

**PEDECIBA Informática**  
**Instituto de Computación – Facultad de Ingeniería**  
**Universidad de la República**  
**Montevideo, Uruguay**

---

---

**Reporte Técnico RT 09-15**

---

---

**Adaptación de RUP para PSP:  
experiencia en el PIS**

**Leticia Pérez-Queiruga    Diego Vallespir**

**2009**

Pérez-Queiruga, Leticia; Vallespir, Diego  
Adaptación de RUP para PSP: experiencia en el PIS  
ISSN 0797-6410  
Reporte Técnico RT 09-15  
PEDECIBA  
Instituto de Computación – Facultad de Ingeniería  
Universidad de la República

Montevideo, Uruguay, 2009

# Adaptación de RUP para PSP: Experiencia en el PIS

Leticia Pérez-Queiruga, Diego Vallespir

Instituto de Computación, Facultad de Ingeniería, Universidad de la República,  
Montevideo, Uruguay.  
{lperez, dvallesp}@fing.edu.uy

**Resumen.** Este artículo presenta un modelo de procesos que tiene como base al proceso MUM y al Personal Software Process (PSP). MUM es un modelo de procesos símil RUP. De esta manera se fusiona un proceso a nivel grupal (MUM) con un proceso de desarrollo individual (PSP). También se presenta la ejecución y resultados primarios de 3 casos de estudio en el marco de un Programa de pruebas de procesos que se realizó durante 2007 y 2008, utilizando el modelo de procesos propuesto. Los resultados de estos casos de estudio muestran una diversidad de opiniones de los participantes que depende fuertemente de la experiencia previa de los mismos, y permite intuir algunas fortalezas y debilidades del modelo integrado, que se espera confirmar en futuros experimentos.

**Palabras clave:** Procesos de desarrollo de software, Ingeniería de software empírica, Personal software process, Rational unified process

## 1. Introducción

En el marco de un programa de construcción y prueba de modelos de procesos [1], surge la idea de conducir un estudio sobre la factibilidad de incorporar el Personal Software Process (PSP) [2] a un proceso de desarrollo de nivel grupal como forma de brindar soporte al desarrollo de software a nivel del individuo, en un contexto de desarrollo de software en equipo. En particular se estudia la factibilidad de adaptar el Rational Unified Process (RUP) [3] para trabajar en conjunto con el PSP. Existen ya algunos trabajos [4] que estudian la factibilidad de construir procesos que integren las prácticas del RUP y el PSP y logren brindar soporte tanto a nivel grupal como individual. En el caso particular de esta experiencia se adapta el PSP a un proceso tipo RUP llamado Proceso Modularizado Unificado y Medible [5].

El PSP es un modelo importante, sin embargo, normalmente es usado con el Team Software Process (TSP) [6], proceso de desarrollo para grupos cuyo uso aún no se ha extendido en la industria de Software. Un modelo difundido a nivel grupal es el RUP, por esto es interesante incorporar PSP a RUP y evaluar dicha incorporación.

En las secciones 2 y 3 se realiza una breve descripción del PSP y del MUM. A continuación, en las secciones 4 y 5 se presenta la adaptación construida y el caso de estudio llevado adelante. Por último, en la sección 6, se detallan las conclusiones.

## 2. Personal Software Process

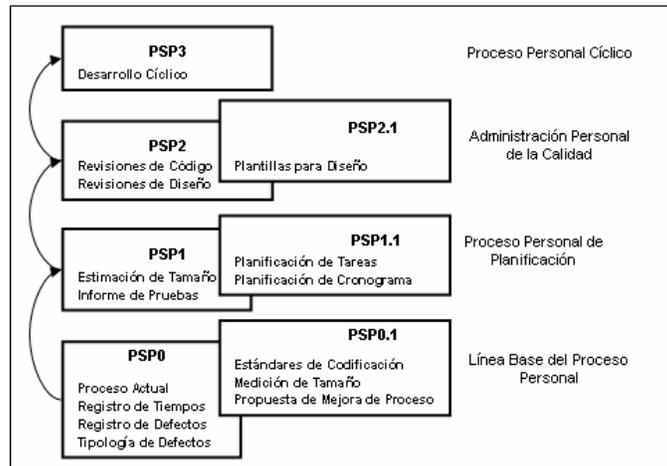
El PSP es un proceso formal de software, bien definido, estructurado y medible, diseñado para ser utilizado en el desarrollo individual de software. Es bien definido pues posee un conjunto de instrucciones de trabajo que guían en todo momento y sin ambigüedades al profesional en sus actividades como desarrollador de software. Es estructurado pues se compone de distintas fases o etapas de proceso que se suceden a lo largo del desarrollo, comenzado por la fase de planificación hasta llegar a la fase de postmortem y la obtención de un producto construido. Es medible pues posee un conjunto de métricas que permiten medir el proceso como base para la mejora continua del mismo. Fue desarrollado por Watts Humphrey del Software Engineering Institute (SEI) [2,7] de la universidad de Carnegie Mellon, para gestionar la práctica de la ingeniería de software a nivel individual. Su principal objetivo es, según Humphrey, ayudar a los ingenieros de software a mejorar sus prácticas personales de trabajo, y así mejorar la capacidad de planificación y seguimiento de su trabajo y la calidad de los productos obtenidos.

