

Resultados preliminares más significativos tras cuatro años de aplicación de la metodología SCRUM en las prácticas de laboratorio

Luis Castillo Vidal
Departamento de Ciencias de la Computación e I.A.
Universidad de Granada
18071 Granada
L.Castillo@decsai.ugr.es

Resumen

El uso de metodologías de desarrollo ágil como SCRUM está cobrando adeptos a nivel profesional y, cada vez más, se introduce como materia académica en los estudios universitarios de informática, sobre todo en las prácticas de laboratorio organizadas como trabajo en equipo. Este trabajo presenta un estudio en el que se han recogido datos durante los últimos cuatro cursos académicos en las prácticas de una de estas asignaturas y demuestra una serie de hechos interesantes para el profesorado. Por un lado, que los alumnos planifican bien el tiempo mediante esta metodología, independientemente de que sean equipos de trabajo grandes o pequeños. Y por otro lado, que existe una fuerte correlación entre una buena gestión del tiempo y la alta calificación obtenida por los equipos de alumnos. Este trabajo también proporciona las evidencias estadísticas necesarias para soportar estos hechos.

Abstract

The use of agile development methodologies such as SCRUM is becoming widely adopted at professional levels, but also increasingly adopted as an academic subject within computer science degrees, mainly in team-based laboratory practices. This work presents a study based on data gathered over the last four academic years and it proves several facts that might be relevant for teachers. On one hand, teams of students carry out a good temporal planning of their projects, regardless they are small or large teams. And, on the other hand, there exist a strong correlation between a good temporal planning and a high grade obtained by team of students. This work also provides the necessary statistical evidences to support those facts.

Palabras clave

Metodologías de desarrollo ágil, Prácticas de laboratorio, Planificación del tiempo

1. Introducción

Aprender a planificar el tiempo durante un proyecto de desarrollo de software es una de las competencias transversales más importantes durante los estudios del Grado en Informática [1], pues puede influir de forma importante en el futuro rendimiento profesional de los alumnos. Está demostrado que el uso de metodologías ágiles como SCRUM [4] en entornos industriales lleva a procesos de desarrollo de software más rápidos, baratos y mejores [5]. En un entorno académico debería permitir a los alumnos planificar el tiempo que se le dedica a un proyecto y su seguimiento a lo largo del mismo y evitar que este proceso sea percibido como una caja negra.

Los trabajos analizados en la sección 2 coinciden en corroborar estas bondades, pero sus argumentos no pasan de consideraciones subjetivas vagas o testimonios generales de los alumnos. Este trabajo pretende cuantificar de forma objetiva estas bondades y responder a preguntas como ¿Se está gestionando bien el tiempo de desarrollo? ¿Influye el tamaño de los equipos de trabajo? ¿Influye una buena gestión temporal del proyecto en una buena calificación del mismo? ¿Con qué significación estadística se pueden argumentar estos resultados?

Para ello, este artículo presenta un estudio, de cuatro años de duración [2], en el que se han recogido datos sobre la planificación y seguimiento temporal diario de más de 30 grupos de alumnos durante el desarrollo de las prácticas de una asignatura obligatoria de cuarto curso del Grado en Informática de la Universidad de Granada con el objetivo de analizar si, efectivamente, la introducción de esta metodología ha supuesto una

mejora cuantificable o no. Cuantificar el éxito de la planificación temporal de un proyecto software desarrollado por un equipo de programadores puede ser una tarea subjetiva, por ello, se presentan y justifican distintas métricas de éxito y se comparan con diferentes variables independientes para explorar si existe alguna correlación entre estos parámetros que sea estadísticamente significativa con un nivel de significación del 5 % ($\alpha = 0,05$) en base a los datos observados.

El objetivo último de este estudio, es instrumentalizar estos resultados para dotar al profesor de herramientas que le permitan ayudar a los alumnos a definir sus grupos de trabajo y a planificar el tiempo que le dedican a sus prácticas de la manera más adecuada para, en la medida de lo posible, buscar un mayor éxito en la gestión del tiempo disponible en el proyecto y en la calidad del software desarrollado.

El artículo está estructurado de la siguiente forma. La sección 2 repasa la literatura existente y la sección 3 introduce el contexto en el que se ha llevado a cabo este estudio. La sección 4 introduce la implantación de SCRUM en el laboratorio de prácticas y las principales variables muestreadas en el estudio. La sección 5 describe el análisis realizado sobre estas variables y las relaciones encontradas entre ellas y termina con la sección 6 en la que se trazan las principales conclusiones, las fortalezas y las debilidades de este estudio y el trabajo futuro.

