

Proceso de gestión de Trabajos Fin de Carrera

Carlos López, Raúl Marticorena, Juan José Rodríguez y Andrés Bustillo

Área de Lenguajes y Sistemas Informáticos
Universidad de Burgos
Escuela Politécnica Superior Edf. C
09006 Burgos, España
{clopezno, rmartico, jjodriguez, abustillo}@ubu.es

Resumen

El desarrollo de trabajos final de carrera en las actuales titulaciones de informática es una de las materias fundamentales a la conclusión de los estudios. En las propuestas de grado y máster este punto sigue siendo fundamental, no existiendo apenas discusión en la necesidad de realizar un proyecto donde se aglutinen todos los conocimientos y competencias, que han sido adquiridos a lo largo de los estudios realizados. En este trabajo, se propone un modelo de gestión de la asignatura dentro de un contexto universitario, aplicando buenas prácticas de ingeniería del software: monitorización, control de proceso, revisiones, evaluación y medición. A partir de dicho modelo se describe una herramienta que lo automatiza, y se muestran dos casos de estudio resultado de su aplicación. El primero muestra los productos obtenidos de la aplicación del proceso. El segundo compara los resultados al utilizar la herramienta de evaluación entre trabajos de dos asignaturas distintas de proyectos: la de ingeniería y la de ingeniería técnica en informática.

1. Introducción

Tanto en los planes de estudio de informática actuales como en las futuras titulaciones de grado y máster, existen asignaturas relacionadas con el desarrollo de un proyecto informático, generalmente conocidas como trabajo final de carrera. Independientemente de la Universidad donde se imparta, la propia naturaleza de la asignatura tiene un conjunto de características que la distinguen del resto: alto grado de profesorado involucrado, gran cantidad de créditos/trabajo para los alumnos, tutorías personalizadas, contenidos personalizados a

cada proyecto, gran cantidad de información que gestionar, normas institucionales especiales, marco de relaciones con empresa, etc. Para poder incidir sobre estas características, a través de planes de mejora, se debe definir un proceso de gestión y aportar un conjunto de indicadores, métricas de proceso y métricas de producto.

La gestión de proyectos es un tema ampliamente tratado en ingeniería del software tanto desde un punto de vista industrial [11] [12][4], como desde un punto de vista docente [5]. En los procesos de gestión de proyectos existen gran cantidad de actividades: planificación, gestión de configuración, monitorización, control, revisiones, evaluación y medición [4]. La propia asignatura de trabajo final de carrera se convierte en un marco de enseñanza práctica de algunas de las actividades, tanto para alumnos como para tutores involucrados [3].

En un contexto universitario, existe un gran número de proyectos, con gran cantidad de información a ser procesada, contenida en sus documentos asociados, y poco tiempo de trabajo asignado a los tutores involucrados. El contexto de desarrollo de proyectos universitario, a nivel de técnicas y herramientas utilizadas, es muy heterogéneo, imposibilitando la recogida automática de información. Esta heterogeneidad es una de las grandes fortalezas de la asignatura, la dota de un alto grado de innovación tanto a nivel de herramientas como de técnicas. Pero a su vez, es una de sus debilidades, en el sentido que dificulta la gestión conjunta de algunas de las actividades expuestas desde un punto industrial. Esta situación no debe llevar al desánimo en el profesorado a la hora de aplicar buenas prácticas de la ingeniería del software adaptadas al tiempo disponible. Por otro lado, puede provocar en el alumnado una visión

distante del concepto de gestión de proyectos entre los contextos más productivos, como los industriales, y más experimentales como los universitarios.

Con este trabajo se pretende avanzar en el proceso de gestión de proyectos, incorporando y adaptando algunas actividades propias de la ingeniería del software como: monitorización y control del proceso, revisiones, evaluación y medición. Además se incide en fomentar la cultura de calidad entre el alumnado de una manera práctica.

En lo que sigue el artículo se estructura de la manera siguiente. En la sección 2 se describen las principales motivaciones de este trabajo que han dado lugar a una mejora del proceso, cuya especificación parcial se muestra en la sección 3. En la sección 4, se describe una herramienta para automatizar el proceso. En la sección 5, se presenta un caso de estudio donde se aplican todas las tareas del proceso. En la sección 6, se aplica la tarea de evaluación de trabajos que sirve como base de un experimento donde se comparan las asignaturas homólogas de trabajos fin de carrera en Ingeniería Informática (II) e Ingeniería Técnica en Informática de Gestión (ITIG). Por último, en la sección 7, se numeran unas conclusiones y líneas futuras de actuación.