### 2.1. Modelo de Madurez de PSP

Para instruir a los ingenieros en los conceptos del PSP se utiliza un proceso de adaptación o madurez, a través del cual el ingeniero se va familiarizando gradualmente con sus buenas prácticas. Para esto Humphrey construye un framework de madurez el cual presenta una serie de siete niveles del proceso: PSP0, PSP0.1, PSP1, PSP1.1, PSP2, PSP2.1 y PSP3. Cada uno de estos niveles extiende el anterior introduciendo al ingeniero en un nuevo conjunto de prácticas y proporcionándole nuevos elementos como ser formularios, scripts y estándares, que lo asisten en la ejecución de dichas prácticas. Al terminar el nivel 2.1 el ingeniero domina todos los conceptos de PSP y puede ejecutar el proceso de forma completa. Durante el nivel 3 se introducen técnicas para el desarrollo de proyectos a gran escala. La estrategia de PSP3 es dividir el proyecto en pequeños módulos de forma de poder aplicar PSP2.1 en cada unos de ellos. La **Fig. 1** ilustra el modelo de madurez de siete niveles de PSP.

### 2.2. Estructura

El PSP proporciona una serie de elementos que guían y facilitan el trabajo de los profesionales. Estos elementos son: procedimientos, instrucciones, formularios, estándares y métricas.

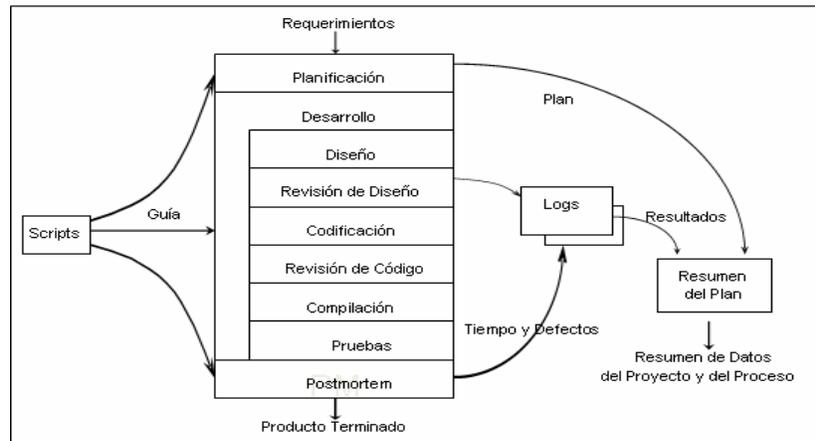


**Fig. 1.** Modelo de Madurez del PSP.

Los procedimientos describen la forma como el profesional debe realizar distintas tareas básicas de su labor como desarrollador de software. Las instrucciones (*Scripts*) definen en forma clara y sin ambigüedades los pasos a seguir para cada procedimiento. Los formularios y *logs* proveen el marco para recopilar datos del proceso. Le indican al profesional qué datos debe obtener y cómo debe registrarlos. Los estándares informan al profesional sobre la forma establecida o regulada de realizar determinadas tareas. Las métricas permiten al profesional medir su proceso y conocer su rendimiento. Las principales métricas utilizadas por el PSP son el tiempo dedicado en cada fase del proceso y la cantidad de defectos encontrados y removidos en el producto.

La figura **Fig. 2** muestra las distintas fases que componen la estructura del proceso PSP. Se comienza por la fase de Planificación, donde se construye una planificación para el proyecto. En esta fase el profesional cuenta con un conjunto de lineamientos, contenidos en un script de planificación, que le sirven de guía para realizar la planificación concreta, y un formulario para documentar el resumen del plan del proyecto le indica cómo y dónde registrar la planificación realizada. Luego de planificado el proyecto, comienza la fase de Desarrollo donde se implementa el producto. El desarrollador avanza en las distintas etapas de esta fase y el PSP le provee instrucciones (*scripts*) con los lineamientos de trabajo sugeridos para cada una de ellas. También el desarrollador registra el tiempo dedicado en cada etapa de esta fase (diseño, revisión de diseño, codificación, revisión de codificación, compilación y pruebas) y los datos sobre defectos encontrados y removidos en el producto en cada una de ellas. Para este registro el PSP le provee *logs* con un formato predefinido que le indican qué debe registrar y cómo, para que los datos resulten útiles. La última fase del proceso es la fase de Postmortem. Durante esta se vuelcan los datos de tiempos y defectos registrados en los *logs*, y el tamaño real del programa obtenido, al formulario que resume el plan de proyecto. Como resultado del proceso se obtiene no solo el producto de software construido sino también un resumen de lo ocurrido durante el proyecto documentado en el formulario de resumen del plan de proyecto. Este formulario contiene los resul-

tados planificados y reales sobre tiempos de desarrollo y los defectos detectados y removidos, junto con otras métricas derivadas, y es información historia a utilizar como base para la mejora continua del proceso.



**Fig. 2.** Estructura del Proceso PSP

### 2.3. Resultados Observados

En 1997 el SEI presenta dos informes donde se adelantan resultados sobre la aplicación del PSP tanto a nivel universitario [8] como en la industria [9]. En ellos se analiza el efecto del PSP sobre distintas áreas de mejora de procesos, tales como precisión en las estimaciones y calidad del producto y del proceso. El estudio realizado a nivel universitario observa una mejora en estas áreas, tras la aplicación del PSP, sin que esto resulte en una baja en la productividad. Asimismo, el estudio realizado en la industria expresa resultados similares. Se observa una mejora en la estimación de los cronogramas y un mejor cumplimiento de las fechas prometidas. Se describe también una importante mejora en la calidad de los productos obtenidos. Se observa una disminución de la cantidad de defectos por KLOC siendo que en algunos proyectos no se encontraron defectos en las pruebas de aceptación.

