Integrando Casos de uso 2.0 y el Desarrollo Ágil de Software con procesos metacognitivos

Girbal Ernesto R., Gardella Carlos H., Peñalva Mirta del Carmen
Universidad Tecnológica Nacional, Facultad Regional La Plata

Abstract
El desarrollo de software se ve afectado conforme se modifican las necesidades del negocio. Los requerimientos cambian siguiendo las exigencias del mismo y su mercado, en este contexto, la planificación de proyectos y la administración de recursos en términos de tiempo, personas, y costos resulta altamente impactada. La importancia de los requerimientos manifestada en estudios sobre fracasos del software [1][2] y la evolución de la Ingeniería de Requerimientos han impulsado a las metodologías de desarrollo de software al tratamiento exhaustivo de los mismos como base fundamental de la producción de software de calidad. Han surgido nuevos enfoques denominados "métodos ágiles" como una respuesta que privilegia la adaptabilidad por sobre la predictibilidad permitiendo entregas tempranas y progresivas de software de valor.
Nuevos paradigmas irán emergiendo como solución a este y otros problemas, en consecuencia surge la necesidad de una continua adaptación del sujeto que aprende. Esto implica la responsabilidad de una intervención pedagógica adecuada para procurar se desarrollen procesos cognitivos no sólo para aprender sino para aprender mejor, es decir con menor esfuerzo y mejor rendimiento[3].
Al integrar Casos de Uso 2.0 con un enfoque metodológico ágil, pretendemos que los estudiantes construyan una nueva experiencia, que transfirieran los resultados del aprendizaje, a nuevas situaciones, logrando un "aprendizaje significativo"[4]. La metacognición se ve representada en la resolución de problemas, como la capacidad que adquiere el sujeto, observando procesos de pensamiento propios implicados al ejecutar una tarea, reflexionando y autorregulando su aprendizaje y como consecuencia transfiriendo ese capital a nuevas acciones.

Palabras Clave:
Requerimientos cambiantes, métodos ágiles, Casos de Uso 2.0, metacognición.

Introducción

El objetivo del presente artículo es presentar una nueva aplicabilidad de los Casos de Uso (C.U.) definidos en UML [5] y su integración con metodologías ágiles, que permita integrar conceptualmente la captura de requerimientos con una metodología ágil de desarrollo de software.
En la resolución de todo problema ingenieril la elección de un método o una técnica no sólo implica la aplicación del conocimiento adquirido, sino que subyacen otros aspectos como el desarrollo de una actividad metacognitiva [6] que presupone la capacidad que los sujetos tienen para planificar qué estrategias han de utilizar ante una situación, saber qué objetivos se quieren conseguir (*saber qué*), y saber cómo conseguirlos (*saber cómo*), aplicando, controlando, y evaluando el proceso para consolidarlo o modificarlo.
Existen varias formas de transitar un proceso de aprendizaje, en uno el sujeto cuenta con los conocimientos para enfrentar una situación particular y estar en condiciones de resolverla metodológicamente. En este aprendizaje que denominamos asociativo [7], las habilidades o técnicas aplicadas resultan básicas: repetir, memorizar, copiar, etc. No hay en este caso cambios estructurales sobre el saber adquirido. Consideramos adecuado un modelo pedagógico donde el objetivo sea el aprendizaje superador, por reestructuración [7] que permita la organización y la elaboración del conocimiento a través de otras habilidades tales como: hacer analogías, generar mapas conceptuales, redes de conceptos, identificar estructuras, definir palabras claves, etc.
De la misma manera que la aplicación de Casos de Uso 2.0 es extensiva a otras metodologías de desarrollo de software: Cascada, RAD, RUP, etc., también es utilizable en otros abordajes como por

ejemplo la modelización de procesos de negocios. A través de la interacción entre docente y alumno se pretende motivar y estimular el proceso metacognitivo, de forma tal que el estudiante pueda, sobre la base del conocimiento adquirido y la experiencia previa del caso planteado, visualizar y comprender la extensibilidad y la aplicabilidad de este nuevo enfoque de los Casos de Uso en nuevos escenarios.

