

# Agile Framework for Software Maintenance: Application in Academic Management Domains

**Sonia I. Mariño, Pedro L. Alfonso**

Facultad de Ciencias Exactas y Naturales y Agrimensura.  
Universidad Nacional del Nordeste. 9 de julio 1449, Corrientes, Argentina  
simarinio@yahoo.com, plalfonso@hotmail.com

**Abstract**— Higher Education offers project management a variety of application domains oriented towards Knowledge Management. ICT devices constitute a potential tool to mediate Knowledge Management processes. The work proposes an agile framework based on SCRUM practices to manage the software maintenance process. The methodological proposal and its validation oriented to support the sub-management networks of a University Pedagogical Department are included.

**Keywords**— Knowledge Management; Maintenance in Software Engineering; Human Resources Management; Higher Education.

## 1. INTRODUCCIÓN

La expresión “Sociedad del Conocimiento” surge desde las teorías de generación del conocimiento (Senge, 2005; Nonaka y Takeuchi, 1999). Su abordaje se describe en numerosos trabajos (Perez Lindo et al., 2005; Krüger, 2006) y referencia a cambios en las áreas tecnológicas y económicas estrechamente relacionadas con las Tecnologías de la Información y Comunicación (TIC), en el ámbito de planificación de la educación y formación, en el ámbito de la organización (Gestión de Conocimiento) y del trabajo (de conocimiento) como se menciona en Davenport y Prusak (1998); Pérez Lindo, Ruiz Moreno, Varela, Grosso, Camós, Trottini, Burke y Darin (2005); Perez Lindo (2007); Perez Lindo (2007b); Del Canto, Araujo, Ferrer y Fernández, 2016). Entre sus características se mencionan el empleo intensivo de los sistemas de información y comunicaciones, el valor del conocimiento, la alfabetización en informática, el aprendizaje continuo y la organización electrónico-digital de la actividad humana (Perez Lindo et al., 2005; Bedoya Marrugo, Behaine Gómez, Severiche Sierra, Marrugo Ligardo y Castro Alfaro, 2018).

Una de las claves en este paradigma de conocimiento es como las organizaciones lo crean, procesan, difunden y manipulan. Por ello, es imprescindible asegurar la calidad de los datos, de la información generada y del conocimiento derivado para la toma de decisiones. Así, las Tecnologías de la Información y Comunicación (TIC) a través de sus diversas herramientas se constituyen en los medios que actúan en la captura, el almacenamiento, el procesamiento y la difusión de la información. En este sentido, se menciona a Ruiz Tapia, Martínez y Sánchez (2016, p. 29) quienes definen cómo las TIC configuran nuevas relaciones comunicacionales en la sociedad, en las universidades y en las actividades que ellas involucran.

Las Instituciones de Educación Superior (IES) deben tratar con este paradigma y maximizar las potencialidades y aplicaciones que ofrecen las TIC para su desarrollo. Por ello, funcionarios y gestores deben contemplar su empleo en actividades de educación, investigación, extensión y administración y además visualizar las alternativas de integración y difusión que otorga a su comunidad y a su ámbito de influencia.

### 1.1 Ingeniería del Software en el desarrollo de sistemas informáticos

Desde la Ingeniería del Software (IS) se promueve el uso de métodos y herramientas para lograr el cumplimiento de los objetivos organizacionales y apoyar la generación de información oportuna para la toma de decisiones.

La IS comprende los aspectos de la producción de software desde las etapas iniciales de la especificación de un sistema, hasta el mantenimiento de éste después de su implementación (Sommerville, 2015; Pigoski, 2015; April y Abran, 2008).

Asociaciones profesionales como la IEEE Computer Society (IEEE, 2015) y la Association for Computing Machinery (ACM, 2015) han aunado sus trabajos y desarrollaron una guía del Cuerpo de Conocimientos de la Ingeniería de Software o SWEBOK (Software Engineering Body of Knowledge).

La guía SWEBOK V3.0 (Bourque y Fairley, 2014) presenta actualizaciones en todas las áreas de conocimiento (KAS-por sus siglas en inglés-) a fin de reflejar los cambios de la Ingeniería de Software, previamente publicados en la guía SWEBOK 2004.

Bourque y Fairley (2014) en la guía SWEBOK V3.0 exponen 15 áreas del conocimiento, identificadas como: Requisitos de Software, Diseño de software, Construcción de software, Pruebas de software, Mantenimiento del software, Gestión de la Configuración de Software, Gestión de Ingeniería de Software, Proceso de Ingeniería de Software, Modelos y Métodos de Ingeniería de Software, Calidad del software, Práctica Profesional de la Ingeniería de Software, Economía de Ingeniería de Software, Fundamentos de Computación, Fundamentos matemáticos, Fundamentos de Ingeniería.

Este trabajo se enmarca en el área SWEBOK de “Proceso de Ingeniería de Software”, debido a que lo que se pretende es gestionar de la mejor manera los procesos de desarrollo de software, en este caso particular, la construcción de nuevas funcionalidades, comprendidas en el “Mantenimiento del software”.

Respecto al mantenimiento del software (MS), el objetivo es modificar el software existente conservando su integridad (Bourque y Fairley, 2014). El Instituto de Ingeniería Eléctrica y Electrónica (IEEE, 1993) define al mantenimiento del software como la modificación de un producto software, después de su entrega para corregir fallos, mejorar el rendimiento u otros atributos, o para adaptar el producto a otro entorno. Por ello, según April y Abran (2008); Polo, Piattini y Ruiz (2003); Sommerville (2015), el proceso de desarrollo del software continúa durante su tiempo de vida, dado que en la práctica sufren cambios para su permanencia y utilidad.

En SWEBOK se menciona que el proceso de mantenimiento, facilita las acciones necesarias y definidas para las entradas y salidas a esas actividades, las que se describen en las normas de mantenimiento del software como es la IEEE STD 1219 (1993) e ISO/IEC 14764 (2006).