2. Trabajos relacionados

Existen muchos trabajos en la literatura que presentan la introducción exitosa de SCRUM en diversas asignaturas del Grado en Informática como por ejemplo [6] o en [12] donde se analiza el testimonio positivo de los estudiantes ante la introducción de estas metodologías, incluso reconociendo las dificultades y restricciones inherentes a un entorno académico frente a un entorno industrial [8].

Sin embargo, aunque a nivel académico no se analiza su relación con las calificaciones obtenidas por los alumnos, sí es cierto que, a nivel industrial, se pueden encontrar iniciativas que sí intentan vincular la introducción de SCRUM al éxito en la consecución de objetivos empresariales. En [9] se analizan distintas métricas de calidad asociadas a SCRUM desde distintos puntos de vista, no siempre compatibles entre sí: el punto de vista del equipo de desarrollo, el del equipo de gestión y el del cliente. En [5, 7] se analiza la influencia del uso de métodos ágiles en la calidad del software, fundamentalmente en el número de defectos del software resueltos, llegando a la conclusión de que no produjo menos errores pero sí que hizo el proceso de control de calidad del software más estable y eficiente y que el desarrollo fuese más productivo.

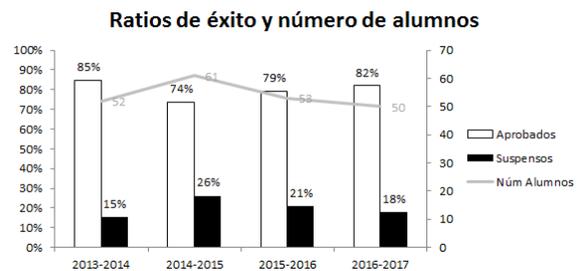


Figura 1: Número de alumnos matriculados y porcentajes de éxito y fracaso durante los últimos cuatro años.

3. Contexto del estudio

La asignatura en la que se ha llevado a cabo el estudio se denomina “Desarrollo Basado en Agentes”, y es una asignatura obligatoria del cuarto curso del Grado en Informática de la Universidad de Granada (perfil de Ingeniería del Software) y está centrada en el diseño e implementación de sistemas multiagente mediante el uso de la plataforma Magentix [11]. Es una asignatura de evaluación continua cuyo esquema de evaluación es el siguiente, en el que se puede ver que hay dos prácticas de programación cuya puntuación es de cinco puntos sobre un total de 10 puntos totales.

- Práctica 1. Ensayo de investigación. 1 punto.
- Práctica 2. Programación. 2 puntos.
- Práctica 3. Programación. 3 puntos.
- Examen escrito. Sólo modelado. 3 puntos.
- Participación activa. 1 punto.

Estas dos prácticas de programación se llevan a cabo por grupos de alumnos, en sendos ciclos de desarrollo, siguiendo la metodología de desarrollo SCRUM según se detalla en [2] y utilizando un laboratorio virtual que registra todas las interacciones del software desarrollado por los alumnos [3]. Los grupos de trabajo pueden ser de 5 o 6 alumnos y excepcionalmente de 4 alumnos (de forma similar a otros estudios [8]). Es una asignatura con un alto grado de éxito (ver Figura 1) y una dedicación muy alta por parte del alumnado (ver Figuras 2 y 3).

3.1. La metodología SCRUM

Esta sección no pretende introducir todas las características de SCRUM [2, 4] sino sólo aquellas necesarias, y de forma muy concisa, para el ámbito de este estudio, concretamente las relacionadas con la gestión del tiempo de trabajo.

SCRUM es una metodología ágil de desarrollo de software que es iterativa e incremental en la cual un equipo de programadores trabaja como una unidad auto organizada para alcanzar un objetivo común. El objetivo es un producto de software que se ha diseñado a partir de una serie de características que se desean



Figura 2: Número de sesiones de trabajo abiertas en el servidor de la asignatura según la hora del día. Datos del curso 2016-2017.



Figura 3: Número de sesiones de trabajo abiertas en el servidor de la asignatura según el día de la semana. Datos del curso 2016-2017.

obtener, a las que se les suele llamar historias, y que se encuentran catalogadas en lo que se conoce como backlog de producto.