PROCESO METACOGNITIVO

En todo proceso de enseñanza y aprendizaje es necesario contar con sujetos activos y pensantes para una apropiación exitosa del conocimiento. Mientras el rol del docente incluye activar el pensamiento del sujeto, éste deberá esforzarse mejorando la calidad de su trabajo intelectual a través de procesos de autorreflexión y crítica, es decir pondrá en ejecución procesos metacognitivos. Comenzaremos expresando que es la metacognición. Entre sus varias acepciones, el término *metá,* significa en griego "posterior a" o "que acompaña", por lo que podemos pensarlo como "*lo que acompaña a la cognición"*. Desde las primeras investigaciones a fines de la década de 1960 hasta la actualidad se ha revelado a la metacognición como un atributo del ser humano para supervisar y controlar su propia cognición. Es una actividad recursiva del pensamiento donde el sujeto, ante un problema, piensa sobre lo que conoce, planifica estrategias de solución, las aplica y evalúa la productividad de las mismas para mejorar su desempeño en futuras situaciones. Pero la metacognición va más allá, y además abarca el conocimiento interior de la persona: sus valores, sistemas de ideas, fortalezas y debilidades, es decir debe tener conciencia de los recursos con que cuenta. La visión actual de la psicología cognitiva asocia la motivación, la autorregulación, el interés por la meta y la metacognición como un sistema cooperante que facilita la eficacia ante nuevos aprendizajes [8].

A partir de esto, consideramos un aspecto central la incorporación de procesos de pensamiento sobre la actividad específica desarrollada en la integración de los dos enfoques planteados, que denominaremos procesos metacognitivos.
Algunos indicadores sugeridos para evaluar el proceso metacognitivo:

a) Buena ejecución de tareas cognitivas complejas.
b) Flexibilidad y perseverancia en la solución de la tarea.
c) Aplicación conciente de habilidades intelectuales.
d) Autointerrogación y crítica.
e) Buena gestión de recursos personales (motoras, percepción, lenguaje, creencias, conocimientos previos, memoria, destrezas de aprendizaje) para lograr el objetivo

BASE CONCEPTUAL DE CASOS DE USO 2.0

El sujeto que aprende tomará conocimiento de los principales conceptos básicos de la aplicación de Casos de Uso 2.0 con relación a los requerimientos. Esto comprende en primer término, conocer una serie de criterios que se deben considerar en relación a los mismos, para posteriormente ver los artefactos principales que lo constituyen. [9]

Criterios fundamentales:

Mantener los Casos de Uso simples

Los Casos de Uso proveen una forma simple y comprensiva de identificar y capturar las historias narradas por los stakeholders, permitiendo que los requerimientos del sistema sean fácilmente capturados, compartidos y comprendidos, además de ser ejecutables y testeables.

Entender el todo

Los Casos de Uso que modelan el sistema permiten rápidamente descubrir qué está incluido y que no lo está, creando una visión global del sistema fácilmente comprensible del contexto y los objetivos del sistema.

Enfocarse en el valor

Es importante determinar qué valor el sistema proveerá a los usuarios. Esto solamente se genera si el mismo es utilizado productivamente en el logro de sus metas. Los casos de uso que permiten capturar los requerimientos facilitan que los de mayor valor sean identificados inicialmente.

Construir el sistema en porciones

Identificar los requerimientos de mayor valor y uso, luego cortarlos en porciones, definir los casos de prueba que representen la aceptación de dichas porciones. Elegir la porción que constituye el núcleo que viaja a través del ciclo de desarrollo, de extremo a extremo del mismo y comenzar a construirlo. Porciones adicionales pueden ser tomadas del Caso de Uso e implementadas hasta que haya suficientes para proveer una solución usable a un usuario particular tal como se muestra en la Figura 1.