Existen diferentes factores que inciden en la ejecución del Mantenimiento del Software. En IEEE 1219 (1993) se establecen los siguientes tipos: i) Mantenimiento adaptativo: modificación de un producto de software que se realiza después de su entrega para mantener un programa operativo mientras se cambia el entorno de producción; ii) Mantenimiento correctivo: modificación de un producto software realizado después de su entrega para corregir fallas encontradas; iii) Mantenimiento de emergencia: implica un mantenimiento correctivo no programado, ejecutado para mantener un sistema operacional; iv) Mantenimiento perfectivo: modificación de un producto software después de su entrega para mejorar su rendimiento o mantenibilidad. Además, en la ISO/IEC 14764 (2006) se considera el mantenimiento preventivo, el cual consiste en modificar un producto de software después de entregado para detectar y corregir fallas latentes antes de que se conviertan en fallas funcionales.

Ramírez, Reyes, Gil y Durgam (2015) exponen que las modificaciones pueden orientarse a: la adaptación del software a las necesidades específicas de los usuarios, su incorporación a desarrollos más amplios o a la corrección de errores. Estas entre otras alternativas vinculan el estudio sobre los desarrollos basados en la filosofía FLOSS o que referencia al software libre y de código con la investigación sobre el mantenimiento y la evolución.

Las organizaciones se esfuerzan para maximizar el rendimiento de su inversión en el desarrollo de software, manteniéndolo operativo el mayor tiempo posible (Bourque y Fairley, 2014). Desde la Ingeniería del Software y particularmente desde la evolución como parte integral del ciclo de vida, es relevante disponer de software operativo el mayor tiempo posible, maximizando las inversiones de las organizaciones. Sin embargo los desarrollos informáticos creados con la incorporación de los especialistas del dominio se debe ajustar y reconfigurar como consecuencia de diversas situaciones surgidas de su utilización, dándose constantes incrementos.

Mon y López Gil (2014, p. 2) mencionan que en los “proyectos de sistemas de información (SI) existen diversos modelos de proceso y de gestión que guían el ordenamiento y sistematización del desarrollo del software, así como la implantación y el mantenimiento de los productos software instalados”. En un contexto complejo al que debe responderse con celeridad ante los requerimientos de cambios, las metodologías ágiles surgen como una estrategia de adaptación y mejora continua.

## **1.2 SCRUM**

Uno de los métodos ágiles adoptados como práctica en la IS es SCRUM. Éste consiste en una colección de procesos para la gestión de proyectos. Su estrategia se centra en la entrega de valor para el cliente y la potenciación del equipo para lograr su máxima eficiencia, en un esquema de mejora continua (Schwaber, 1995; Sutherland, 2004; Sutherland, Viktorov y Blount, 2007; Deemer, Benefield, Larman y Vodde, 2009; Pérez, 2011). Como método ágil se caracteriza por:

- Ser un modo de desarrollo adaptable, antes que predictivo.
- Estar orientado a las personas, más que a los procesos.
- Emplear el modelo de construcción incremental basado en iteraciones y revisiones.

Las prácticas empleadas por SCRUM para mantener un control ágil en el proyecto son: i) Revisión de las iteraciones, ii) Desarrollo incremental, iii) Desarrollo evolutivo, iv) Auto-organización del equipo y v) Colaboración. Estas prácticas pueden presentarse en distintos niveles de desarrollo siendo fundamental el involucramiento de los sujetos que participan en el proyecto.

Se optó por SCRUM como base de la propuesta, dado que esta metodología proporciona avances parciales y regulados del producto final en función de las necesidades del usuario, facilita la elaboración de proyectos con resultados de forma inmediata. Sutherland (2012) señalan que SCRUM puede ser desarrollada en tres agrupaciones:

- ProductOwner(PO), dueño del producto, es la voz del cliente y responsable de generar tareas a desarrollar en base a sus requerimientos o necesidades;

- Scrum Master (SM) es el líder del proyecto y el responsable de evitar cualquier inconveniente que el equipo de desarrollo pueda encontrar; y
- DevelopmentTeamMembers(DTM), miembros del equipo, encargados del desarrollo de la aplicación.

Además, Sutherland (2012) afirma que el corazón de la metodología SCRUM es el Sprint. Cada Sprint se puede considerar como un proyecto en que se especifica el artefacto a construir, el diseño y un plan flexible que guía el trabajo a realizar.

### 1.3 Contexto de Educación Superior en que se desarrolla la propuesta

Bedoya Marrugo et al. (2018) establecen algunos considerandos de la importancia de redes y ejemplifican en el ámbito de la Educación Superior. Además, en el éxito de un proyecto de gestión del conocimiento “es fundamental observar, interpretar y entender el funcionamiento de las organizaciones” (Del Canto et al., 2016, p. 167).

La presente propuesta metodológica promoverá el intercambio de información entre diversas unidades académicas utilizando herramientas software con información referente a la Gestión Departamental. La captura y procesamiento de estos datos se constituirá en una fuente de centralización de documentación e insumo para el monitoreo de la carrera.

La Red de Gestión de Recursos Humanos (RRHH) propuesta se enmarca en lo expuesto, dado que las TIC son las herramientas de las que se vale la gestión del conocimiento y sus procesos de administración transforman los datos en información valiosa.

El trabajo expone actividades concernientes al mantenimiento del software ejecutadas por un equipo de docentes-gestores, desarrolladores y validadores con la finalidad de lograr un sistema informático de apoyo a la producción de información confiable y oportuna para la toma de decisiones.

Por otra parte, se propone un marco de trabajo que integra un método ágil como es SCRUM con aspectos del mantenimiento del software. La propuesta se diferencia de otras dado que - sustentada en la prácticas de SCRUM - asocia a cada lista de requerimientos (Product-Backlog) una Sub-Red de Conocimiento (Figura 1).