2. Motivación

A continuación se describen las principales motivaciones de este trabajo.

El gran número de profesores, tanto tutores como miembros de tribunal exige un alto grado de gestión no exigible en otras asignaturas. Se debe establecer una base de información que sirva como elemento de comunicación entre ellos.

Hasta donde alcanza nuestro conocimiento, en los procesos de gestión actuales no existen medidas experimentales asociadas a los trabajos de los alumnos que ayuden a cuantificarlo y evaluarlo. Como cualquier otra asignatura, necesita incorporarse al proceso de adaptación Europea con la correspondiente asignación de créditos ECTS, donde es necesaria este tipo de información.

Se busca complementar trabajos relacionados con las tareas del proceso de gestión persiguiendo un enfoque automatizable

en entornos universitarios. En concreto, en [2] se abordan guías para alumnos, en [10] se presentan aspectos metodológicos y tareas de evaluación y en [3] se aborda la relación entre tareas y roles.

Se pretende mejorar la calidad de los proyectos incorporando al proceso de gestión actividades de medición. Con esto, se puede evaluar el proceso contra criterios específicos para determinar sus fuerzas y registrar las lecciones aprendidas. Así, cada mejora puede ser analizada como cambios en forma de indicador, o cambios en unidades de medida [6]. La incorporación al proceso de métricas de producto permite avanzar en la búsqueda de un criterio específico de evaluación, que pueda servir de autoevaluación interna para el alumnado y de base de conocimiento para tutores y evaluadores del tribunal. Esta nueva tarea de medición de producto mejora los métodos pedagógicos a través de enseñanzas prácticas, incentivando al alumnado a la incorporación de procesos de medición en el desarrollo del software y en la cultura de calidad.

3. Especificación del proceso

Esta sección presenta los elementos de un proceso de gestión en asignaturas de trabajos fin de carrera. Para la descripción de los elementos se utiliza la notación Software Process Engineering Metamodel Specification (SPEM) en su versión 2.0 [9].

Este apartado comienza dando una visión general del proceso. Posteriormente se describe de manera detallada la tarea de evaluación de trabajos. En [8] se puede obtener una descripción más detallada de los distintos elementos del proceso.

3.1. Visión general

En la Figura 1 se muestra la estructura general del proceso. Se observan las diferentes relaciones entre los elementos del proceso: tareas, roles y productos. Las relaciones representadas son: la responsabilidad de un rol con una tarea y los productos obtenidos como resultado de la ejecución de una tarea.

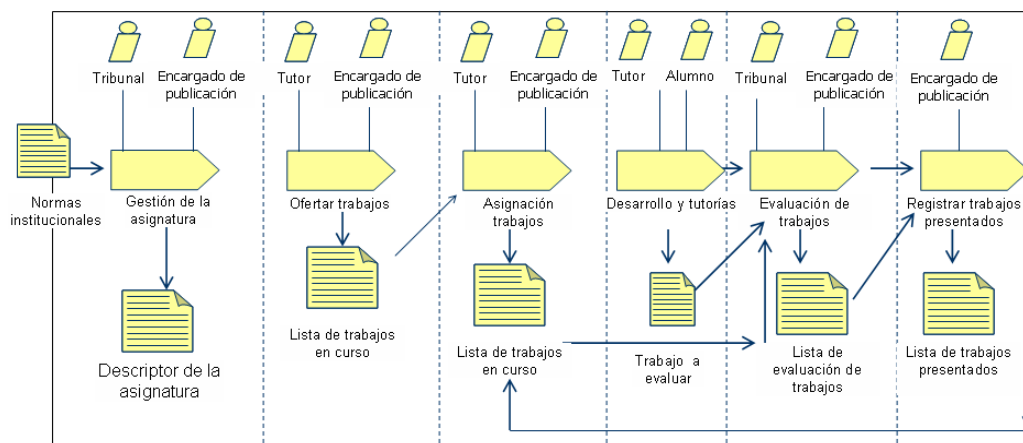


Figura 1. Especificación del proceso de gestión de trabajos fin de carrera

Aunque existen distintas variantes minoritarias en la gestión de proyectos (erasmus, tutorizados en empresas, propuestos por los alumnos) el proceso de gestión básico seguido para la gestión de trabajos fin de carrera es el siguiente:

1. Se establecen unas normas básicas de gestión de la asignatura: miembros del tribunal, calendario, normas y bibliografía.
2. Un tutor oferta sus trabajos (título, descripción, número de alumnos, fecha de publicación) que se envía al encargado de publicación. El encargado de la publicación realiza ajustes para garantizar que existan trabajos para todos los alumnos y publica las ofertas de los trabajos en el curso.
3. Los alumnos seleccionan un trabajo a través de entrevistas con el tutor. Éste selecciona a los alumnos candidatos. La asignación de alumnos al proyecto es comunicada al encargado de publicación.
4. Cuando el trabajo está asignado el seguimiento del mismo es misión del tutor. Se desarrolla el trabajo obteniendo los distintos productos software del proyecto, manual de usuario, manual instalación, código fuente, etc.
5. Se presenta y los miembros del tribunal lo evalúan.
6. Por último se registran los datos del trabajo incluyendo un histórico sobre su desarrollo.

3.2. Tareas: evaluación de trabajos

Después de que los alumnos entreguen sus trabajos existe un periodo en el que el tribunal debe evaluarlos. A juicio de los autores, el establecimiento de un criterio de evaluación cuantitativo determinante de la nota final con un modelo de evaluación muy detallado podría requerir mucho tiempo y añadir una complejidad excesiva. Un enfoque mixto puede consistir en evaluar cualitativamente algunas características y compagarlo con la evaluación cuantitativa de algún subproducto concreto. Para enfrentarse al problema temporal, la adquisición de datos debe de hacerse a través de herramientas de medición. Generalmente este tipo de herramientas disponen de valores recomendables de sus métricas. Por cada métrica, se busca mantener dos indicadores: uno que permita comparar los trabajos realizados en la asignatura y otro que compare con proyectos externos a través de valores recomendados. El indicador de los trabajos internos a partir de métricas, se consigue aplicando las medidas estadísticas percentil 25, mediana, percentil 75, media aritmética y desviación típica sobre todos los trabajos. Se obtiene un indicador de tipo intervalo con los valores recomendados relativos dentro de la propia organización.

No todas las métricas consideradas tienen que disponer de intervalos recomendables.

Además el conjunto de métricas a procesar debe ser lo más abierto para poder evaluar nuevas características de trabajos. La restricción está marcada por el tiempo necesario para recoger los valores y la objetividad de los mismos. Una vez seleccionado el conjunto de métricas y calculados sus intervalos relativos correspondientes a todos los trabajos, se pueden establecer las posibles necesidades de mejora. El inicio del proceso de mejora de la asignatura puede ser definido a través de indicadores de cobertura de métricas respecto a los intervalos recomendados. Se cuenta el número de los valores de las métricas que están dentro de los intervalos recomendados y se divide entre número total de valores. El indicador de cobertura es equivalente al concepto de *compliance* definido en los modelos de calidad de producto ISO [7]. Pueden existir diferentes tipos de cobertura, dependiendo del valor estadístico utilizado: cobertura respecto a la media aritmética, cobertura respecto a la mediana, cobertura respecto intervalo marcado por el percentil 25 y 75.

La evaluación de un trabajo particular se puede realizar desde dos puntos de vista, utilizando los intervalos recomendados o los relativos al resto de trabajos, pudiendo calcular sus coberturas respecto a la mediana para cada uno de ellos. Esto permite obtener un criterio objetivo de evaluación y de comparación de trabajos.

4. Herramienta de gestión de información

Uno de los fines perseguidos con la especificación de un proceso es definir herramientas que ayuden a automatizarlo. Ésta se debe adaptar a las realidades que permitan las tecnologías, herramientas y personas.

Una de las realidades tecnológicas que utilizan los profesores para la gestión de los datos de sus asignaturas son las hojas de cálculo. Algunas de las características que las hacen destacar sobre otros sistemas de almacenamiento de datos son: su fácil manejo en operaciones básicas de mantenimiento de datos y su facilidad para transportar y exportar los datos. Otra de las características, no tan conocida de las hojas de cálculo, es su interoperabilidad con otras

aplicaciones como las bases de datos. Para ello se debe estructurar la información a través de su funcionalidad de definición de nombres.