### 3. Proceso Modulado Uniforme y Medible (MUM)

El proceso MUM se desarrolla en el marco de un programa de construcción y prueba de modelos de procesos de desarrollo de software del Grupo de Ingeniería de Software (GrIS) del Instituto de Computación (InCo) de la Facultad de Ingeniería, de la Universidad de la República<sup>1</sup>. Es un proceso de desarrollo de software basado fuer-

<sup>1</sup> Grupo de Ingeniería de Software: <http://www.fing.edu.uy/inco/grupos/gris>

temente en el RUP. En particular se trata de una adaptación o configuración del RUP para un ambiente académico. Esta adaptación se aplica a la construcción de software en el contexto de la asignatura de grado "Proyecto de Ingeniería de Software" (PIS) [10], de la carrera de Ingeniero en Computación.

### **3.1. Construcción y Prueba de Modelos de Procesos**

La construcción y prueba de los modelos de procesos es dirigida por docentes del GrIs, utilizando como base el proceso MUM, y para su implementación se trabaja con estudiantes que están finalizando la carrera de Ingeniero en Computación. Los procesos diseñados son ciertamente restringidos en su uso ya que son concebidos para ser aplicados y probados en el marco de la asignatura PIS, la cual tiene ciertos criterios de trabajo establecidos. Los criterios más relevantes son:

- La duración del proyecto es de 14 semanas
- Los grupos de estudiantes se conforman por entre 10 a 14 personas.
- La carga horaria prevista es de 15 horas por persona por semana.
- El cliente es externo al cuerpo docente y la mayoría de las veces a la Facultad.
- Cada grupo de estudiantes tiene un docente como director del proyecto.
- Una vez por semana los estudiantes tienen reunión de monitoreo con su director.

Los grupos tienen una agenda definida que deben cumplir dentro de las 14 semanas. En las últimas dos semanas se instala el producto construido, en el ambiente del cliente y no se realiza ningún mantenimiento.

Este programa trabaja en base a tres etapas o fases en la construcción y prueba de modelos de procesos. Primero se define el proceso tomando como base las necesidades o el contexto del ambiente bajo el cuál será aplicado. Como segundo paso se valida dicho proceso a través de su aplicación en la asignatura PIS. En un tercer y último paso se evalúa la aplicación del proceso y los resultados obtenidos. De esta forma se identifican las mejoras al proceso definido, obteniendo un proceso ajustado y mejorado que puede ser puesto en práctica nuevamente al siguiente año. Esto permite la mejora continua de los modelos de procesos construidos.

### **3.2. Concepción del Proceso MUM**

El MUM surge en el año 2005, de la evolución de sucesivos procesos de software creados y probados a lo largo de 5 años [11], en el marco de las investigaciones del grupo GrIs.

Su nombre deriva directamente de las características de su creación. Es modularizado pues para su construcción se modularizó la información de sus componentes, de forma que sea sencilla la modificación o sustitución de componentes del proceso. Es unificado debido a que se unifican las actividades comunes a los distintos lenguajes de programación o metodologías, en un esqueleto común e instanciando las actividades que son propias de cada metodología. Es Medible porque se incorporaron métricas al proceso con el fin de poder medir y servir como elemento objetivo para cuantificar mejor la calidad de los procesos y tener una medida fiable para comparar con

futuros procesos. Desde el 2005 hasta la fecha este proceso ha sido ajustado y mejorado en forma continua a través de las sucesivas experiencias de aplicación dentro del PIS [12].

### 3.3. Aspectos Clave

Por tratarse de una adaptación del RUP, el proceso MUM toma de él sus características más relevantes. Es así que al igual que el RUP, el MUM está guiado por casos de uso, centrado en la arquitectura y es iterativo e incremental.

El MUM incorpora también desde el RUP muchas de las buenas prácticas actuales en el desarrollo de software como ser:

- Administración de requerimientos.
- Uso de arquitecturas basadas en componentes.
- Modelado visual de software, utilizando para ello el lenguaje UML
- Verificación de la calidad del software
- Control de cambios.

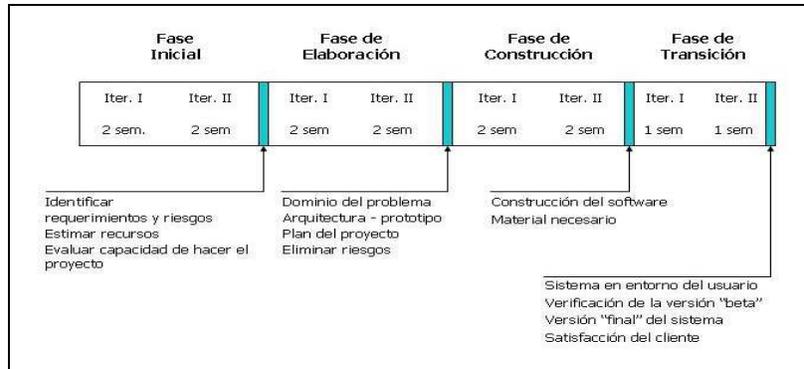
### 3.4. Estructura

La estructura del modelo MUM se describe, al igual que el RUP, en dos dimensiones: dimensión del tiempo y dimensión del modelo.