El desarrollo del producto se lleva a cabo de forma incremental en ciclos de trabajo relativamente cortos, de 2 a 4 semanas, por ejemplo, que se llaman sprints. En cada iteración, dependiendo de la duración del sprint y del tamaño del equipo, el equipo de desarrollo selecciona un subconjunto de las historias del backlog de producto que cree que será capaz de desarrollar y diseña (o planifica) lo que será el backlog del sprint, cuantificando el tiempo, normalmente en días, que le dedicará a cada historia seleccionada. El total de tiempo que se le espera dedicar al sprint se conoce como tamaño del sprint.

De forma diaria el equipo de desarrollo se reúne, detalla cuánto tiempo de desarrollo efectivo ha realizado hasta la fecha y lleva la cuenta del tiempo que queda por desarrollar para cada historia, dibujando lo que se conoce como el diagrama burndown de evolución del sprint, como el mostrado en la Figura 4.

En ella se puede observar cómo debería ser la evolución ideal del tiempo que queda por desarrollar acumulado en las historias del sprint (línea recta descendiente de color claro) y la evolución real del tiempo que queda por desarrollar (línea oscura quebrada). Cuando el tiempo que queda por desarrollar (línea oscura) es superior al ideal (línea clara), el equipo se entiende que

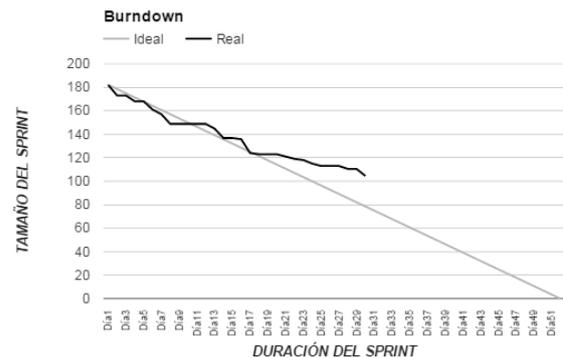


Figura 4: Diagrama burndown de un equipo de desarrollo real, más o menos a mitad del sprint. Se puede ver que el tamaño del sprint es de 180 unidades y que la duración del sprint es de 51 días. También indica un leve retraso acumulado en el desarrollo de las historias del backlog del sprint.

va retrasado. Cuando sucede al revés entonces el equipo va adelantado en el desarrollo del sprint.

Lejos de ser una norma estricta a la que adherirse, el diagrama de burndown pretende concienciar al equipo, de forma diaria, del tiempo de desarrollo estimado que queda en el sprint para poder llegar con éxito al final del mismo. También se puede entender como realimentación del equipo de desarrollo con respecto a la planificación del tamaño del sprint que se hizo al comienzo del mismo, con el objetivo de mejorar esta capacidad de planificación en futuros sprints. En definitiva, una de las ventajas de la metodología SCRUM es dotar de una metodología simple pero efectiva que evite ver el proceso de desarrollo de software como un proceso oscuro e impredecible y aprender, iteración a iteración, a planificar el tiempo de forma eficaz.

4. Implantación de SCRUM

La metodología SCRUM se ha implementado en las prácticas de la asignatura objeto de este estudio (ver [2] para una descripción completa¹) durante los últimos cuatro años académicos. El ciclo de vida que siguen los alumnos de forma típica es el descrito en Figura 5 y se comenta en las subsecciones siguientes.

4.1. Creación del equipo

Sucede al comienzo de la asignatura y, para darle más homogeneidad al desarrollo, los grupos no cambian durante toda la asignatura. La principal variable en este momento es el tamaño del equipo creado (TE),

¹Los alumnos utilizan una hoja de cálculo compartida con el profesor en Google Drive para monitorizar la aplicación de la metodología SCRUM.

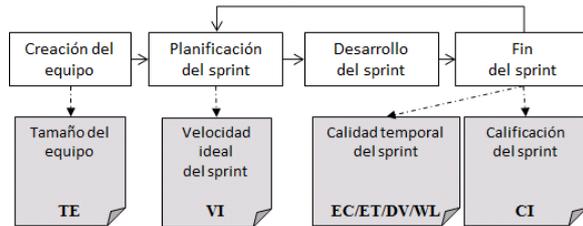


Figura 5: Etapas en el desarrollo de las prácticas y principales variables recogidas en cada etapa durante el estudio. El significado de estas variables se introduce a lo largo de estas subsecciones.

que varía entre 4 y 6 alumnos (33 grupos estudiados, 4 grupos de 4 alumnos, 8 grupos de 5 y 21 equipos de 6).