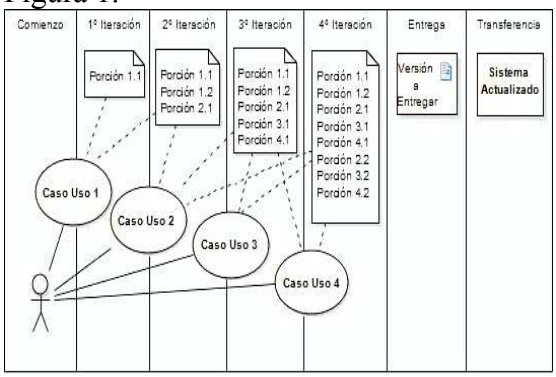

Figura 1. Casos de Uso, Porciones de Casos de Uso, Incrementos y Entregas

Contenidos Casos de Uso 2.0

Los Casos de Uso 2.0 comprenden los Casos de Uso, las historias de usuarios, y la "porción de caso de uso". El mapa conceptual de C.U. 2.0 descripto en la Figura 2 permite visualizar la relación existente entre los distintos conceptos involucrados en el ciclo de desarrollo de software.

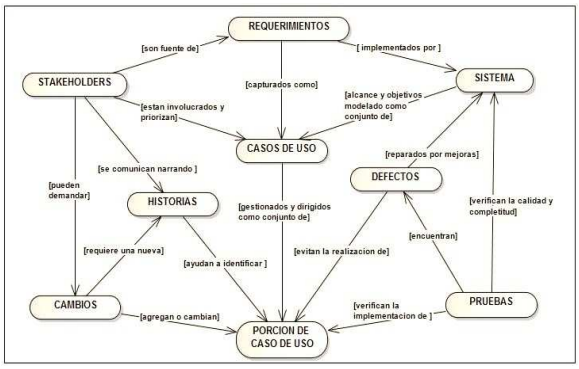

Figura 2. Mapa conceptual Casos de Uso 2.0

Ciclo de vida de un Caso de Uso

Un C.U. sufre muchos cambios de estado desde su identificación inicial hasta su realización por el sistema. Los estados constituyen un punto importante en la comprensión y el desarrollo de los mismos, dichos estados se describen en la Figura 3:

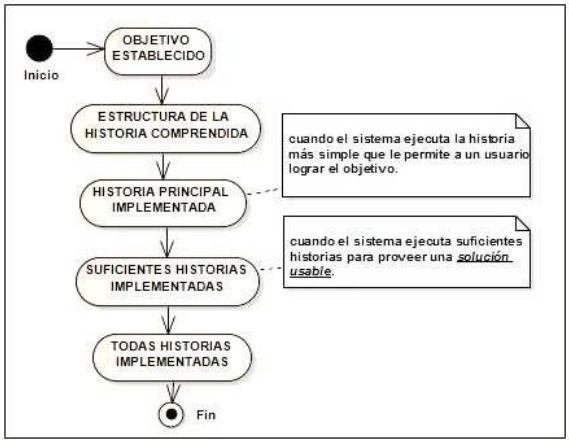

Figura 3. Ciclo de vida de un Caso de Uso

El modelo de Caso de Uso es complementado por información de soporte, que captura la definición de los términos usados en el modelo y en las historias que describen el caso [10] [11].

Ciclo de vida de una porción de Caso de Uso

Los casos de uso cubren muchas historias que varían en importancia y prioridad. Existen demasiadas historias para ser entregadas en forma total y generalmente demasiadas para trabajar en un solo incremento. Por esto se debe dividir el C.U. en piezas más pequeñas llamadas porciones, las cuales constituyen la característica más importante de los Casos de Uso 2.0 ya que conducen el desarrollo

del sistema hasta completarlo. Esta estrategia permite seleccionar qué piezas entregar, provee unidades adecuadas para desarrollar y probar, manejar piezas de trabajo pequeñas y de tamaño similar, que pueden ser rápidamente codificadas.
Los estados de las porciones de Casos de Uso permiten evaluar el progreso que se está realizando en comprender e implementar un Caso de Uso. Mientras una porción está siendo verificada, otra porción se puede estar implementando y una tercera siendo analizada como se describe en la Figura 4.