La información a tratar esta organizada en cuatro niveles de gestión, de manera que el nivel superior incluye la gestión del nivel inmediatamente inferior a él. Los niveles se corresponden con los productos generados en el proceso.

El nivel 1 genera el descriptor de la asignatura a partir de la siguiente información:

- Calendario(Descripcion,Convocatoria, Fecha)
- Documento(Descripcion, Url)
- Norma(Descripcion)
- Tribunal(Cargo, NombreApellidos, Nick)

El nivel 2 obtiene los productos del nivel 1 y genera la lista de trabajos en curso a partir de la siguiente información:

- Alumno(Numero, ApellidosNombre, Dni, Repetidor, Asignado)
- Proyecto(Titulo, Descripcion, Tutor1, Tutor2, Tutor3, Alumno1, Alumno2, Alumno3, Curso_Asignacion)

El nivel 3 obtiene los productos de los niveles 1, 2 y genera la lista de trabajos presentados a partir de la siguiente información:

- Historico(Titulo, Descripcion, Tutor1, Tutor2, Tutor3, Alumno1, Alumno2, Alumno3, Nota, TotalDias, FechaAsignacion, FechaPresentacion,)

El nivel 4 obtiene los productos de los niveles 1, 2, 3 y genera la lista de evaluación de trabajos presentados a partir de la siguiente información:

- DescripcionExperimento(Descripcion)
- MetricaDescripcion(Descripcion, Id, Tipo, MinValor, MaxValor, Visible)
- MetricaValores(M0,M1,M2,M3,J0,J1,J2,J3, J4, J5, J6, J7, J8, J9, J10, J11, J12, J13, J14)

De toda la información presentada, cabe destacar la relacionada con la descripción de las métricas y de los valores de las métricas. Cada uno de los de los identificadores de las filas de la tabla *MetricaDescripcion* se corresponde con las columnas de la tabla *MetricaValores*. De esta forma se persigue disponer de un mecanismo que posibilite la utilización de un conjunto

abierto de métricas. Además la tabla *MetricaDescripcion* contiene una columna de *tipo* para posibilitar diferentes escalas de medida, ya sean nominales o numéricas. Con esto se consigue el objetivo que se mencionó en la tarea de evaluación de trabajos: la elección del conjunto de métricas utilizadas en la evaluación debería estar abierta a decisiones de los encargados de la asignatura y su contextualización particular.

El tratamiento de los datos estructurados ha sido automatizado a través de una herramienta desarrollada desde la propia asignatura. La herramienta es un gestor de información desarrollado en java que accede a los datos a través de ODBC. Genera los productos especificados en el proceso como informes en formato html, cuyo contenido permite a los administradores tomar decisiones justificadas.

Además del procesamiento automático de los datos se incorporan otras funcionalidades en las páginas generadas como:

- fecha de actualización de la páginas
- licencias del contenido
- enlaces a validadores léxicos de páginas web
- sistema de control de acceso a la página a través de servicios de *Google Analytics*.
- Difusión del contenido y monitorización de la difusión a través de sistemas como *Addthis*.

5. Caso de estudio: gestión completa

El proceso aquí expuesto ha sido aplicado en la asignatura de Sistemas Informáticos de 5º de Ingeniería Informática de la Universidad de Burgos y los resultados de los diferentes productos son accedidos por el personal implicado a través de la plataforma de docencia UBUCampus-e. Es una plataforma de e-learning que proporciona recursos para cada asignatura: foros, personal asignado, estadísticas de acceso por alumno, comunidades virtuales para crear redes sociales. El acceso a los resultados del proceso también puede obtenerse en la siguiente dirección

<http://pisuerga.inf.ubu.es/lsi/Asignaturas/SI/index.html>.

Como conjunto de métricas de producto se han elegido las asociadas al código fuente, puesto que en la asignatura la mayoría de los

trabajos llevan asociados un desarrollo software. La herramienta de medición utilizada es SourceMonitor [1]. El criterio de decisión principal para utilizar esta herramienta, ha sido su capacidad para calcular métricas sobre códigos escritos en distintos lenguajes de programación (C++, C, C#, VB.NET, Java, Delphi, Visual Basic 6). Dada la propia naturaleza heterogénea de los proyectos, esta característica se considera imprescindible.

A continuación se describen algunos datos de los distintos productos obtenidos en la aplicación del proceso en este contexto concreto.

