadas para completar el *paquete de despliegue de Software Requirements Analysis (Análisis de Requisitos del Software)* de ISO/IEC 29110-5-1. La facilidad con la que se ha podido adaptar a este *paquete de despliegue* es un ejemplo de las virtudes de la definición de las plantillas a través de *unidades atómicas*. Esta modularidad es un aspecto que busca hacer flexible la implantación de los procesos en la empresa.



Index

1. Introduction	2
1.1 Purpose	2
1.2 Scope	2
1.3 Definitions, Acronyms and Abbreviations	2
1.4 References	2
2. Overall Description	2
3. Non-functional Requirements	2
3.1 Usability	3
3.1.1 <Usability requirement 1>	3
3.2 Reliability	3
3.2.1 <Reliability Requirement One>	3
3.3 Performance	3
3.3.1 <Performance Requirement One>	3
3.4 Supportability	3
3.4.1 <Supportability Requirement One>	4
3.5 Portability	4
3.6 Design Constraints	4
3.6.1 <Design Constraint One>	4
3.7 Reusability	4
3.8 Interfaces	4
3.8.1 User Interfaces	4
3.8.2 Hardware Interfaces	4
3.8.3 Software Interfaces	4
3.8.4 Communications Interfaces	4
3.9 Licensing Requirements	4
3.10 Legal, Copyright, and Other Notices	4
4. Functional Requirements	5

Figura 5. Contenido resultante de la plantilla del Paquete de Despliegue *Software Requirements Analysis*.

4. Ejemplos de uso

Las plantillas propuestas para los procesos de AP y DSW, han sido utilizadas en el desarrollo de dos productos de software inmersos en un entorno académico. El primer ejemplo se aplicó a un proyecto de desarrollo de una herramienta llamada HEPALE! (Herramienta Educativa Para la Adopción de Los Estándares) compuesta por dos módulos, que en conjunto, integran estándares propios de la educación a distancia para difundir los modelos COMPETISOFT e ISO/IEC 29110. Además recaba información estadística útil que puede ser analizada para proporcionar medios con el fin de mejorar la calidad de la oferta educativa y contribuir con la difusión de los modelos de referencia a nivel internacional.

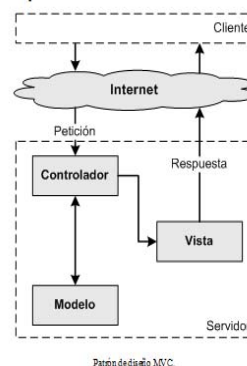
El segundo sistema fue la creación de un repositorio que cuenta con los cambios necesarios en su estructura para la manipulación de Objetos de Aprendizaje y E-learning. El repositorio es construido bajo una arquitectura orientada a servicios implementando servicios web para la manipulación, búsqueda y almacenamiento de *paquetes de despliegue* utilizando estándares para el manejo de seguridad, búsqueda y almacenamiento. El repositorio facilita el intercambio de contenidos entre diferentes Sistemas de

Administración de Aprendizaje además de facilitar la publicación de estos.

A continuación a manera de ejemplo, se muestra el documento resultante de utilizar la plantilla de la fase de Análisis en el desarrollo de HEPALE! En la Figura 6, se muestra el resultado de aplicar las técnicas propuestas para la *unidad atómica de Representación de la Arquitectura*, donde se presenta el diagrama, metas y restricciones impuestas a la arquitectura del sistema.



2. Representación de la Arquitectura



2.1 Metas de la arquitectura

La arquitectura deberá ser modular, para poder separar cada una de las características del modelo, en este sentido se debe de proveer una arquitectura que separe la lógica del sistema, de la capa de presentación y el acceso a datos.

Esto brindará al sistema una claridad para el flujo de los datos y al mismo tiempo para su mantenimiento y mejora, en este sentido se utilizará el patrón de diseño arquitectónico Modelo Vista Controlador (MVC), este patrón separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El patrón MVC es ampliamente utilizado en aplicaciones para la web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página; el modelo es el Sistema de Gestión de Base de Datos y la Lógica de negocio; y el controlador

Figura 6. Ejemplo de uso de la Representación de la Arquitectura.