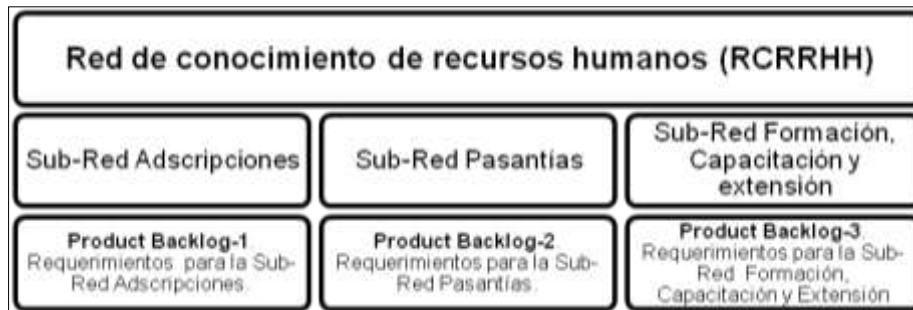


Figura 1. Red de conocimiento de recursos humanos (Fuente: elaboración propia).

Mon y López Gil (2014, p. 2) establecen como estos modelos “dividen los procesos en sub-procesos, definen el conjunto de actividades que deben llevarse a cabo en el desarrollo, así como las relaciones entre estas y los productos de salida de cada una”. Es decir, cada Product Baklog de SCRUM representa una sub-red de conocimiento, que desde la IS se encuentra representada por cada uno de los módulos informáticos disponibles. Por ello en su definición se involucran recursos humanos que abordan el análisis, diseño, desarrollo e implementación informática.

## 2. METODOLOGÍA

La metodología adoptada en este trabajo consta de fases, las que se describen a continuación:

Fase 1: Estudio y selección de temas de la disciplina informática para sustentar la propuesta. Particularmente se optó por SCRUM como metodología ágil y el Mantenimiento del software como un aspecto fundamental desde la Ingeniería del Software. Con la finalidad de lograr un encuadre conceptual se situó al Mantenimiento del Software desde la guía SWEBOK.

Fase 2: Elaboración de un marco de trabajo ágil sustentado en la integración de prácticas de SCRUM y el mantenimiento del software. El enfoque teórico que sustenta la propuesta metodológica, un producto de este trabajo, se presenta en las secciones 1.1 y 1.2.

Fase 3: Elección de un contexto de validación en dominios de la administración en la educación superior.

Fase 4: Implementación de la propuesta en el contexto de la educación superior:

- Búsqueda y compilación de fuentes de datos para sistematizar y organizar la Red de Conocimiento de Recursos Humanos (RCRRHH). Se consultaron a distintas áreas de la Unidad Académica a la cual se transferirá e implementará la propuesta. Se optó como objeto de validación la información producida por el Departamento de Informática en el periodo 2017/2018.
- Aplicación del marco de trabajo ágil propuesto para la definición y desarrollo de un conjunto de herramientas de aporte a la construcción de conocimientos de un departamento pedagógico de una Unidad Académica en el cual los autores del trabajo se desempeñan. Se optó como objeto de validación la información producida por el Departamento de Informática en el periodo 2017/2018.
- Aplicación del mantenimiento perfectivo a la RCRRHH con la finalidad de incrementar su funcionalidad con la Sub-Red Adscripciones, Sub-Red Pasantías y Sub-Red Formación, Capacitación y Extensión (Figura 1), considerando los requerimientos de información para la toma de decisiones.

### 3. RESULTADOS

En esta sección se presentan los resultados exponiendo primeramente el marco de trabajo ágil propuesto, a continuación, se ejemplifica con un caso de estudio en el contexto de la educación superior.

#### 3.1 MARCO DE TRABAJO ÁGIL PROPUESTO

La propuesta de integración de prácticas de SCRUM en el mantenimiento del software, se sustenta en la mejora de cada una de las Sub-Redes de Conocimiento. Estas se entienden como subsistemas de apoyo a la gestión administrativa focalizados en ciertos aspectos de la organización con miras a lograr aspectos competitivos para la toma de decisiones.

La Figura 2 ilustra el marco de trabajo propuesto que integra las prácticas y artefactos de SCRUM, mencionados anteriormente, y asociados a las actividades propuestas por el estándar IEEE 1219 (1993). Como se visualiza, el proceso de iteración del Sprint se representa en las tareas que componen la elaboración de cada versión del prototipo de software. En cada Sprint se promueven evaluaciones que generan datos de realimentación, no se pueden cambiar los elementos elegidos. Al finalizar, el equipo obtiene comentarios y observaciones que se pueden incorporar al siguiente Sprint.

SCRUM pone el énfasis en productos que funcionen al final del Sprint, es decir aquellos realizados (Deemer et al., 2009; Palacio y Ruata, 2009). De este modo al realizar otro incremento, se avanza en la generación de la solución requerida.

En la Figura 3 se visualiza el proceso de mantenimiento asociado a una Sub-red y la aplicación de las fases propuestas por el estándar IEEE 1219 (1993), documentándose en cada una de ellas los cambios realizados. En la Tabla 1, se presentan las fases y actividades asociadas a la Sub-red según el estándar IEEE 1219 (1993).