La lista de trabajos en curso contiene la información de asignación. Además proporciona datos cuantificables utilizados para distribuir la carga de trabajo entre tutores y datos para gestionar la actividad de asignación. En concreto, a fecha actual están activos 19 trabajos, de los cuales 2 están sin asignar y hay 15 tutores involucrados.

La lista de trabajos presentados almacena un histórico con datos de proyectos presentados junto con su información cuantitativa organizada por cursos: número de trabajos ofertados, número de trabajos presentados, número de alumnos asignados, número de tutores asignados, media aritmética de calificaciones, media aritmética de tiempo de desarrollo medido en meses (ver <http://pisuerga.inf.ubu.es/lsi/Asignaturas/SI/HistoricoSist.html>).

Además se presentan los datos estadísticos (media, mínimo, máximo, desviación estándar) relacionados con la calificación (8.47, 5.3, 9.8, 0.95) y los datos relacionados con el tiempo en días desde que se asigna el proyecto hasta que se presenta (310.83, 120, 1.096, 212.7). Este último indicador de esfuerzo no mide la actividad real, pero puede servir como referencia para estimar créditos ECTS y evaluar posibles acciones de mejora a llevar a cabo.

En la lista de trabajos evaluados se ha seleccionado el conjunto de métricas de código mostrado en la Tabla 1. En la descripción de cada métrica se indica el tipo de posibles valores asignables (columna Tipo), y en caso de algunas métricas numéricas los intervalos de valores recomendados (columnas MinValor y MaxValor).

Descripción	Identificador	Tipo	MinValor	MaxValor
Nombre Proyecto	M0	string		
Lenguaje de Programación	M1	string		
Líneas de Código	J0	number		
Nº de Sentencias	J1	number		
% de Sentencia Condicionales	J2	number		
Nº de Llamadas a Métodos	J3	number		
% Líneas de Comentarios	J4	number	8	20
Nº de Clases e Interfaces	J5	number		
Nº de Métodos por Clase	J6	number	4	16
Avg Sentencias por Método	J7	number	6	12
Max Complejidad	J10	number	2	8
Max Profundidad de Bloques	J12	number	3	7
Avg Profundidad de Bloques	J13	number	1	2,2
Avg de Complejidad	J14	number	2	4

Tabla 1. Descripción de métricas

Con los valores de este conjunto de métricas, se define una comparativa entre los valores estadísticos (intervalo $Q1=$ percentil 25 y $Q3 =$ percentil 75, Avg = media aritmética, Med = mediana) obtenidos con los trabajos de la asignatura y los recomendados. Esta comparación es recogida con los indicadores de cobertura que indican el porcentaje de valores dentro del intervalo (ver Figura 2). Si se tiene en cuenta que dos de las medidas propuestas, J10 y J12 evalúan valores máximos y el total de medidas considerada es 7, parece correcto mantener una cobertura respecto a la media aritmética del 42,86%. Estos indicadores de cobertura pueden servir para establecer índices globales de calidad de la asignatura.

Porcentaje de Cobertura respecto Q1Q3	Porcentaje de Cobertura respecto Avg	Porcentaje de Cobertura respecto Med
14,29% 	42,86% 	42,86% 

Figura 2. Cobertura respecto a intervalos recomendados en II

La evaluación de cada trabajo se basa en la recogida de medidas de cada uno de ellos y la comparación con dos intervalos: los recomendados en [1] (columnas Tabla 1 MinValor, MaxValor) y el correspondiente al

formado por el percentil 25 y 75 del resto de trabajos de la asignatura. Con los valores de estas comparaciones se vuelven a definir nuevos indicadores de cobertura que sirve como criterio de comparación. En la Figura 3 se muestra una evaluación de un conjunto de trabajos (filas de la tabla) utilizando estos indicadores definidos. Además se muestran los valores ordenados respecto a la cobertura de intervalos recomendados (columna Cobertura Tool).

Los datos de evaluación de proyecto de la lista de trabajos evaluados no permiten una trazabilidad directa sobre los datos de la lista de trabajos presentados, preservando en cierta manera la privacidad de las evaluaciones.