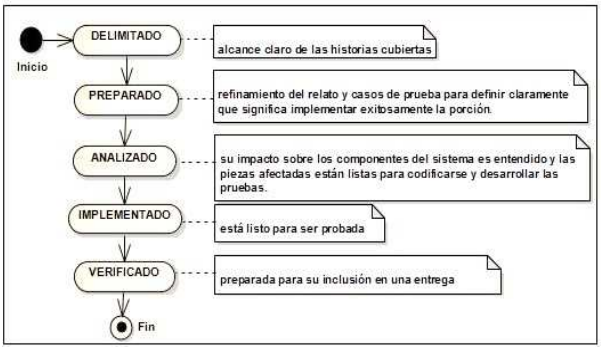

Figura 4. Estados de una porción de Caso de Uso

Historias

Los Casos de Uso se exploran relatando historias, donde cada historia de valor para los clientes, usuarios, y otros stakeholders es un hilo de uno de los casos de uso. Las historias en su naturaleza pueden ser funcionales o no funcionales. Son descritas como un conjunto de flujos, esto puede ser complementado por un conjunto de requerimientos especiales que pueden influir en las historias, permitiendo asignar las historias correctas a la porción de Caso de Uso para su posterior implementación y además definir los casos de prueba correctos. El flujo básico es mostrado como una secuencia lineal de pasos y los flujos alternativos son mostrados como desviaciones de este conjunto de pasos. En la Figura 5 se presenta un caso práctico correspondiente a la extracción de fondos en un cajero automático, mostrando el flujo básico y los alternativos.
Las historias constituyen una herramienta útil para encontrar la porción de Caso de Uso correcta y para determinar los Casos de Prueba que verifican esas historias. Si se requiere se pueden listar las historias del Caso de Uso como una sección extra en el relato del mismo. Si es necesaria mayor formalidad en el relato, se puede utilizar cualquier herramienta de documentación que se considere apropiada.

Figura 5. Ejemplificación de Flujos de actividades

Defectos y Cambios

Si bien no son una parte directa de C.U. 2.0, resulta importante comprender como los defectos y los cambios son tratados en los Casos de Uso y en las porciones. Los cambios requeridos se deben analizar contra el modelo de Caso de Uso, el Caso de Uso, y la porción de Caso de Uso. Esto permite que la magnitud del cambio sea rápidamente comprendida. Adicionar un nuevo Caso de Uso al sistema es un cambio mayor ya que cambia los objetivos totales del sistema y el propósito del mismo. Un cambio en un Caso de Uso existente es más pequeño, particularmente si se refiere a una historia que no ha sido destinada a una porción, o preparada, o analizada, o implementada, o verificada.
Los defectos son manejados mediante el seguimiento de la porción de Caso de Uso, como resultado de su detección durante la implementación o la verificación de la misma. La porción no puede continuar

implementándose hasta que el defecto sea tratado y superado.

ENFOQUE METODOLÓGICO PARA LA CAPTURA DE REQUERIMIENTOS Y EL DESARROLLO SOFTWARE EN C.U. 2.0

Comprendidos los conceptos básicos, el sujeto que aprende tendrá una visión integradora de "Casos de Uso aplicando porciones", a través de una serie de actividades, desde la determinación de los casos de uso hasta la implementación y prueba del software tal se describen en la Figura 6:

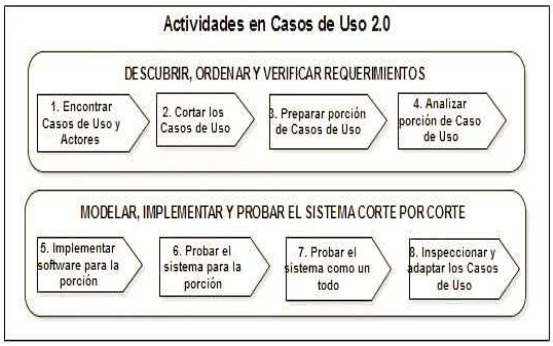

Figura 6. Mapa conceptual de actividades

1. Encontrar Casos de Uso y Actores

Se necesita encontrar actores y Casos de Uso que permiten:

- Acordar sobre los objetivos del sistema.
- Acordar sobre el comportamiento del sistema.
- Definir el alcance de las entregas del sistema.
- Acordar con el valor que el sistema provee.
- Identificar formas de usar y probar el sistema.

2. Cortar los Casos de Uso

Para crear la primera porción de Casos de Uso:

- Crear una porción de tamaño adecuado para trabajar.
- Ajustar el tiempo y el presupuesto disponible.
- Entregar el valor más alto a los usuarios.