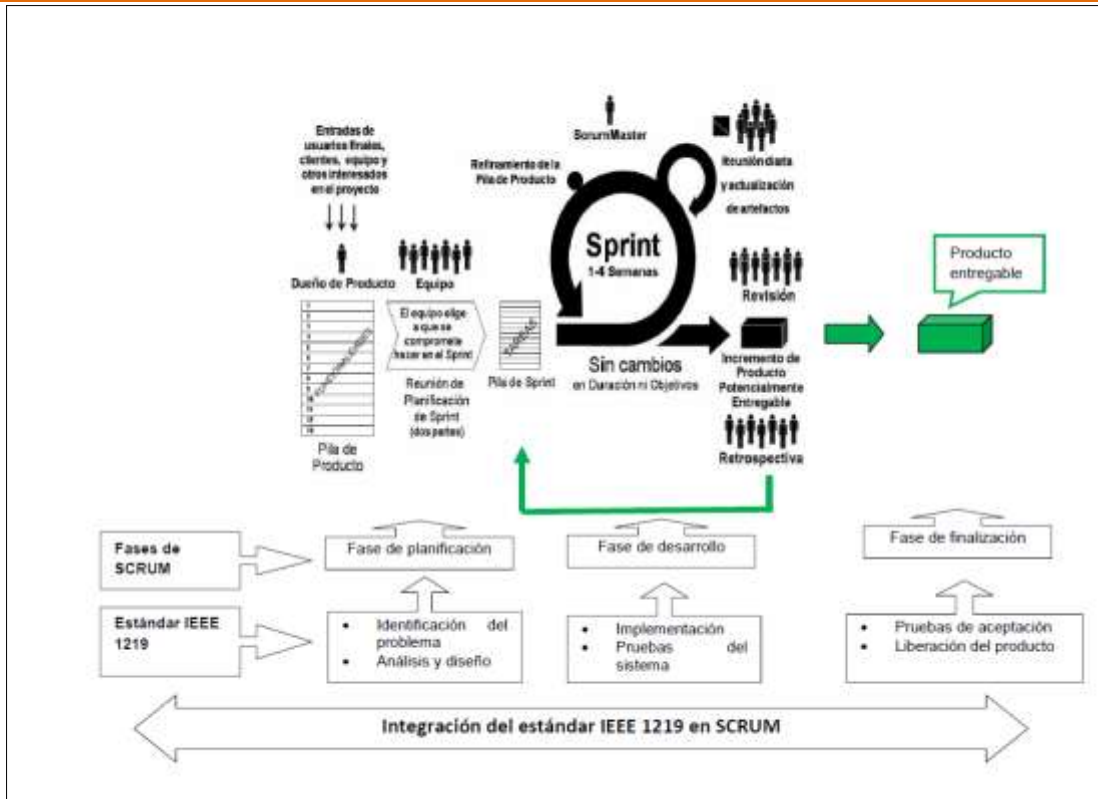


Figura 2. Marco de trabajo ágil propuesto para gestionar el proceso de mantenimiento de cada Sub-Red (Fuente: elaboración propia adaptada de Deemer et al., 2009).

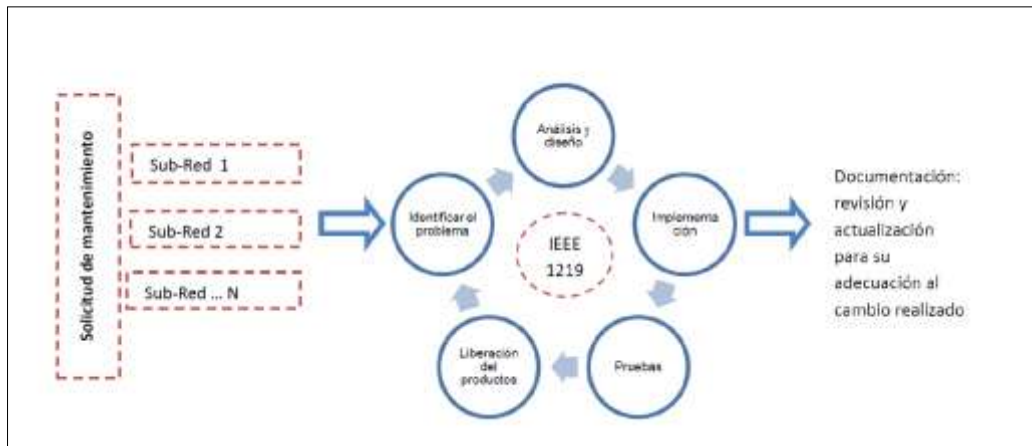


Figura 3. Proceso de solicitud de modificación (Fuente: elaboración propia).

Tabla 1. Fases y actividades relacionadas al estándar IEEE 1219 (1993)

Fases	Actividades
Identificación del problema	<ul style="list-style-type: none"> <li>Identificar el problema. Se planteó la necesidad de diseñar interfaces que faciliten desplegar información específica.</li> <li>Asignar prioridad. Se estableció como primacía la redefinición de la interfaz inicial y la localización de información específica.</li> <li>Obtener aprobación de la solicitud de modificación y las tareas a llevar a cabo. Se analizó la</li> </ul>

Fases	Actividades
	<p>solicitud y se procedió a su aceptación.</p> <ul style="list-style-type: none"> <li>Estimar inicialmente los recursos necesarios para modificar el sistema existente. Básicamente se asociaron a la disponibilidad de conexión a la web para el despliegue de las modificaciones realizadas.</li> </ul>
Análisis	<p>Se identifican los elementos a modificar. A través de prototipos se plantean soluciones alternativas y como estos cambios impactan en la información contenida en la base de datos y en el funcionamiento del sistema. Se procede a:</p> <ul style="list-style-type: none"> <li>Evaluar el impacto.</li> <li>Evaluar los costos.</li> <li>Estudiar la viabilidad y el alcance de las modificaciones.</li> <li>Desarrollar un plan preliminar de diseño, implementación, pruebas y liberación del software.</li> <li>Desarrollar estrategias de pruebas</li> </ul>
Diseño	<p>En esta fase se diseñan los elementos de interfaces que facilitan la selección de información contenida en la base de datos. A fin de validar las modificaciones realizadas se generan y ejecutan las pruebas previamente planteadas.</p>
Implementación	<p>Las fases consistieron en:</p> <ul style="list-style-type: none"> <li>Desarrollar y probar las modificaciones realizadas.</li> <li>Codificar y generar pruebas unitarias.</li> <li>Integrar el software modificado con el sistema existente.</li> <li>Analizar el riesgo.</li> <li>Revisar la preparación para las pruebas</li> </ul>
Pruebas del Sistema	<p>Aborda proceder a :</p> <ul style="list-style-type: none"> <li>Realizar pruebas sobre el sistema modificado</li> <li>Revisar integridad.</li> <li>Obtener aprobación</li> </ul>
Pruebas de Aceptación	<ul style="list-style-type: none"> <li>Realizar pruebas sobre el sistema completamente integrado</li> <li>Pruebas desarrolladas incrementalmente y en paralelo al mantenimiento perfecto aplicado en el módulo seleccionado.</li> </ul>
Liberación del Producto	<ul style="list-style-type: none"> <li>Desarrollar un plan.</li> <li>Notificar a los usuarios.</li> <li>Realizar una copia de seguridad de la versión del sistema.</li> <li>Instalar y capacitar a los usuarios.</li> <li>En proceso, se elabora un plan destinado a la liberación del producto. En referencia a su difusión, se entiende este trabajo como un producto derivado de actividades de I+D. Se planifican capacitaciones instrumentales relacionadas con el mantenimiento del código de cada Sub-Red y con el uso para asegurar su implantación organizacional.</li> </ul>

A continuación se especifican las prácticas de SCRUM asociadas a cada Sub-Red (Figura 1).