5. Conclusiones y trabajo futuro

Al menos tres cuartas partes de las empresas de desarrollo de software entran en la categoría de VSE y tienen 25 o menos empleados, por lo que es necesario fomentar la aplicación de modelos de mejora que se adapten a ellas y no a la inversa. Tomando en cuenta lo anterior, se propuso una guía basada en plantillas para apoyar la adopción de los modelos con el fin de proveer una mejora tomando en cuenta la realidad y las necesidades de este tipo de organizaciones. En este trabajo se muestran las plantillas que componen esta guía para implantar COMPETISOFT. Dicha guía tiene la característica de ser flexible debido a la estructura modular de las plantillas, por lo que resulta fácilmente adaptable a las necesidades de la organización.

Por otro lado, la aportación del uso de entidades del proceso en *unidades atómicas* hace posible su

utilización y adaptación a otros modelos de referencia, como a la ISO/IEC CD TR 29110-5-1.

Como aspectos positivos descritos por las personas que usaron las plantillas en los ejemplos de uso, podemos listar:

- Las plantillas facilitan y muestran el camino a seguir al realizar las actividades de un proceso según un modelo de referencia.
- Las plantillas integran los aspectos necesarios y solicitados por el modelo de referencia, por lo que no es posible pasar por alto ninguno.
- En las plantillas se puede concentrar todo lo que resulta de realizar las actividades del modelo, por lo que al final se obtiene la documentación integral, completa y estandarizada sin incongruencias ni contradicciones.
- El uso de *unidades atómicas* facilita la trazabilidad de los elementos, además provee de una flexibilidad notable a la plantilla, ya que la decisión de usar o no cada *unidad atómica* incrementa su utilidad para el desarrollador.
- La manera en que están formuladas las *unidades atómicas*, en especial de AP y MSW, acorta el tiempo y simplifica su diligencia, además facilita su aplicación a varios modelos de referencia.

Como trabajo futuro se identificaron los siguientes puntos:

- Integrar a las plantillas, mediante EPF Composer, a todo el trabajo previo que existe sobre COMPETISOFT, para hacerlas accesibles y disponibles para usarse por las empresas que lo deseen.
- Obtener un mayor número de aplicaciones de las plantillas para evaluar, refinar y mejorar su contenido.
- Iniciar una aplicación industrial de las plantillas.

Agradecimientos

Este trabajo fue posible gracias al apoyo de ARMONIAS (PII2109-0223-7948) e INGENIO (PAC08-0154-9262) proyectos de la Junta de Comunidades de Castilla-La Mancha, Consejería de Educación y Ciencia y por el Proyecto Institucional de Apoyo a los Estudios de Posgrado PAEP y por el programa de Becas Mixtas otorgada por el CONACyT.

Referencias

[1] Qualidade no Setor de Software Brasileiro-Pesquisa 2001, Quality in the Brazilian Software Sector-Survey 2001, Ministry of Science and Technology (MTC), 2001;

<http://ctjovem.mct.gov.br/index.php/content/view/34854.htm> (Texto en portugués) 15/01/2009.

[2] Software Industry Statics for 1991-2005, Enterprise Ireland, 2006; www.nsd.ie/htm/ssii/stat.htm, 15/01/2009.

[3] Oktaba, H.; García, F., Piattini, M., Pino, F., Alquicira, C., Ruiz, F. "Software Process Improvement: The COMPETISOFT Project". *IEEE Computer*, October, 2007. Vol. 40(10), pp. 21-28.

[4]COMPETISOFT,

<http://alarcos.infr.uclm.es/Competisoft/>,15/01/2009.

[5] ISO/IEC CD TR 29110-5-1 Software Engineering - Lifecycle Profiles for Very Small Entities (VSE) - Part 5-1: Management and Engineering Guide - Basic Profile, http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=51160, 15/01/2009.

[6] Richardson Ita, Gresse von Wangenheim Christiane, "Why Are Small Software Organizations Different?", *IEEE Software*, January/February 2007. Vol. (40). pp.18-20.