4.2. Planificación del sprint

Como se ha comentado anteriormente, hay dos iteraciones o sprints, asociados a la Práctica 2 y a la Práctica 3 respectivamente, cuya planificación se ve reflejada en las siguientes variables:

Duración del sprint. Número de días dedicados a cada práctica. Normalmente, en la vida profesional sólo se contabilizan los días laborables como duración del sprint y cada día se contabiliza, normalmente, como 8 horas de trabajo. No obstante, en este caso, dado que es una asignatura continua a lo largo de un semestre y la alta dedicación de los alumnos día a día (ver Figuras 3 y 2) se ha optado por considerar útiles todos los días del sprint entendiéndose que el número de horas diarias dedicadas puede ser muy variable según el grupo.

Tamaño del sprint. Dado que es un desarrollo dentro del trabajo de una asignatura, que comparte semestre con otras asignaturas del Grado, el tamaño del sprint (y las historias de desarrollo, por tanto) se mide en horas, no en días. Todos los grupos tienen un número mínimo de horas que planificar, el cual coincide con el número de horas prácticas de la asignatura durante el sprint, pero cada grupo es libre de planificar más horas de trabajo adicional en casa según la propia disponibilidad de cada miembro del equipo.

Velocidad ideal del sprint (VI). Es la ratio entre el tamaño del sprint y su duración (pendiente de la recta ideal) y es constante a lo largo del sprint.

4.3. Desarrollo del sprint

Durante el sprint, cada equipo se reúne a trabajar en el desarrollo de la práctica correspondiente y es libre de elegir reunirse únicamente durante los días de clase práctica o en sesiones compartidas de trabajo en casa, según la disponibilidad del grupo. En cualquier caso

todos los equipos anotan en una hoja de cálculo personalizada en Google Drive [2] las horas de desarrollo que se han hecho en cada sesión y se repasan con el profesor, de forma semanal, durante las clases prácticas de la asignatura. La principal variable que se puede observar es la velocidad real de desarrollo (pendiente de la línea de desarrollo real del burndown –ver Figura 4) la cual varía a lo largo del sprint, según la disponibilidad de trabajo de cada grupo y la dificultad de las historias que se estén desarrollando.

4.4. Fin del sprint

Una vez finalizado el sprint se obtienen indicadores sobre la bondad de su evolución bajo dos puntos de vista distintos: la calidad temporal del sprint y la calificación del sprint.

La calidad temporal del sprint es un aspecto controvertido para el que no existe un posicionamiento claro. En [9] se reconoce que las distintas métricas de calidad de la evolución del sprint son cuantitativas (gestión del tiempo) y cualitativas (calidad del producto software) y que, en cualquier caso también dependen del punto de vista del actor durante este proceso: el cliente, el equipo de desarrollo o los gestores de la empresa de desarrollo. Una de las métricas que se sugiere es que la varianza de la velocidad de desarrollo real sea mínima, para mantener un ritmo constante de desarrollo y evitar picos de trabajo que puedan poner en peligro el éxito del sprint. El proveedor de unos de los principales servicios de software orientado a la implantación de SCRUM también reconoce esta dificultad y se argumenta que, incluso la velocidad ideal del sprint puede depender del equipo de trabajo y de su capacidad de planificar y resolver historias². No obstante, se argumenta que es de esperar que, conforme el equipo de desarrollo vaya mejorando su experiencia sprint tras sprint, la velocidad real de desarrollo se incremente. Por ello parece que la primera métrica de calidad en la gestión del tiempo es mantener una velocidad de desarrollo homogénea, con poca variación, es decir que la desviación estándar de la velocidad real de desarrollo (DV) tienda a 0.

En [2] se propone la ratio entre las áreas de la función de desarrollo ideal respecto a la función de desarrollo real como medida de la eficacia temporal (ET). Esta métrica tiende a 1 conforme ambas funciones se acercan y tiende a 0 conforme el desarrollo se va atrasando y alejando del ideal y, además, normaliza la duración y tamaño del sprint.

Existen otras perspectivas más clásicas en la comparación de dos series temporales que se comentan a continuación.