- Product Backlog. Requerimientos priorizados de cada Sub-Red atendiendo la propuesta elaborada.
- Product Owner y Scrum Master. Responsable de cada Sub-Red. Con la finalidad de mejorar los tiempos de desarrollo, cada Sub-Red se genera en paralelo. El responsable define el Product Backlog, la priorización de éstos y la aceptación del producto a entregar. Se verifican el cumplimiento de los valores y principios de SCRUM.
- Development Team Members (equipo). Se conforman distintos equipos de desarrollo atendiendo a cada Sub-Red.
- Se gestiona el riesgo en forma continua, a través de las reuniones de revisión y retrospectiva.
- Se prioriza, estima y define el alcance de cada versión del prototipo a través de la pila de productos.
- Se realizan reuniones de planificación del Sprint a partir del Product Backlog, en éstas participan el Product Owner y Scrum Master quienes priorizan las tareas a incluir en el Sprint Backlog.
- Se organiza el Sprint Backlog, con las tareas seleccionadas del Product Backlog.
- Se gestiona el Sprint Backlog. Al inicio de cada iteración se seleccionan los requerimientos y se estima el esfuerzo de

cada tarea, para elaborar la versión del prototipo software. La duración se establece en 4 semanas.

- Se realizan reuniones de planificación de actividades, diferenciándose en aquellas inherentes al análisis, diseño, desarrollo, validación y documentación.
- Se realizan revisiones (Sprint Review). Al finalizar el Sprint el DTM, presenta la versión del prototipo generado a los participantes del proyecto para identificar falencias o mejoras en aquellas Sub-Red de las cuales no participaron como desarrolladores. Asisten todos los involucrados en el proyecto: Product Owner, Scrum Master y el Development Team Members.
- Se diseñan gráficos de Burn-down, para visualizar y gestionar el avance de las tareas y del Sprint.

Al finalizar el proceso de desarrollo, es decir, cuando el Product Backlog carezca de requerimientos, se continuará con la fase de finalización (Figura 2), realizando las actividades descritas en la Tabla 1.

### 3.2 Propuesta de validación en un contexto de educación superior

Mon, Estayno y Scalzone (2016, p. 232) establecen que “la característica principal del proceso seleccionado, es que está basado en un ciclo de vida de proyectos iterativo e incremental. Esto es ideal para mejorar la productividad del área, sobre todo cuando se cuenta con recursos humanos escasos para el volumen de trabajo”. Por ello se decidió asignar a cada sub-red al responsable administrativo de la IES, quien asume el rol de Product Owner.

En un contexto de la Educación Superior, conceptualizar las Sub-Redes de Conocimiento se fundamenta en la normativa existente, datos registrados en planillas electrónicas, documentos y el conocimiento tácito de los recursos humanos que intervienen en cada uno de los procesos administrativos susceptibles de gestionar en cada una de ellas.

#### 3.2.1 Antecedentes de software asociados a las Sub-redes

Se mencionan algunas generalidades definidas para los artefactos software asociados a cada Sub-Red de Conocimiento. La definición de requerimientos derivó en especificaciones funcionales determinándose la necesidad de disponer de los siguientes perfiles: administrador del sistema, operador, gestores, usuarios. El administrador es responsable de la gestión de usuarios, tablas auxiliares y el mantenimiento y seguridad del sistema en general.

Cada artefacto software –que representa computacionalmente una Sub-Red de Conocimiento- posibilita consignar los datos necesarios para realizar: ingresos, modificaciones y eliminaciones; así como el seguimiento de cada expediente (estado y ubicación) desde su inicio hasta su finalización. El marco de trabajo asociado a la Sub-Red Adscripciones permite la interacción con diferentes roles de usuarios según su perfil, y garantiza acceso seguro a la información.

Se define como "pasantía educativa" al conjunto de actividades formativas que realicen los estudiantes en empresas y organismos públicos, o empresas privadas con personería jurídica, sustantivamente relacionado con la propuesta curricular de los estudios cursados en unidades educativas, que se reconoce como experiencia de alto valor pedagógico, sin carácter obligatorio (Ley 26.427, 2008).

El marco de trabajo asociado a la Sub-Red Pasantía proporciona el acceso a las siguientes categorías de información: Alarmas de presentación de informes; Nómina de pasantes aplicando diversos criterios como por ejemplo discriminados según departamento, área, asignatura, categoría, en un periodo de tiempo o con informes aprobados; nómina de pasantías de un alumno dado que ofrecerá información de la formación disciplinar del recurso humano; nómina de pasantías de un director que podría reflejarse en el aporte docente a la institución dado que es menester la conformación de redes de conocimiento; nómina de pasantías por organización oferente.

Se entiende por adscripción, la designación para el desarrollo de una actividad docente, de investigación o de extensión en la categoría de Auxiliares Docentes, sin que ello genere derecho a remuneración alguna, debiendo el Adscripto cumplir con las obligaciones que se establecen en la presente reglamentación (Resol. 1286, 2013).

El marco de trabajo asociado a la Sub-Red Adscripciones es similar a la Sub-Red Pasantía. El despliegue de información varía en ofrecer datos referentes a los adscriptos quienes se forman en el interior de la facultad.