[7] Laporte, C. Y., Alexandre, S., O'Connor, R., "A software engineering lifecycle standard for very small enterprises", in R. V. O'Connor et al. (Eds.): EuroSPI 2008, *CCIS* 16, pp. 129-141

[8] RUP, <http://www-01.ibm.com/software/awdtools/rup/>,1/05/2009

[9] Jacobson I., Booch G., Rumbaugh J., *El Proceso Unificado de Desarrollo de Software*, Pearson, 2000

[10] PMBoK, <http://pmbook.ce.cmu.edu/>,1/05/2009

[11] Charbonneau, S. "A mapping between RUP and the PMBoK",

<http://www.ibm.com/developerworks/rational/library/4721.html#author>, 01/05/2009

[12] Cánepa K., Dávila, A., "Mapeo de los Procesos de RUP respecto a MoProSoft", *JISIC* 08, pp. 139-146

[13] Kong, B., Luo, X., Jiang, Y., Mao, M. "Template-based software process improvement", *Journal of Information and Computational Science* 1 (2), pp. 175-180

[14] Claude Y. Laporte; Simon Alexandre, Alain Renault, "Developing International Standards for Very Small Entities", *IEEE Computer* 41(3): 98-101 (2008).

[15] van Loon, H. (2007a) "Process Assessment and ISO 15504", *Springer*.

[16] Pino, F., F., Triñanes, G., García, F. y Piattini, M. "Agil_MANTEMA: Una metodología de mantenimiento de software para pequeñas organizaciones". *JISBD* 08, pp.171-182

[17] Oktaba, H., Piattini, M., Pino, F., Orozco, M. Julia., Alquicira, C., *COMPETISOFT: Mejora de Procesos de Software para Pequeñas y Medianas Empresas y Proyectos*, Ra-Ma, 2008

[18] Word Template Index, <http://homepages.fh-giessen.de/~hg14540/SQM/projekt/rup/wordtmpl/index.htm>

[19] Rational Unified Process: Artifact Overview http://www.ts.mah.se/RUP/RationalUnifiedProcess/process/artifact/ovu_arts.htm

6. Apéndice



DSW Construcción

Construcción

1. Introducción

[La introducción del Documento de Construcción del Software (DCS) ofrece una visión general de todo el documento. Incluye el propósito, alcance, definiciones, acrónimos, abreviaturas, referencias, y una descripción general de la fase de Construcción.]

1.1 Propósito

[Breve descripción del propósito del DCS. Se describe brevemente la estructura del documento, primordialmente se deberá indicar a quién va dirigido este documento así como la manera en que se espera sea interpretado el contenido de éste.]

1.2 Alcance

[Breve descripción del alcance del DCS, indicando qué es influenciado o afectado con el contenido de este documento.]

1.3 Definiciones, Acrónimos y Abreviaturas

[Proveer de las definiciones, términos y acrónimos requeridos, para la correcta interpretación del documento. Se puede incluir una referencia al glosario de términos en caso de que éste exista.]

Término	Definición
[Término, acrónimo o abreviatura]	[Definición del término]

1.4 Referencias

[Esta sección provee un listado de todos los documentos a los que se haga referencia dentro del contenido del DCS. Este debe ser lo suficientemente específico para poder localizarse, se puede incluir el identificador o nombre del documento referido. Especificar la fuente de donde se ha obtenido la referencia.]

2. Descripción general

[Esta sección debe describir lo que el resto del DCS contiene y explica la forma en que el documento está organizado.]

3. Componentes

[Esta sección debe contener el listado de los componentes necesarios para la construcción del sistema.]

El grueso de la producción de esta fase es el código para las clases dependiendo del lenguaje de programación señalado en la especificación de la implementación el equipo establecerá un estándar para los archivos que contengan el código, pudiendo ser éste el propuesto por en el estándar del lenguaje o alguno ideado por los miembros del equipo. Dicho formato deberá facilitar la identificación de las clases mediante la forma de nombrarlas y la distribución del código dentro

A 1. Plantilla correspondiente a Construcción.



DSW Construcción

5.1.2 <Descripción del defecto>

5.2 Pruebas de Caja Negra

5.2.1 <Nombre de la clase>

Método	Recibió	Esperado	Regresó
<Nombre del método>	<valor de los datos recibidos>	<valor de los datos esperados>	<valor de los datos obtenidos>

5.2.2 <Descripción del defecto>

6. Justificación de las decisiones de la Construcción

[Esta sección debe contener las justificaciones de las decisiones tomadas durante la Construcción.]