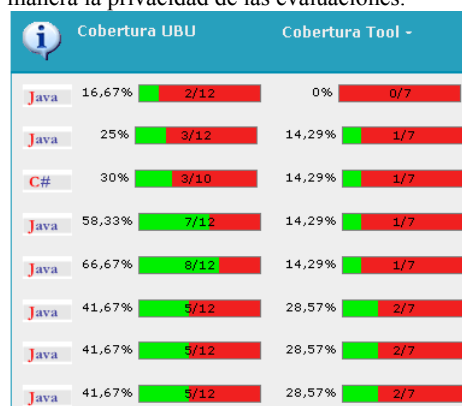


Figura 3. Evaluación de proyectos en II

6. Caso de estudio: comparativa de evaluación en asignaturas diferentes

Una vez establecido un contexto de trabajo, conjunto de métricas seleccionadas, en la tarea de evaluación de trabajos, se posibilita comparaciones globales entre asignaturas. Bajo el contexto elegido nos planteamos la siguiente hipótesis: respecto a métricas de código, ¿existen diferencias entre los trabajos presentados por los alumnos en la ingeniería (II) e ingeniería técnica (ITIG)?.

La tarea de evaluación de trabajos expuesta en el proceso (ver sección 3.2), no había sido llevada a cabo en la titulación de ingeniería técnica. Para simularla, se solicitó a seis profesores del Área de Lenguajes y Sistemas Informáticos copias de los trabajos presentados en las últimas convocatorias. En la Tabla 2 y Tabla 3 se muestran los resúmenes de los

tamaños de los experimentos realizados en ambas titulaciones. Se toma como medida básica de tamaño el número de líneas de código analizadas (NLOC) y su desglose con los distintos lenguajes de programación considerados.

	Total	Java	CSharp
NLOC	276.072	251.893	24.179
%		91,24%	8,76%

Tabla 2. Tamaño experimento ITIG

	Total	Java	CSharp	C
NLOC	876.806	779.164	82.806	14.836
%		88,86%	9,44%	1,69%

Tabla 3. Tamaño experimento II

Después de realizar el análisis estadístico (Q1=percentil 25 y Q3 = percentil 75, Avg = media aritmética, Med = mediana), para cada una de las métricas, se calcula la cobertura respecto a los valores recomendados por la herramienta de medida. La Figura 4 muestra los distintos valores de cobertura obtenidos para ITIG.



Figura 4. Cobertura respecto a intervalos recomendados ITIG

Comparando los resultados con los datos de la titulación de II (ver Figura 2) se aprecia que se mantienen los mismos valores de cobertura respecto al intervalo Q1 Q3 (14,29%) y cobertura respecto Avg (42,86%). Pero hay una mejora sustancial en la cobertura respecto a la mediana, pasa de 42,86% en II a 71,42% en ITIG.

En la evaluación de trabajos es destacable mencionar que la cobertura mínima respecto a valores de la herramienta es 0% en II frente al 28,57% ITIG. Este matiz puede ser importante para justificar la mejora respecto a la mediana en ITIG. El proceso de evaluación en II ha sido completo desde el curso 2003_04, mientras que en ITIG se han solicitado los trabajos a los tutores, constatando luego que habían presentado proyectos con altas calificaciones.

En la Tabla 4 se presenta una comparativa entre ambas titulaciones de dos características

de proyectos, tamaño y complejidad, cuyas métricas asociadas no disponen de valores recomendables en la herramienta de medida. A la característica de tamaño se han asociado las métricas: líneas de código (J0), número de llamadas a métodos (J3) y número de clases e interfaces (J5). A la característica de complejidad se asocia la métrica: porcentaje de sentencias condicionales (J2). A partir de los datos se deduce que el tamaño de los trabajos en ambas titulaciones es bastante similar. Tomando los datos de los límites inferiores de la métrica asociada a complejidad (J2) se puede concluir una mayor complejidad en los trabajos de II respecto a ITIG.

		[Q1-Q3] II	[Q1-Q3] ITIG
Tamaño	J0	[8.320 - 28.185]	[8.670 - 35.896]
	J3	[3.775 - 10.056]	[2.796 - 9.770]
	J5	[40 - 186]	[31 - 191]
Complejidad	J2	[9,8 - 15,9]	[6 - 15,6]

Tabla 4. Comparación entre II e ITIG con métricas sin intervalos recomendados

De esta comparativa se concluye que respecto a las métricas de código existe poca diferencia entre los trabajos realizados por los alumnos en II e ITIG.