El marco de trabajo asociado a la Sub-Red Capacitación y Formación de Recursos Humanos (CITA), centra su interés en el modelado y gestión de las siguientes categorías: Proyectos voluntariados, Proyectos universitarios en el Medio, Proyectos de transferencia, Seminarios, Cursos, Talleres, Workshop, Congresos. Igualmente se generan reportes según diversos criterios entre los que se mencionan: periodos de tiempo, categorías de acciones de Capacitación y Formación de Recursos Humanos, fortaleciendo distintas interacciones desde la Academia hacia su contexto.

### 3.2.2 Implementación de la propuesta.

La propuesta metodológica tuvo como punto de partida reuniones de planificación. Éstas permitieron determinar la complejidad del sistema Red de conocimiento de recursos humanos. Se consideran como antecedentes las Sub-Red Adscripciones, Sub-Red Pasantías y Sub-Red Formación, Capacitación y Extensión descritas en (Godoy et al., 2017; FaCENA, 2013) (Figura 1), cuya implementación definitiva impactará positivamente en las funciones del departamento pedagógico y así en la Unidad Académica y posiblemente con trascendencia hacia la universidad. Por ello, asociada a cada Sub-Red conceptual descrita se disponen de productos funcionales que pueden operar en forma independiente con posibilidad de integración en un sistema de mayor complejidad y alcance. Se optó como caso de estudio un departamento pedagógico de una Unidad Académica.

La Figura 3 muestra como se aplican las actividades definidas en el estándar IEEE 1219 (1993) para lograr el mantenimiento perfectivo de cada Sub-Red descrita anteriormente. En este sentido se mejoraron la calidad de los registros de datos según se explicitan en la documentación que sustentan las actividades de cada una de ellas.

En referencia a la validación de la propuesta, ésta se realizó en dos instancias:

- La primera validación se ejecutó al finalizar cada Sprint, asumiendo como Product Owner el Responsable de cada Sub-Red. Lo expuesto permitió poner en funcionamiento y liberar la propuesta de mantenimiento perfectivo basada en SCRUM y aplicado al sistema Red de conocimiento de recursos humanos, compuesto por la Sub-Red Adscripciones, la Sub-Red Pasantías y la Sub-Red Formación, Capacitación y Extensión (Figura 1), considerando los requerimientos de información para la toma de decisiones en el departamento, y previamente mencionadas en la sección anterior.

En este sentido, para la Sub-Red Adscripciones y la Sub-Red Formación, Capacitación y Extensión, se recopilaron y registraron información correspondiente al año 2018. La carga de datos permitió subsanar algunas cuestiones inherentes a la consistencia y disponibilidad de información emanada de las resoluciones o instrumentos académicos.

Respecto a la Sub-red Pasantías, la validación se realizó con datos generados en el Departamento de Informática con la finalidad de capturar y registrar esta información. Además, con fines de lograr –en el mediano plazo- la captura y registro de información de las pasantías gestionadas con la unidad académica se solicitó a una secretaria de la unidad académica la correspondiente validación del sistema. Es así, como se detectaron algunas cuestiones que mejorarían las funcionalidades implementadas. Por ejemplo, se requiere disponer de la base de datos de docentes y alumnos para recuperar la información registrada en otros sistemas informáticos y así mejorar la carga. Este mantenimiento perfectivo implicaría realizar otros trámites administrativos que sostenga la propuesta de mejora informática.

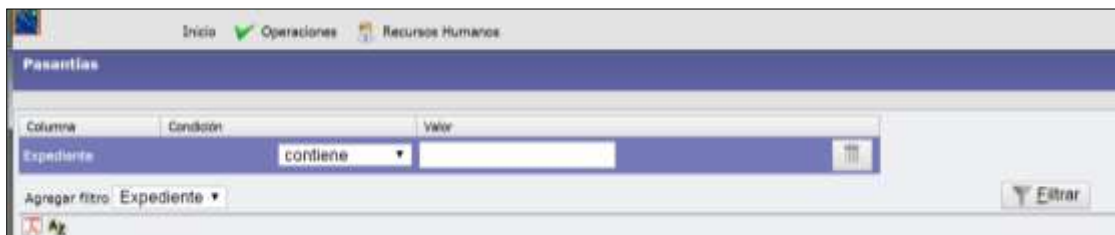
- La segunda instancia de validación consistió en la implementación de las Sub-Redes para su prueba en un ambiente real, dado que estos módulos se encuentran instalados en un servidor de la facultad. En este sentido, con la finalidad de lograr el uso de la herramienta se liberaron el acceso a los módulos de las diferentes Sub-redes a los efectos que distintas áreas de la unidad académica, en donde se genera la información iniciada por el departamento: adscripciones, pasantías, y los datos referentes a temas de Formación, Capacitación y Extensión sea registrada por el personal designado.

Según sean los destinatarios y tomadores de decisiones de las distintas Sub-Redes, es menester la incorporación de filtros y reportes. A modo de ejemplo, las Figuras 4, 5, 6 y 7 muestran la información producida desde las tres Sub-Redes para su despliegue en los reportes académicos. Continuando la ilustración, la información producida permitiría detectar sujetos con conocimientos disciplinares específicos o con formación en un área determinada, lo que aportará a la vinculación desde la Universidad con áreas del Gobierno y Empresas.