#### – Dimensión del Tiempo

Los aspectos básicos de esta dimensión se toman sin modificaciones desde el RUP, el proceso se divide en ciclos donde cada ciclo trabaja para construir una nueva versión del sistema. Asimismo, cada ciclo se divide en cuatro fases: Inicial, Elaboración, Construcción y Transición. Cada fase se divide en iteraciones y tiene objetivos definidos que al cumplirlos indican su finalización y habilitan a comenzar con la fase siguiente.

La adaptación del modelo en esta dimensión responde a que la duración de los proyectos en el PIS es acotada a 14 semanas de ejecución. Por tanto es el alcance del proyecto lo que debe ajustarse al tiempo de ejecución y no al revés. Se define una agenda de trabajo de 14 semanas de duración, la cual indica claramente la duración de cada fase y sus iteraciones en semanas e indica también para cada semana las actividades que se deben realizar y los objetivos a cumplir. La **Fig. 3** muestra dicha agenda. En particular se realiza sólo un ciclo, las fases tienen solamente dos iteraciones y cada una de las iteraciones tiene una duración de dos semanas. La excepción es la fase de transición la cuál tiene también dos iteraciones pero de una semana de duración cada una.



**Fig. 3.** Agenda del MUM: Fases, Iteraciones y Objetivos

– Dimensión del Modelo

Los conceptos básicos de la dimensión del modelo también se toman sin modificaciones desde el RUP. Se basa en cuatro elementos de modelado principales para describirla, las disciplinas, los roles, las actividades y los entregables que se corresponden directamente con los artefactos del RUP. El modelo MUM utiliza los mismos elementos de modelado utilizados en el proceso RUP y no se realizan cambios en la definición de los mismos.

La adaptación del modelo en esta dimensión está dada por las actividades a realizar, los roles que deben ejecutarlas y los entregables que deben producir, así también como las disciplinas necesarias para agrupar y ordenar dichas actividades en forma lógica. El MUM cuenta también con un juego de planillas, checklists y plantillas predefinidas para cada entregable, que guían a los distintos roles en la ejecución de sus tareas y la elaboración de los entregables resultantes.

El modelo MUM, conserva la mayoría de las disciplinas principales del RUP aunque elimina algunas y agrega nuevas que resultan necesarias en el contexto del PIS. Las disciplinas del modelo MUM se dividen en dos grupos, Disciplinas Básicas y Disciplinas de Gestión. Las disciplinas básicas involucran las actividades de ingeniería tradicionales del desarrollo de software y son: Requerimientos, Diseño, Implementación, Verificación, Implantación. Las disciplinas de gestión son aquellas que brindan “soporte” a las Disciplinas Básicas y se realizan en forma paralela a éstas, siendo aquellas: Gestión de Configuración (SCM), Gestión de Calidad (SQA) y Gestión del Proyecto (GP), Comunicación y, Formación y Entrenamiento.

El MUM toma sus roles desde el RUP pero realiza algunas modificaciones ya que en algunos casos no aplican pues corresponden a las disciplinas no incluidas y en otros casos las actividades que realizan están distribuidas en otros roles. Asimismo, algunos roles son creados para ejecutar las nuevas actividades necesarias para adaptar el modelo al contexto del PIS. Algunos de los principales roles definidos en el modelo MUM son:

- Analista: Responsable de entender el sistema a construir, conocer en detalle todos los requerimientos del sistema, y modelarlos en función de Casos de Usos.

- Arquitecto: Responsable de la arquitectura del sistema. Tiene la visión global del producto a construir, coordina las actividades técnicas a lo largo del proyecto.
- Implementador: Responsable de implementar, verificar y documentar los componentes y subsistemas que le son asignados.
- Especialista Técnico: Responsable de definir la herramienta de desarrollo que mejor se adapte al sistema a construir. Estudia las tecnologías que se consideren necesarias para el desarrollo del sistema.
- Responsable de Verificación: Responsable de que el sistema cumpla con los requerimientos establecidos para el mismo.
- Responsable de SQA: Responsable de la calidad del sistema y del apego al proceso en la construcción del mismo.
- Administrador: Responsable del seguimiento y control del proyecto, y de las mediciones y estimaciones del proyecto. Asigna recursos, establecer prioridades y coordinar las interacciones con clientes y usuarios.
- Coordinador de Desarrollo: Responsable de planificar el desarrollo de forma tal de cumplir con la planificación del proyecto.
- Director de Proyecto: Encargado de todos los factores políticos del proyecto y de asesorar al equipo de trabajo.
- Consultor: Encargado de realizar las reuniones de apoyo en las disciplinas que sean necesarias y brindar apoyo a los distintos roles dentro del equipo en la realización de sus actividades.

### 3.5. Métricas

El proceso MUM provee un juego de métricas que permiten medir determinados procesos, para cuantificar su calidad. Este se compone de:

- Métricas de Aceptación: Están vinculadas con la satisfacción del cliente.
- Pruebas Cubiertas: Evalúan qué porcentaje de las pruebas es efectivamente probado. Están vinculadas a la verificación del sistema.
- Desempeño de las Pruebas. Mide en qué porcentaje se corrigieron los errores detectados.
- Efectividad de las Pruebas: Mide la efectividad del conjunto de pruebas, para encontrar errores.
- Métrica de Productividad Orientada al Tamaño del Producto: Se utiliza para obtener una estimación de la productividad al comienzo del proyecto. Al final del mismo, en la fase de implantación, se toma nuevamente para observar la productividad real y comparar.
- Estado del Funcionamiento: Mide qué porcentaje de las funciones del sistema dan como resultado un dato no esperado o inválido, o el usuario encuentra dificultad al utilizarlas.