7. Conclusiones y líneas futuras

Se ha proporcionado una estructura para un repositorio de información de gestión de trabajos fin de carrera, donde se han aplicado buenas prácticas de ingeniería del software. En concreto se han definido un conjunto de métricas de producto y proceso que pueden ayudar tanto a la evaluación global de la asignatura como a la evaluación de individual de trabajos. Éstas han sido adaptadas a un contexto universitario donde existen unas restricciones específicas: tiempo de dedicación por parte del profesorado y alumnado, carga de trabajo del profesorado, naturaleza heterogénea de los proyectos y fase de evaluación pública.

Aunque somos conscientes de las limitaciones del criterio de evaluación expuesto, a partir de medidas estáticas sobre el código fuente, se proporciona una forma objetiva de evaluar subproductos resultados de la realización del trabajo. El caso de estudio presentado en la sección 5, proporciona una

referencia cuantificable de otros trabajos que puede servir como base tanto para seguimiento como para la comparativa. Además se posibilita la comparación desde un punto de vista global de asignaturas distintas.

Se han mejorado los hábitos de autoevaluación de desarrollos software a través de métricas de producto. De esta manera, se pueden satisfacer dos unidades de conocimiento propuestas en el núcleo de ACM SE 2004 [5], y que son difícilmente conseguidas en el contexto de otras asignaturas: cultura sobre la calidad del software (2 horas) y actividades relacionadas con el control de proyectos (2 horas). Esta característica se ha puesto de manifiesto en la incorporación de evaluaciones mediante métricas por parte de los alumnos en la documentación técnica de sus trabajos.

El modelo de gestión aquí propuesto puede ayudar a avanzar en el proceso de convergencia europea en asignaturas de trabajos fin de carrera. En él se aboga por metodologías docentes donde el alumno tenga un aprendizaje más práctico y más personalizado, sin exigir un modelo rígido presencial, mientras el profesor realiza un seguimiento del mismo.

De cara al futuro, el proceso definido está abierto a la posible incorporación de nuevas métricas que vengan de otras fuentes de información. Con la restricción de que la adquisición de valores debe realizarse de manera automática y en un tiempo reducido (< 10 minutos por trabajo). En este sentido, los sistemas de control de versiones pueden dotar gran información sobre la actividad real del proceso desarrollo llevado a cabo por los alumnos.

El experimento planteado de comparación de trabajos entre II e ITIG, ha revelado que no existen grandes diferencias en los mismos. Para poder afianzar estos resultados sería interesante realizar una simulación con datos de otras Universidades. Además se abriría la posibilidad de dar respuesta a otras nuevas hipótesis: ¿es equiparable el tamaño y complejidad de los trabajos en universidades diferentes?

Referencias

- [1] CampWood Software, SourceMonitor. 2007. Disponible en <http://www.campwoodsw.com/sourcemonitor.html>
- [2] Dawson, C.W. and G. Martín. El Proyecto Fin de Carrera en Ingeniería Informática: una guía para el estudiante. Prentice Hall, 2002.
- [3] Fernández, J., et al. PFC: Los dos lados del espejo. Proyectista - Director ¿Una visión compartida? VII Jornadas de Enseñanza Universitaria de la Informática (Jenui), 2007.
- [4] IEEE, C.S., Guide to the Software Engineering Body of Knowledge: 2004 Edition - SWEBOOK. 2005.
- [5] IEEE, C.S.A..C.M., Software Engineering 2004 Curriculum Guidelines for Undergraduate. Degree Programs in Software Engineering. 2004, National Science Foundation under Grant No. 0003263.
- [6] ISO/IEC, ISO/IEC 15939:2002, Software Engineering-Software Measurement Process. 2002.
- [7] ISO/IEC, Software engineering -- Product quality Part 1: Quality model. 2001.
- [8] López C., Marticorena R., Rodríguez J.J, Gestión de Trabajos Fin de Carrera. X Simposio Internacional de Informática Educativa (SIIIE), 2008.
- [9] OMG, Software Process Engineering Meta-Model 2.0 2006.
- [10] Polo Marquéz, A., J. Matínez Gil, and L.J. Arevalo Rosado. Hacia una metodología para el desarrollo de trabajos y Proyectos Fin de Carrera en Ingeniería Informática. VII Jornadas de Enseñanza Universitaria de la Informática (Jenui), 2007.
- [11] Pressman, R.S., Ingeniería del software : un enfoque práctico. 6ª ed. 2005, McGraw-Hill.
- [12] Sommerville, I., Ingeniería del software. 7ª ed. 2005, Pearson Educación.