3. Preparar una porción de Casos de Uso

Una vez que una porción es seleccionada para ser desarrollada:

- Prepararla para la implementación.
- Claramente definir que significa implementar exitosamente la porción.
- Definir las características del requerimiento.
- Enfocar el desarrollo del software hacia las pruebas que debe afrontar

4. Analizar porción de Caso de Uso

Antes de codificar se debe analizar la porción para:

- Comprender su impacto sobre los elementos del sistema que serán utilizados para implementarlo.
- Definir las responsabilidades de los elementos del sistema afectados.
- Definir como los elementos del sistema interactuarán para llevar a cabo el Caso de Uso.

5. Implementar software para una porción

Diseñar, codificar, probar la unidad, e integrar los componentes de software necesarios para implementar la porción de Caso de Uso.

6. Probar el sistema para la porción

Probar el software independientemente para verificar que la porción de Caso de Uso ha sido implementada exitosamente. Cada porción necesita ser probada antes de que pueda ser considerada completa y verificada. Esto se logra ejecutando los Casos de Prueba de la porción, la independencia de las porciones permite probarlas tan pronto como son implementadas y provee un feedback inmediato.

7. Probar el sistema como un todo

Cada incremento del sistema de software necesita ser probado para verificar que implementa correctamente todas las nuevas porciones de Casos de Uso sin afectar ninguna otra parte del mismo. Cada vez que

se genere un incremento, no es suficiente probar la porción implementada, sino que se debe también probar el sistema como un todo para asegurar que las porciones implementadas sean compatibles con el software generado.

8. Inspeccionar y adaptar los Casos de Uso

Se necesita continuamente refinar y evaluar los Casos de Uso y sus porciones para:

- Manejar los cambios.
- Seguir el progreso.
- Adecuar el trabajo en el tiempo y el presupuesto disponible.
- Refinar el tamaño de las porciones para incrementar el rendimiento del trabajo.

CASOS DE USO 2.0 Y SU APLICACIÓN EN UN MÉTODO ITERATIVO CONDUCIDO POR EL PRODUCTBACKLOG

El desarrollo ágil de software dentro de sus premisas prioriza [12]:

- El software que funciona, por encima de la documentación exhaustiva
- La respuesta al cambio, por encima del seguimiento de un plan.

El enfoque de Casos de Uso 2.0 permite establecer una correspondencia directa con los llamados métodos ágiles para el desarrollo de software en la cual el sujeto que aprende verá cómo los contenidos previos y las actividades de los Casos de Uso 2.0 se integran particularmente con una metodología ágil como Scrum, un marco de trabajo para la gestión y desarrollo de software basado en un proceso iterativo e incremental [13] [14].

En el marco de Scrum, el ProductBacklog es una lista ordenada de todos los requerimientos y representa la única fuente de los mismos.

Cuando se usa Casos de Uso 2.0 las porciones son los ítems primarios del ProductBacklog, el uso de las porciones asegura que los ítems del mismo están bien construidos, y que naturalmente sean independientes, valorizables y testeables. La estructura del relato del Caso de Uso que los define, asegura que los mismos son estimables y negociables y que el mecanismo de corte permite obtener porciones de los Casos de Uso lo más pequeñas posible, tanto como sea necesario.

En este enfoque el ProductBacklog no está construido por adelantado, sino que es continuamente trabajado y refinado a través de las porciones de Casos de Uso.

Cabe destacar que las actividades que contribuyen al desarrollo metacognitivo se llevan adelante durante todo el proceso de producción de software razón por la cual su carácter es transversal, tal como se describe en la Figura 7:

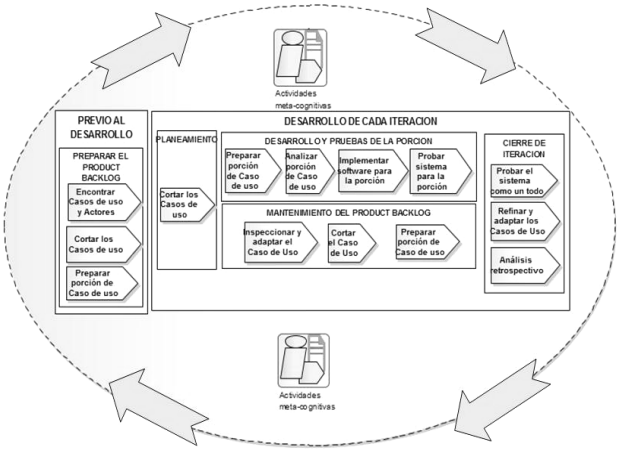

Figura 7. Secuencia de actividades en enfoque ágil

"Encontrar actores y Casos de Uso" se utiliza para construir el modelo inicial de Casos de Uso y ambientar el sistema.

"Cortar los Casos de Uso" es usado para crear el conjunto inicial más importante de porciones de Casos de Uso para alimentar el ProductBacklog.

"Preparar porción de Casos de Uso" es usado para conseguir uno o más de estos, listos para ser desarrollados en la primera iteración.

Una vez que el ProductBacklog está listo se puede comenzar el primer desarrollo iterativo.

Toda iteración comienza con algún planeamiento, durante este planeamiento puede ser necesario usar "Cortar el Caso de Uso" para promover porciones que sean lo

suficientemente pequeñas para completar en la iteración.
Para desarrollar los cortes identificados y agregarlos al sistema, se usa “Preparar porción de Casos de Uso”, “Analizar Casos de Uso, “Implementar software para la porción”, y “Probar el sistema para la porción”.
Mientras se ejecuta el desarrollo también se utiliza “Inspeccionar y adaptar el Caso de Uso”, “Cortar los Casos de Uso”, y “Preparar un corte de Caso de Uso” para mantener el ProductBacklog, manejar cambios, y asegurar que hay suficientes ítems listos para ser manejados en la próxima iteración.
Se debe “Probar el sistema como un todo” para ver el grado de avance logrado, e “Inspeccionar y adaptar el Caso de Uso” para analizar la calidad y la efectividad de los Casos de Uso y de las porciones implementadas. Al final de cada iteración se necesita hacer al análisis retrospectivo que permita definir la visibilidad del proyecto.
En el desarrollo de las actividades, el docente pondrá en ejecución acciones tendientes a estimular y facilitar el desarrollo del proceso metacognitivo, de tal manera que el sujeto que aprende pueda reflexionar sobre su propio aprendizaje, regular y organizar el conocimiento adquirido, identificar las destrezas aplicadas, generar nuevo conocimiento en base a los conocimientos previos y esbozar una medida de valor sobre su proceso de aprendizaje.
A tal fin proponemos una herramienta que asista a la autoevaluación en la cual el sujeto que aprende marcará qué capacidades y destrezas puso en juego en cada proceso del desarrollo de software al integrar Casos de Uso 2.0 con una metodología ágil, este cuadro estará asociado a una temporalidad acordada para que ambos, alumnos y docente puedan monitorear los avances del proceso de adquisición del conocimiento, tal como se describe en la Tabla 1:

Alumno: Fecha:	Encontrar actores y C. U.	Cortar C. U.	Preparar porción de C.U.	Analizar C.U.	Implementar software (porción)	Probar el sistema (porción)	Inspeccionar y adaptar C.U.	Probar el sistema todo	Análisis Retrospectivo	.....
Capacidades Cognitivas										
comprender										
relacionar										
interiorizar conceptos										
pensamiento										
sintetizar										
analizar										
razonamiento lógico										
sentido crítico										
percibir										
clasificar										
Destrezas										
vocabulario adecuado										
técnica búsqueda datos										
uso fuentes de información										
formular y comprobar conjeturas										
representación mental										
inferencias y deducciones										
rigor y precisión										
formulación adecuada y correcta										
buscar referencias										
exploración alternativas										
explicación causal										
expresión gráfica y lógica										
sentido de clasificación										
análisis-síntesis										

Tabla 1. Tabla de Identificación de Capacidades y Destrezas en Actividades Ingenieriles