## **4. Adaptación MUM Extensión PSP**

La extensión del proceso MUM con la incorporación de las prácticas PSP busca obtener los beneficios reportados por el PSP en cuanto a las mejoras en áreas como planificación y calidad de proyectos, mientras que se mantienen las bondades de un proceso tipo RUP que asiste en la dirección de proyectos de desarrollo de software en equipo.

### **4.1. Construcción de la Extensión**

La extensión PSP [13] toma como base el proceso MUM y el PSP en su nivel de madurez PSP2.1. Se trabaja sobre el proceso MUM realizando las modificaciones necesarias a su conjunto de elementos de proceso para incorporar las prácticas de software requeridas por el PSP. De esta manera se mantiene el proceso PSP original y las condiciones bajo las cuales se han observado las ventajas de su aplicación práctica en los distintos estudios realizados desde su creación.

La selección del nivel de proceso PSP2.1 se debe a que es a este nivel donde se cuenta con un proceso bien definido, disciplinado, ordenado y medido que incorpora todas las buenas prácticas personales a utilizar en el desarrollo de pequeños programas o módulos de software. Asimismo, si bien existe un siguiente y último nivel de madurez del PSP, el PSP3, se decide no utilizarlo. El PSP3 incorpora el concepto de proceso cíclico para la construcción de software a mayor escala. En el caso particular de esta extensión, el marco cíclico lo proporciona el proceso MUM ya que el mismo es un proceso iterativo incremental que permite manejar la complejidad de un sistema de software a gran escala.

### **4.2. Estructura de la Extensión PSP**

Las adaptaciones más importantes al proceso MUM se dan en la dimensión del modelo, ya que el PSP incorpora actividades, junto con sus plantillas y entregables asociados, que no están definidos originalmente en el proceso MUM.

#### **4.2.1. Dimensión del Modelo: Roles**

No se incorporan nuevos roles a la extensión dado que el PSP no lo requiere. El impacto a nivel de los roles se percibe en que algunos de ellos se ven directamente afectados en su trabajo ya que deben ejecutar nuevas actividades y en algunos casos deben cambiar la forma de ejecución de algunas actividades que ya le son propias.

Los roles en los que ocurren modificaciones en sus actividades son: Implementador, Consultor PSP, Coordinador de Desarrollo y Especialista Técnico. Los restantes roles no presentan modificaciones en las actividades que ejecutan.

- Consultor PSP: Si bien no es necesario incorporar nuevos roles a los ya existentes, se decide utilizar en la extensión PSP el rol consultor PSP configurado a partir del rol de consultor el cuál está definido en el MUM como un rol externo y opcional que puede utilizarse si el proyecto lo requiere. La finalidad de la incorporación del

rol consultor PSP es brindar apoyo a los restantes roles dentro del equipo en la ejecución de sus actividades PSP.

- Implementador: Es el rol más afectado por la extensión PSP ya que las prácticas de software incorporadas por el PSP afectan directamente a los ingenieros que realizan la implementación del código. Es por esto que el rol implementador aumenta la cantidad de actividades que debe realizar, dado que ahora debe incluir nuevas prácticas a su proceso personal.
- Especialista Técnico: Una de las principales responsabilidades de este rol es la definición de estándares. Dado que el PSP promueve la utilización de estándares, algunos de los cuales no están contenidos en el proceso MUM, este rol será el encargado de definir los nuevos estándares requeridos por la extensión PSP.
- Coordinador de Desarrollo: Este rol modifica la forma de ejecución de algunas de sus actividades para aprovechar las ventajas provistas por el hecho de que los implementadores usan el PSP

#### **4.2.2. Dimensión del Modelo: Plantillas**

Junto a la incorporación de nuevas actividades PSP a las distintas disciplinas, también es necesario incorporar las plantillas PSP que se utilizan en la ejecución de las mismas. Asimismo, dado que no se identifican plantillas duplicadas entre el proceso MUM y el proceso PSP, la mayoría de las plantillas provistas por el proceso MUM se utilizan sin modificaciones en la extensión PSP.

#### **4.2.3. Dimensión del Modelo: Disciplinas**

La extensión PSP debe incorporar los objetivos de calidad de procesos, promovidos por el PSP y que no están contemplados en el proceso MUM. Esto implica definir nuevas actividades que se agrupan en las distintas disciplinas y permiten llevar adelante las prácticas de software introducidas por el PSP tendientes a cumplir con estos objetivos. Estas prácticas de calidad están vinculadas a cuatro disciplinas, Diseño, Implementación, Gestión de Proyecto y Gestión de Calidad, las cuales son modificadas para incorporar dichas actividades a su flujo de trabajo. Las restantes disciplinas no son modificadas en esta primera versión de la extensión, por no verse afectadas directamente.

#### **4.2.4. Dimensión del Modelo: Actividades**

La extensión PSP incorpora actividades que permiten llevar adelante las prácticas de software propias del PSP. Pero no solamente es necesario agregar actividades, sino que también se modifican actividades preexistentes del proceso MUM para adaptarlas al trabajo conjunto con el PSP.