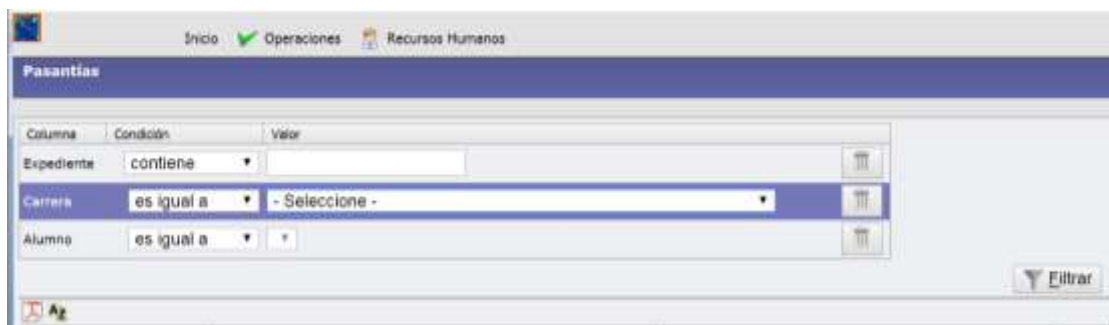




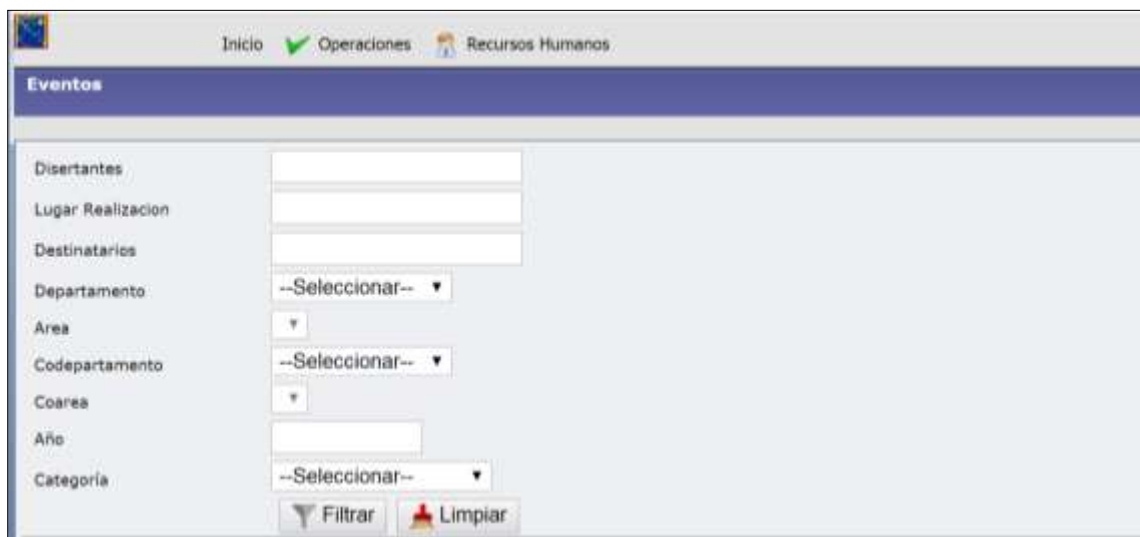
**Figura 4.** Filtros de la Su-Red Adscripciones (Fuente: elaboración propia).



**Figura 5.** Filtros por “Expedientes” de la Sub-Red Pasantía (Fuente: elaboración propia).



**Figura 6.** Filtros por “Carrera” de la Sub-Red Pasantía (Fuente: elaboración propia).



**Figura 7.** Filtros de la Sub-Red Pasantía (Fuente: elaboración propia).

#### 4. CONCLUSIONES

El mantenimiento del software es un proceso clave para lograr la implantación de sistemas de información en contextos organizacionales complejos y diversos, caracterizados por cambios constantes. Uno de estos contextos es el que atañe a las IES que, además de centrar su atención en los procesos de enseñanza y aprendizaje, deben abordar los procesos administrativos mediados por artefactos software.

Se expuso un marco de trabajo ágil que integra en el proceso del mantenimiento del software practicas sustentadas en SCRUM. Su validación en un contexto específico de IES, generó información de retroalimentación con miras a lograr mejoras en las prácticas desarrolladas.

Empíricamente se prevé la definición de nuevos Product Backlog, orientados a la mejora de los software que componen el Sistema red de conocimiento de recursos humanos; entre los que se mencionan: interoperabilidad, escalabilidad, acceso desde dispositivos móviles.

Por lo expuesto, la implementación se realiza gradualmente interactuando con distintos actores de la comunidad académica: gestores universitarios, docentes, alumnos y otros sujetos quienes aportan sus visiones e interaccionan en procesos decisorios; posibilitando la definición y adecuación del Sistema red de conocimiento de recursos humanos.

Cada producto software definido para cada una de las Sub-Redes mencionadas, permite afrontar la sistematización y la generación de indicadores de apoyo a la toma de decisiones. Así, se podrían adaptar a otros contextos de administración en Educación Superior y generar estudios comparativos, longitudinales y territoriales que aportarían a la definición de estrategias superadoras en contextos de las Redes de Conocimientos.

Es así como las actividades de mantenimiento perfectivo logradas por la integración de un enfoque ágil en el mantenimiento del software, se reflejaron en un software que integra aspectos conceptuales de las tres Sub-Redes de gestión de la información propuestas para apoyar la toma de decisiones en espacios de Educación Superior. Es decir, su utilización producirá valiosa información orientada a distintos actores del quehacer educativo y de otros ámbitos de la sociedad en la cual la Universidad tiene injerencia.