²Atlassian: Cinco métricas ágiles que no odiarás. <https://es.atlassian.com/agile/metrics>

- Error cuadrático medio (EC). Muy útil para cuantificar oscilaciones alrededor del ideal.
- Porcentaje de días con retraso acumulado. Esta es una métrica muy utilizada en problemas de scheduling industrial [10] pero en nuestro caso es insensible a la geometría del sprint, pudiendo tener pocos días pero con mucho retraso y no ser perceptible. Para resolver este problema también se propone el número de días atrasado ponderado por su retraso correspondiente con respecto al ideal (WL).

El uso de uno o más de estos indicadores para medir la bondad de la curva de desarrollo real frente a la ideal, es decir, si los alumnos están planificando su tiempo bien con respecto al ideal, parece estar correctamente argumentado y no parece claro decantarse por una o por otra de las métricas presentadas: DV (desviación de la velocidad real de desarrollo), ET (eficacia temporal), EC (error cuadrático medio) o WL (suma de los retrasos acumulados). Todos ellos se pueden utilizar al final del sprint para validar si ha evolucionado bien, pero también se pueden utilizar durante las reuniones semanales/diarias por el profesor para detectar, en la medida de lo posible, errores tempranos en la gestión del tiempo y desviaciones respecto a lo esperado para tomar las medidas correctoras que estime necesarias para reconducir el proyecto. No obstante, en la Sección 5 se analizarán los datos históricos recogidos para arrojar algo de luz sobre la adecuabilidad de cada una de estas variables.

Respecto a la calificación del sprint, corresponde al profesor evaluar la calidad del producto de software tanto desde el punto de vista general del Grado como de la adecuación a las directrices de la asignatura y del enunciado de las prácticas en último extremo, emitiendo una calificación final (CI). La naturaleza de esta evaluación es muy subjetiva, puede ser incompleta y depende de factores humanos relacionados con la composición del grupo, pero es la realimentación más importante que se tiene en la asignatura sobre la calidad del producto de software que se ha desarrollado.

5. Análisis preliminar

Durante los últimos cuatro años académicos se ha recogido información diaria sobre todas las variables mencionadas en la sección anterior, dando por hecho que la introducción de SCRUM en la elaboración de las prácticas de laboratorio es una buena herramienta para la gestión de proyectos de desarrollo de software, pero este análisis pretende investigar el impacto que esta técnica puede estar teniendo entre los alumnos y buscar, en la medida de lo posible, correlaciones entre estas variables, sobre todo las variables controlables

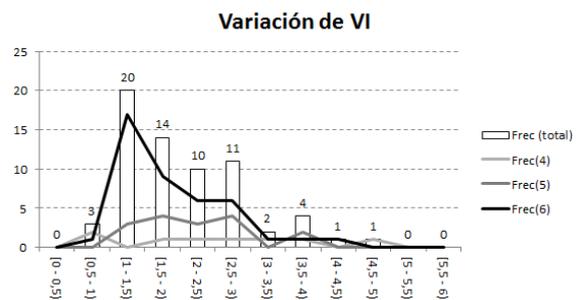


Figura 6: Variación de la variable VI, la velocidad ideal del sprint, en horas/día, para un total de 66 sprints analizados ($M=2,06$ y $SD=0,89$). Se muestran también las frecuencias parciales para los grupos de 4, 5 y 6 alumnos.

por el profesor como el tamaño de los grupos, el tamaño o la velocidad de los sprints o la eficacia temporal del desarrollo diario del sprint.

En total se han recogido datos de 33 grupos durante 2 sprints cada uno (lo habitual en este tipo de asignaturas es realizar 2-3 sprints por semestre [8, 12]). La unidad temporal de planificación es por horas (más adecuado que por días, dado que es una asignatura que comparte espacio con otras asignaturas del grado). En las siguientes subsecciones aparecen los histogramas en la recogida del resto de variables mencionadas en la sección anterior para cada uno de los 66 sprints analizados.

5.1. Análisis de las planificaciones

Dado que la duración de estas prácticas se debe adaptar al calendario académico, la duración de cada sprint está fijada de antemano por el profesor y se corresponde, para el primer sprint (Práctica 2) una media de 37 días y el segundo sprint (Práctica 3) una media de 51 días.