- Disciplina Diseño

El PSP tiene como uno de sus objetivos principales la administración de la calidad basada específicamente en la gestión de defectos. Para el PSP un bajo contenido de defectos es un prerequisite esencial para un proceso de software de calidad. Así, el PSP incorpora la remoción temprana de defectos basada en revisiones individuales de diseño y código, y la construcción de diseños de software, completos y no ambiguos, por parte de los desarrolladores, para alcanzar el bajo contenido de defectos en el

software. Esto implica la incorporación a la disciplina *Diseño* de nuevas actividades que llevan adelante estas prácticas y que no estaban soportadas por el proceso MUM.

– Disciplina Implementación

Las prácticas de calidad propuestas por el PSP afectan tanto a la disciplina de diseño como a la de implementación. Para alcanzar un bajo contenido de defectos en el software desarrollado, no solo es necesario realizar un buen diseño sino también codificar correctamente en base a ese diseño, siguiendo estándares de codificación y realizar la posterior revisión del código en busca de los posibles errores cometidos y de su apego a los estándares.

Las revisiones tendientes a la remoción temprana de defectos propuestas por el PSP requieren que los ingenieros en forma individual revisen su código en busca de defectos, antes de compilarlo y testearlo.

Asimismo, el PSP incorpora la idea de considerar la compilación como un mojón importante en el proceso y el cual debe considerarse como una actividad en si misma.

Para contemplar estos objetivos es necesario incorporar actividades a la disciplina *Implementación*, que los lleven adelante y modificar algunas de las ya existentes.

– Disciplina Gestión de Calidad

El PSP enfatiza la idea de que cada desarrollador obtenga y analice sus métricas personales de proceso, de forma tal de utilizar esta información para supervisar, controlar y mejorar sus procesos personales de software. Asimismo, propone que cada desarrollador identifique, elabore y registre propuestas que ayuden a la mejora de su proceso personal. Para incluir estos dos nuevos objetivos, debe modificarse la disciplina *Gestión de Calidad* con la incorporación de nuevas actividades que los contemplan.

– Disciplina Gestión de Proyecto

El MUM proporciona limitada ayuda a los miembros del equipo de desarrollo sobre cómo gestionar su trabajo en cada asignación de tareas. El administrador del proyecto asigna tareas y responsabilidades y queda en manos de cada miembro del equipo planificar su trabajo individual. El PSP incorpora la idea de que el implementador gestione su trabajo individual realizando sus propias estimaciones de tamaño de software y tiempo de desarrollo basado en sus datos históricos de productividad, y planifique en base a esto su trabajo. Es por esto, que para incluir estos nuevos objetivos, debe modificarse la disciplina *Gestión de Proyecto*, con la incorporación de nuevas actividades y modificaciones a algunas de las ya existentes.

#### **4.2.5. Dimensión del Tiempo**

La extensión PSP en esta dimensión se centra en definir cómo utilizar el proceso MUM para cubrir el marco cíclico para la construcción de sistemas de mediano porte utilizando PSP. Más precisamente se define cómo integrar las iteraciones del MUM con los ciclos PSP2.1 utilizados en la construcción de los distintos módulos de software que finalmente construyen el sistema.

En esta dimensión no se realizan cambios respecto del proceso MUM. Aunque se han agregado nuevos objetivos a las distintas disciplinas para atender las prácticas

promovidas por el PSP, las fases del proceso MUM (Inicio, Elaboración, Construcción y Transición) mantienen su duración y objetivos originales. Incluso se mantiene la duración de las iteraciones tal cual se muestran en el **Fig. 3**. Esto es así porque el tiempo de duración de los proyectos debe continuar de 14 semanas.

En este contexto y analizando la agenda de trabajo original del MUM, se define que un ciclo PSP2.1 para la construcción de cada módulo o nueva funcionalidad del sistema, dura lo que dura una iteración MUM. Esto es así pues en cada iteración MUM un implementador construye un nuevo módulo o funcionalidad que finalmente se integra a lo ya implementado construyendo así, en forma iterativa incremental, el producto final. Justamente es allí, en la implementación de estos módulos componentes del sistema final, donde se aplican los conceptos del PSP. Por lo tanto cada iteración MUM desarrolla un ciclo completo PSP2.1.

#### **4.2.6. Dimensión PSP**

Una nueva dimensión es agregada al proceso, la dimensión PSP. Esta dimensión permite observar la extensión PSP desde el punto de vista del PSP. Representa el proceso PSP con su secuencia de fases. Indica las características del ciclo de vida del proyecto expresado en términos de fases del proceso PSP. En esta dimensión el proyecto se divide en tres fases principales: Planificación, Desarrollo y Postmortem, donde a su vez la fase de diseño se divide en Diseño, Revisión de diseño, Codificación, Revisión de codificación, Compilación y Pruebas. Esta dimensión se incluye solamente con el propósito de orientar al implementador sobre en qué etapa del proceso PSP se encuentra cuando realiza las distintas actividades PSP incorporadas al modelo. Esto lo ayudará en la tarea de registrar el tiempo que dedica a cada fase del proceso PSP.

#### **4.2.7. Otros Elementos del modelo**

El MUM proporciona un conjunto de métricas orientadas a medir la calidad del producto construido y del proceso grupal utilizado para construirlo. Pero, no proporciona métricas a nivel individual que asistan al ingeniero en la tarea de administrar la calidad de su proceso personal de software. Por otra parte, el PSP proporciona un conjunto de métricas de nivel personal. Por esto, la extensión PSP mantiene las métricas provistas por el proceso MUM para medir la calidad del proceso grupal, e incorpora un cuadro de métricas provisto por el PSP para asistir a los implementadores en la tarea de medir, analizar y mejorar sus procesos de software a nivel individual.