7. Información de apoyo

[La información de apoyo hace que el DCS sea más fácil de usar. Incluye:

Tabla de contenido:

Índice

Apéndices]

A 2. Plantilla correspondiente a Construcción.



DSW Construcción

3.1 A implementar

[Esta sección contiene el listado de los componentes que se necesitan implementar totalmente, es decir, aquellos componentes que resultan nuevos objetivos esperados.]

3.2 A modificar

[Esta sección contiene el listado de los componentes que se necesitan modificar, es decir, aquellos componentes existentes en la configuración y que requieren de modificaciones para su posterior integración al sistema.]

Es necesario incluir en el listado la versión y ubicación de los componentes a modificar.

4. Plan de pruebas unitarias

4.1 Descripción

[En el plan de pruebas unitarias se busca validar la correcta ejecución de cada una de las instrucciones que fueron codificadas en cada una de las clases.]

4.2 Requerimientos de las pruebas

[Aquí se debe incluir el listado de los requerimientos necesarios para obtener un ambiente que sea adecuado para realizar las pruebas y considerar a éstas válidas.]

4.3 Estrategia de las pruebas

[Existen varios tipos de pruebas unitarias, la Estructural basada en criterios de cobertura, tales como sentencias, decisiones, condiciones, bucles, cambios, etc.]

La Funcional, que se ocupa de valores límite, partición en clases de equivalencia, tablas de decisión y grafos causa-efecto.]

4.3.1 Objetivo

[Identificar defectos relacionados a la lógica e implementación de cada una de las diferentes unidades que constituyen el sistema, aplicando estrategias funcionales o estructurales.]

4.3.2 Técnica

[En este caso se propone utilizar pruebas de caja blanca que corresponden a aquellas que prueban la correcta ejecución de cada una de las instrucciones codificadas en las clases. Para determinar cuántos casos de prueba son necesarios es conveniente señalar todas las posibles trayectorias que un actor determinado tenga opción de seguir durante la ejecución del programa, tomando en cuenta los operadores condicionales que se encuentren en el código.]

Y por otro lado, las pruebas de caja negra, que toman como punto de partida los posibles valores de entrada asignados por el usuario y los valores esperados de salida del sistema, validando los resultados esperados con los obtenidos.]

4.3.3 Criterio de terminación

[Todas las pruebas previstas se han ejecutado y todos los defectos identificados han sido registrados.]

4.3.4 Consideraciones especiales

[Identificar o describir los temas o asuntos (internos o externos) que afectan a la aplicación y ejecución de las pruebas.]

A 3. Plantilla correspondiente a Construcción.



DSW Construcción

4.4 Herramientas

[Listado de las herramientas empleadas para cada grupo de casos de prueba.]

4.5 Recursos

[Esta sección presenta los recursos necesarios para llevar a cabo las pruebas; es un listado de las principales responsabilidades de cada recurso y el conocimiento y habilidades necesarias para completar las pruebas por parte de cada elemento.]

4.6 Casos de prueba

4.6.1 Pruebas de Caja Blanca

4.6.1.1 <Nombre de la clase>

4.6.1.2 <Descripción de la clase>

Método	Recibe	Regresa	Excepción
<Nombre del método>	<tipo de dato> <descripción del dato>	<tipo de dato> <descripción del dato>	<descripción de la excepción>

4.6.2 Pruebas de Caja Negra

4.6.2.1 <Nombre de la clase>

4.6.2.2 <Descripción de la clase>

Método	Recibe	Esperado
<Nombre del método>	<valor de los datos recibidos>	<valor de los datos esperados>

5. Reporte de pruebas unitarias

[En el reporte de pruebas unitarias se registrarán aquellos casos de prueba que registraron algún defecto, en particular todos aquellos que no regresaron los valores esperados.]

5.1 Pruebas de Caja Blanca

5.1.1 <Nombre de la clase>

Método	Recibió	Regresó	Excepción
<Nombre del método>	<tipo de dato> <descripción del dato>	<tipo de dato> <descripción del dato>	<descripción de la excepción>

A 4. Plantilla correspondiente a Construcción.