## 5. REFERENCES

1. ACM. (2015). Association for Computing Machinery, <https://www.acm.org/>
2. April, A. y Abran, A. (2008). Software Maintenance Management: Evaluation and Continuous Improvement. IEEE Computer Society
3. Bedoya Marrugo, E., Behaine Gómez, B., Severiche Sierra, C. A., MarrugoLigardo, Y. y Castro Alfaro, A. (2018). Redes de Conocimiento: Academia, Empresa y Estado. Espacios, 39(8).
4. Bourque, P. y Fairley, R. E. (2014). SWEBOK Guide to the Software Engineering Body of Knowledge, Version 3.0. IEEE Computer Society, [www.swebok.org](http://www.swebok.org)
5. Davenport, T. y Prusak, L. (1998) Working Knowledge. Harvard Business School Press, Boston, Massachusetts.
6. Deemer, P., Benefield, G., Larman, C. y Vodde, B (2009). Información Básica de ScrumtheScrum Primer Version 1.1. Scrum Training Institute. Traducción de Leo Antoli. Agile, Spain, [http://www.goodagile.com/scrumprimer/scrumprimer\\_es.pdf](http://www.goodagile.com/scrumprimer/scrumprimer_es.pdf).
7. Del Canto, E., Araujo, E., Ferrer, Y. y Fernandes, V. (2016). Gestión del conocimiento y TICS como generador de éxito en las organizaciones, Ingeniería y Sociedad UC. 11(2), pp. 163-173.
8. IEEE (2015). Computer Society, <https://www.acm.org/>
9. IEEE STD 1219:1993. Standard for Software Maintenance. IEEE Computer Society Press. USA
10. FaCENA (2013). Reglamento para el Régimen de Adscripciones, Resolución N° 1286 /13, FACENA, Universidad Nacional del Nordeste.
11. Godoy, M. V., Mariño, S. I., Alfonzo, P. L., Cáceres, M. y Alderete, R. (2017). Sub-redes de gestión y sus herramientas como apoyo en un Departamento Pedagógico Universitario. XIX Workshop de Investigadores en Ciencias de la Computación (WICC 2017), Argentina.
12. ISO/IEC Standard 14764: 2006. International Organization for Standardization. Software Engineering – Software Life Cycle Processes - Maintenance
13. Krüger, K. (2006). El concepto de Sociedad del Conocimiento. Biblio 3W, Revista Bibliográfica de Geografía y Ciencias Sociales, 9(683), <http://www.ub.edu/geocrit/b3w-683.htm>.
14. Ley 26.427. (2008). Creación del sistema de pasantías educativas en el marco del sistema educativo nacional. Publicación en Boletín Oficial: 22/12/2008
15. Mon, A., Estayno, M. y Scalzone, P. (2006). Definición de un marco de trabajo para la implementación inicial de un Modelo de Proceso Software. Aplicación de un caso de Estudio. III Workshop de Ingeniería de Software y Bases de Datos, pp. 230-240, [http://sedici.unlp.edu.ar/bitstream/handle/10915/22121/Documento\\_completo.PDF?sequence=1](http://sedici.unlp.edu.ar/bitstream/handle/10915/22121/Documento_completo.PDF?sequence=1)
16. Mon, A. y López Gil, F. (2014). Guía para la Implantación de Sistemas. XX Congreso de Ciencias de la Computación, Argentina.
17. Nonaka, I. y Takeuchi, H. (1999). The knowledge creating company. Harvard Business Review, 69 (6), pp. 96-104.
18. Palacio, J. y Ruata, C. (2009). Scrum Manager: Gestión de proyectos, [http://www.scrummanager.net/files/sm\\_proyecto.pdf](http://www.scrummanager.net/files/sm_proyecto.pdf)
19. Pérez, O. (2011). Cuatro enfoques metodológicos para el desarrollo de SoftwareRUP – MSF – XP – SCRUM, Inventum No. 10, pp. 64-78.
20. Pérez Lindo A., Ruiz Moreno, L. Varela, C., Grosso, F., Camós, C., Trottini, A, M, Burke, M. L. y Darin, S. (2005). Gestión del conocimiento. Un nuevo enfoque aplicable a las organizaciones y la universidad. Grupo Editorial Norma. Bs. As.
21. Pérez Lindo, A. (2007). De la revolución cognitiva a la gestión del conocimiento. Documento presentado en el curso de postgrado “Gestión del Conocimiento”. Doctorado de Ciencias Cognitivas. Facultad de Humanidades. Universidad Nacional del Nordeste, Inédito.
22. Pérez Lindo, A. (2007b). El Gobierno inteligente en la República Universitaria. De Platón a la Gestión del Conocimiento. Documento presentado en el curso de postgrado “Gestión del Conocimiento”. Doctorado de Ciencias Cognitivas. Facultad de Humanidades. Universidad Nacional del Nordeste, Inédito.
23. Pigoski, T M. (2015). SWEBOK Knowledge Area Description for Software Evolution and Maintenance, <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.573.5287&rep=rep1&type=pdf>.
24. Polo, M., Piattini, M. y Ruiz, F. (2003). Advances in Software Maintenance Management: Technologies and Solutions. Ed. Idea Group Inc
25. Ramírez J., Reyes C., Gil G., y Durgam, F. (2015). Evolución y reusabilidad en FLOSS. In XVII Workshop de Investigadores en Ciencias de la Computación. Argentina.

26. REDUNCI. (2014). Red de Carreras de Grado en Informática. Documento de recomendaciones curriculares, <http://redunci.info.unlp.edu.ar/docs/Documento%20Curricular%20RedUNCI%20Abril%202015.pdf>
27. Ruiz Tapia, J. A., Martínez, M., Sánchez. (2016). El impacto de las TICs en la calidad de la educación superior, *Revista de Investigación en Ciencias Contables y Administrativas*, 1(1), pp. 28, 44.
28. Senge, P. (2005). *La Quinta Disciplina: el arte y la práctica de la organización abierta al aprendizaje*, Buenos Aires: Ed. Granica 2da edición.
29. Schwaber, K. (1995). Scrum Development Process, in *OOPSLA Business Object Design and Implementation Workshop*, J. Sutherland, D. Patel, C. Casanave, J. Miller, and G. Hollowell, Eds. London: Springer.
30. Sutherland, J., Viktorov, A. y Blount, J. (2007). Distributed Scrum: Agile Project Management with Outsourced Development Teams, *hicss*, pp.274a, 40th Annual Hawaii International Conference on System Sciences (HICSS'07).
31. Sutherland, J. (2004). Agile development: Lessons learned from the first scrum, <https://pdfs.semanticscholar.org/8f37/973e26b11fc1120ad1291df2c83df67704b6.pdf>
32. Sutherland, J., (2012). *The Scrum Papers: Nut, Bolts, and Origins of an Agile Framework*, Version 1.1 –2 Apr 2012, Scrum, Inc., <http://jeffsutherland.org/scrum/ScrumPapers.pdf>
33. Sommerville, I. (2015). *Ingeniería del Software*. Ed. Prentice Hall, 10 ed.