Conclusión

La presentación de este nuevo abordaje para la captura de requerimientos basado en los Casos de Uso y su integración con una metodología ágil, está pensado como una prueba de concepto para que el sujeto que aprende a través del proceso de aprendizaje incorpore conceptos teóricos y de modelización de la Ingeniería de Requerimientos y la aplicación de las metodologías ágiles como una solución a la necesidad actual de la entrega temprana de código.
El proceso cognitivo si bien proporciona en este caso un conocimiento útil para una situación puntual, de no tener asociado un proceso metacognitivo no resulta en un aprendizaje adecuado que permita enfrentar nuevas situaciones reelaborando el saber. En este caso se han integrado actividades de la disciplina ingenieril de sistemas de información, con actividades de estimulación del pensamiento, el análisis, y la reflexión, a través del cambio de unos de los miembros del par integrador, en este

caso la metodología de desarrollo. Si el sujeto es capaz de realizar el proceso metacognitivo, éste se evidencia en las nuevas relaciones que pueda establecer en el caso modificado y haber generado otros conocimientos a partir del mismo.

La integración de dos tópicos como Casos de uso 2.0 y el Desarrollo Ágil de Software prevé el mapeo de conceptos de un enfoque hacia el otro, la vinculación de ideas, la correspondencia de artefactos y promueve la consolidación conceptual a través de una visión totalizadora, enriquecida por la inducción del proceso metacognitivo.

La puesta en ejecución de la presente propuesta, consideramos será fuente de motivación y contribuirá al proceso de conocimiento del sujeto que le permitirá autoevaluarse y optimizar el uso de los recursos con que cuenta, e ir en busca de aquellos deficitarios o en desarrollo. Asimismo sostenemos que la contribución excede el ámbito ingenieril contextualizado ya que consiste en una forma universal de aprender y ser.

Referencias

[1] Gibbs W. Wayt, Software's Chronics Crisis. (1994). Scientific American

[2] The Standish Group Report – Chaos. (1996)

[3] Allueva P., Desarrollo de habilidades metacognitivas: programa de intervención. Zaragoza, Consejería de Educación y Ciencia. Diputación general de Aragón. (2002).

[4] Ausubel, D. Teoría del Aprendizaje significativo extraído de http://www.docstoc.com/docs/20972313/La-Teoria-del-aprendizaje-significativo-de-David-Ausubel.

[5] Booch G., Jacobson I., Rumbaugh J. (2006). El lenguaje unificado de Modelado. Ed. Pearson Educación, 2da. Edición.

[6] González Fredy E. Acerca de la metacognición. (1996). Universidad Pedagógica Experimental Libertador, Venezuela.

[7] Pozo, J.I., Estrategias de aprendizaje, Desarrollo psicológico y educación, 1990 - Alianza Madrid

[8] Angulo Delgado, F. (2002). Aprender a enseñar Ciencias: Análisis de una propuesta para la formación inicial del profesorado de Secundaria, basada en la metacognición. Universidad Autónoma de Barcelona.

[9] Jacobson I., Spencer I., Bittner Kurt (2011). Use-Case 2.0 The Guide to Succeeding with Use Cases. Ivar Jacobson International.

[10] Jacobson I., Christerson M., Jonsson P., Overgaard G., Object Oriented Software Engineering: A Use Case Driven Approach Addison-Wesley Professional (1992)

[11] Jacobson I., Ng Pan-Wei, Aspect-Oriented Software Development with Use Cases Addison-Wesley Professional (2005)

[12] Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Martin Fowler, Jeff Sutherland et al. Manifesto for Agile Software Development

[13] Schwaber, K. Advanced Development Methods. SCRUM Development Process Retrieved (2010)

[14] Schwaber, K., Agile Project Management with Scrum, Microsoft Press (2004)

Datos de Contacto:

Ernesto R. Girbal, Universidad Tecnológica Nacional, Facultad Regional La Plata. Calle 60 y 124. La Plata. E-mail: ergirbal@yahoo.com.ar

Hugo Gardella, Universidad Tecnológica Nacional, Facultad Regional La Plata. Calle 60 y 124. La Plata. E-mail: hugarde2002@yahoo.com.ar

Mirta del Carmen Peñalva, Universidad Tecnológica Nacional, Facultad Regional La Plata. Calle 60 y 124. La Plata.
E-mail: mirpenalva@hotmail.com