## **5. Caso de Estudio**

Las actividades del programa de pruebas de procesos se realizan con estudiantes de 4to año de la carrera Ingeniería en Computación dictada en la Facultad de Ingeniería, Universidad de la República, en el contexto de la asignatura PIS, presentada en la sección 3.1. La asignatura PIS se estructura en base a grupos de estudiantes, cada uno de los cuáles lleva adelante un proyecto de software que abarca las etapas de análisis, diseño, construcción y prueba de un producto. Se trata de una experiencia guiada y

controlada, pero sometida a restricciones análogas a las que ocurren normalmente en la industria.

Para la experiencia se forman grupos de estudiantes, que utilizando el MUM Extensión PSP presentado en la sección 4, llevan adelante un proyecto solicitado por un cliente. Los proyectos implican la construcción de sistemas de software de mediano porte, que en todos los casos deben ser implementados utilizando lenguaje Java.

Para analizar comparativamente los resultados obtenidos por los grupos que siguen el proceso MUM Extensión PSP, se crean grupos espejo de los mismos pero que no utilizan el PSP. Son asignados a los mismos clientes y tienen que desarrollar los mismos proyectos, pero utilizando el proceso de desarrollo MUM “puro”, sin PSP.

Antes de poder participar de la experiencia, los estudiantes deben conocer el PSP y sus principios básicos. Es por esto que se les imparte un “Taller de PSP”. Este taller está basado en el libro de Humphrey, “*A Discipline for Software Engineering*” [3]. La estructura y forma de ejecución del mismo es muy similar a la del curso oficial impartido por el SEI, pero adaptado para utilizar el lenguaje Java en lugar de C++.

La experiencia se ejecuta en los años 2007 y 2008, con tres grupos de estudiantes. Los mismos se componen de 13 estudiantes cada uno y son mixtos pues participan estudiantes que cursaron el taller de PSP previamente y estudiantes que no y no conocen el PSP en absoluto. Los estudiantes que cursaron el taller de PSP asumen todos y cada uno, el rol implementador del proceso MUM extensión PSP. Los proyectos a realizar son diferentes para cada grupo y pertenecen a distintos clientes, pero tienen en común que deben ser desarrollados utilizando lenguaje Java. En particular en los tres casos se utiliza la plataforma Java Enterprise Edition (Java EE) con sus tecnologías, EJB3 (Enterprise Java Beans) para resolver la lógica de negocio, Servlets y JSP (Java Server Pages) para la interfaz web y JPA (Java Persistence API) para la persistencia. Se conforman así tres casos de estudio diferentes para probar la extensión PSP.

Los casos I y II se ejecutan en el 2007. Cada grupo se compone de 8 estudiantes que cursaron el taller de PSP y 5 que no. En el caso I desarrollan un sistema enteramente web orientado al manejo de la información derivada del proceso de testing de productos de software. En el caso II construyen un editor de reglas de negocio donde usuarios convencionales pueden crear, modificar y editar reglas.

El caso III se ejecuta en el 2008, y el grupo se compone de 11 estudiantes participantes del taller y 2 que no conocen el PSP. El proyecto llevado adelante implica la construcción de un sistema enteramente web, para la gestión de una librería.

Para el caso I, consultados los estudiantes sobre algunas desviaciones observadas, estos informan que no siguieron el proceso MUM extensión PSP como está descrito y que en algunos casos los datos registrados y entregados fueron intencionalmente alterados. Asimismo el caso III, tampoco siguió el proceso MUM extensión PSP. A mediados del proyecto los estudiantes informaron que les resultaba dificultoso aplicar el PSP ya que en general al implementar trabajan en parejas y esto no les permite identificar y registrar medidas reales y objetivas sobre sus procesos personales.

Es por lo anterior, que para el análisis de los resultados de la ejecución del proceso, se descartan los casos I y III. Pero para analizar la opinión de los estudiantes acerca del proceso, sí se consideran los tres casos, puesto que los obstáculos identificados son la causa probable por la cual los grupos I y III deciden no aplicar el proceso.

Desde nuestro punto de vista y a la luz de los resultados observados durante el seguimiento del caso de estudio II, la experiencia resultó positiva pues se logró aplicar

el proceso y los estudiantes lograron construir el producto requerido por el cliente. Asimismo el cliente externo expresó tanto personalmente como a través de una encuesta de satisfacción su conformidad con los resultados obtenidos. Pero en esta ocasión no nos interesa hacer foco en los resultados particulares sobre la calidad y la planificación sino en la percepción de los estudiantes que participaron de la experiencia actuando como implementadores.

Los tres grupos participan de una encuesta de satisfacción realizada en forma individual, al término de la experiencia. Al consultarles sobre si les gustó participar de la experiencia, el 100% del grupo II responde que sí, mientras que en los restantes grupos la opinión está dividida. Quienes indican que no les gustó la experiencia, mencionan que no se sintieron cómodos trabajando con el modelo. Lo encontraron demasiado formal y tedioso. La mayoría de los estudiantes, independientemente del grupo al que pertenecen, encuentran el proceso complejo. Creemos que esto no es una característica propia del proceso MUM Extensión PSP, sino que ya el MUM puro les resulta complejo pues para la mayoría de estos estudiantes es la primera vez que se enfrentan a trabajar en el desarrollo de un sistema, formando parte de un equipo de desarrollo y siguiendo un proceso formal.