En cuanto a la velocidad ideal de desarrollo (VI) de cada sprint (Figura 6), depende del tamaño del sprint que haya programado cada equipo y de la duración del sprint. En este caso, se puede observar que las velocidades se agrupan entorno a una media de 2,06 h/día (desviación típica de 0,89) con una cierta variabilidad que depende de la disponibilidad temporal de cada equipo en la asignatura, según se ha comentado anteriormente.

5.2. Análisis de la gestión del tiempo

Se puede observar en las Figuras 7 a 10 un buen comportamiento de las variables asociadas a la gestión del tiempo EC, ET y WL, las cuales muestran claramente un comportamiento esperado cuyo valor normal converge al valor óptimo de la serie, es decir, tiende a

1 para ET y tiende a 0 para EC y WL.

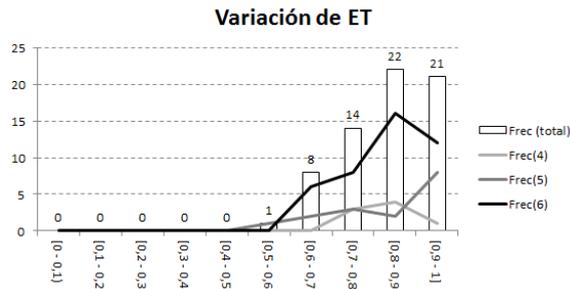


Figura 7: Variación de ET, eficacia temporal (M=0,85 y SD=0,14). Se muestran también las frecuencias parciales para los grupos de 4, 5 y 6 alumnos.

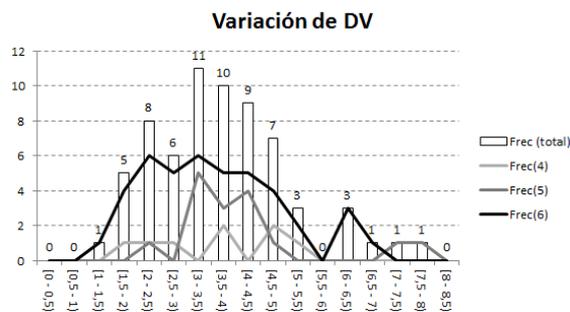


Figura 8: Variación de DV, desviación estándar de la velocidad de desarrollo real (M=3,72 y SD=1,38). Se muestran también las frecuencias parciales para los grupos de 4, 5 y 6 alumnos.

No obstante la variable DV muestra un comportamiento diferente cuyo valor normal, lejos de converger al valor óptimo de la serie, que es 0, se estanca entorno a la media de 3, 7, es decir, que la velocidad de desarrollo real, lejos de estabilizarse, sufre cambios constantes.

5.3. Análisis de las calificaciones obtenidas

La Figura 11 muestra el histograma de calificaciones obtenidas en los 66 sprints recogidos (estas no son las calificaciones de la asignatura, sino únicamente las de las prácticas 2 y 3) y se puede ver que su comportamiento también es normal.

Parece claro, por los resultados anteriores, que los alumnos planifican bien el tiempo que dedican a cada sprint (con las métricas de gestión del tiempo tendiendo a sus valores óptimos) y que las calificaciones son buenas, pero estaría bien conocer si existe una relación entre las variables estudiadas de cara a hacer más accionable por el profesor los resultados de este análisis histórico y permitirle actuar, con la evidencia

Variación de EC

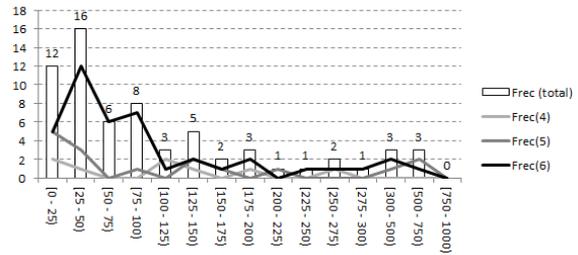


Figura 9: Variación de EC, error cuadrático medio (M=122,61 y SD=147,98). Se muestran también las frecuencias parciales para los grupos de 4, 5 y 6 alumnos.