Los estudiantes de los grupos I y III destacan como un obstáculo importante, el hecho de no poder programar en pares. Los proyectos que deben llevar adelante implican utilizar tecnologías en las cuales no tienen experiencia y la forma que encuentran para lidiar con esta complejidad es trabajar en pares. Esto es para ellos una limitante a la hora de medir sus procesos personales en forma individual. Esta es una de las primeras causas en importancia que los grupos I y III esgrimen para abandonar el MUM Extensión PSP. El grupo II no hizo mención de este problema.

Consultados sobre si utilizar la Extensión PSP perjudicó el trabajo en equipo, las opiniones están divididas, un 50% opina que sí y el otro 50% que no. De todas formas los tres grupos coinciden en que el hecho de que los grupos sean mixtos genera inconvenientes. En particular el grupo I menciona que entre otras cosas, los estudiantes que no conocen el PSP y no cursaron el taller ven su aplicación como una fuente de retrasos en el proyecto. Los tres grupos coinciden en que si los grupos fueran conformados enteramente por estudiantes que conocen el PSP y todos lo aplicaran durante el proyecto se verían mejores resultados.

Más del 50% de los estudiantes consideran que el PSP los ayudó a mejorar sus prácticas personales. Explican que les permitió conocer su productividad y a partir de ella planificar y estimar mejor sus tareas individuales. También señalan que les permitió conocer la tipología de los errores que cometen y trabajar sobre ellos. De hecho señalan estos aspectos como las fortalezas del proceso. Los tres grupos coinciden que conocer la productividad de los implementadores les ayudó a realizar las estimaciones necesarias para definir el alcance del proyecto en forma temprana. El grupo II, señala también que detectaron menos errores de los esperados en la fase de test del producto.

Finalmente se consulta a los estudiantes sobre si piensan seguir utilizando el PSP. El grupo II dicen que sí, mientras que los grupos I y III dicen que no. De todas formas la mayoría de los estudiantes menciona que si bien no piensan aplicar el proceso rigurosamente ya han adquirido algunas de las buenas prácticas y seguirán aplicándolas.

## 6. Conclusiones

Este artículo presenta un modelo de procesos que tiene como base al proceso MUM y al PSP. Este nuevo modelo debe respetar tanto el modelo de desarrollo grupal (MUM) como el modelo PSP, para no perder ninguna de sus bondades. La propuesta se presenta de forma breve y sin entrar en detalles según las distintas dimensiones de MUM: roles, plantillas, disciplinas, actividades y tiempo. El proceso se probó en el marco del Programa de pruebas de procesos del GrIs con interesantes resultados. En particular hemos presentado la percepción del equipo de desarrollo acerca del modelo de proceso propuesto. Varios de los resultados son contradictorios y fueron dependientes del grupo de desarrolladores. Actualmente se están analizando estos y otros resultados de los casos de estudio y planificando nuevas experiencias para 2009 y 2010.

**Agradecimiento:** Los autores agradecen a Héctor Cancela por sus contribuciones al artículo.

## 7. Referencias

1. Triñanes, J.: Construcción de un banco de pruebas de modelos de proceso. JIISIC'04, España (2004)
2. Humphrey, W.S.: The Personal Software Process (PSP). Technical report, CMU/SEI-2000-TR-022, Software Engineering Institute, Carnegie Mellon (2000)
3. Krutchen, P.: The Rational Unified Process: An Introduction, 3 edition, Addison-Wesley Professional (2003)
4. Svensson, H.: Developing Support for Agile and Plan-Driven Methods. Phd Thesis, Royal Institute of Technology, Department of Computer and System Sciences (2005)
5. Proceso Modularizado, Unificado y Medible (MUM), Instituto de Computación, Facultad de Ingeniería, Universidad de la República, <http://www.fing.edu.uy/inco/cursos/ingsoft/pis/memoria/experiencia2008/MUM/index.htm>. Última fecha de acceso: 25/04/2009
6. Humphrey, W.S.: The Team Software Process (TSP). Technical report, CMU/SEI-2000-TR-023, Software Engineering Institute, Carnegie Mellon (2000)
7. Humphrey, W.S.: A Discipline for Software Engineering. Addison-Wesley (1995)
8. Hayes, W., Over, J.: The Personal Software Process: An Empirical Study of the Impacts of PSP on Individual Engineers. Technical Report, CMU/SEI-97-TR-001 (1997)
9. Ferguson, P., Humphrey, W.S., Khajenoori, S., Macke, S., Matvya, A.: Results of Applying the Personal Software Process, pp. 24--31. IEEE Computer (1997)
10. Proyecto de Ingeniería de Software, Instituto de Computación, Facultad de Ingeniería, Universidad de la República, <http://www.fing.edu.uy/inco/cursos/pis/wikiPIS/field.php/Curso/Programa>. Última fecha de acceso: 25/04/2009
11. Pedrana, M. L., Bellini, M. C.: Mejora de la Calidad de los Procesos de Ingeniería de Software: Proceso Modularizado Unificado y Medible. Proyecto de Grado de la Carrera Ingeniería en Computación, Instituto de Computación, Universidad de la República (2005).
12. Triñanes, J.: TLREQ: Proceso para desarrollo a distancia. JIISIC'03, Chile (2003)
13. MUM extensión PSP, Instituto de Computación, Facultad de Ingeniería, Universidad de la República, <http://www.fing.edu.uy/inco/cursos/ingsoft/pis/memoria/experiencia2008/MUM/psp/>. Última fecha de acceso: 25/04/2009