Variación de WL

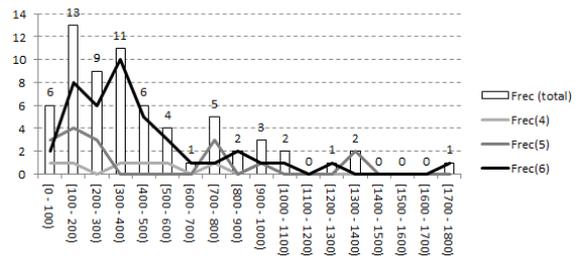


Figura 10: Variación de WL, retraso total acumulado (M=455,79 y SD=372,30). Se muestran también las frecuencias parciales para los grupos de 4, 5 y 6 alumnos.

suficiente, para detectar y corregir posibles desviaciones siempre con el objetivo de que el rendimiento de los alumnos sea superior. Por ello, a partir de los datos recogidos durante estos cuatro años, se plantean las siguientes hipótesis.

Hipótesis 1 . El tamaño del equipo (TE) está relacionado con la eficacia de la gestión temporal del sprint, se mida esta como se mida (EC, DV, ET o WL).

La respuesta a esta hipótesis permitiría saber si el hecho de crear un grupo demasiado pequeño o

Variación de CI

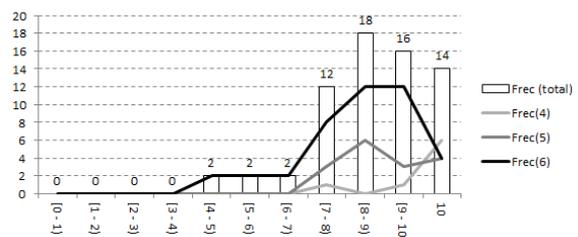


Figura 11: Variación de CI, calificación del sprint (M=8,53 y SD=1,34). Se muestran también las frecuencias parciales para los grupos de 4, 5 y 6 alumnos

Tamaño del equipo (TE)			Calificación del sprint (CI)		
	Correlación de Pearson	Significación		Correlación de Pearson	Significación
ET	-0,007	0,958	ET	+0,352	0,004
DV	-0,086	0,494	DV	-0,044	0,723
EC	-0,050	0,687	EC	-0,377	0,002
WL	-0,081	0,517	WL	-0,369	0,002

Cuadro 1: Correlación de Pearson entre TE y las diferentes medidas de eficiencia temporal de sprint y sus valores de significación estadística.

Cuadro 2: Correlación de Pearson entre CI y las diferentes medidas de eficiencia temporal de sprint y sus valores de significación estadística.

demasiado grande puede afectar a la capacidad de gestión del tiempo planificado en el sprint, permitiendo al profesor reducir o ampliar los grupos en caso necesario.

Hipótesis 2 . La eficacia en la gestión temporal, se mida esta como se mida (EC, DV, ET o WL) está relacionada con la calidad final del producto (CI).

Esta es una hipótesis muy potente pues permitiría saber si los equipos que gestionan bien el tiempo, además, consiguen una buena calidad en sus productos, algo que a la inversa casi es una obviedad, pues se intuye que los equipos que no gestionan bien el tiempo se exponen a picos de actividad muy intensos y erráticos que ponen en peligro la calidad del producto que se está desarrollando.

Para responder a estas hipótesis se ha realizado un análisis de correlación de Pearson de las muestras de las respectivas variables obteniendo los resultados mostrados a continuación.

Hipótesis 1

Los resultados mostrados en el Cuadro 1 desvelan que no se puede demostrar una correlación significativa entre ambas variables, se mida la gestión temporal como se mida, con unos valores de correlación muy bajos y unos valores de significación muy pobres ($p >> 0,05$) por lo que, en el marco de este estudio, y a la luz de los datos históricos recogidos, la elección del tamaño del grupo por parte de los estudiantes no va a suponer un sesgo en la capacidad del equipo para gestionar bien el sprint.

Hipótesis 2

Los resultados mostrados en el Cuadro 2 son muy relevantes en relación con esta hipótesis. Muestran que, a partir de la evidencia de los registros históricos, existe correlación entre la evolución correcta del sprint medida con las variables ET, EC o WL, y la calificación del sprint (CI), esta correlación va en el sentido correcto (directa para ET e inversa para EC y WL) y es muy significativa en todas ellas ($p < 0,01$) lo que las convierte en buenos indicadores, no sólo de la calidad de gestión temporal del sprint, como se ha visto al

comienzo de la sección, sino como estimadores de la calidad del producto software que se está desarrollando.

De hecho, en los datos recogidos también se puede apreciar que ningún grupo que tenga valores de $ET \geq 0,94$ o valores de $EC \leq 5,2$ o valores de $WL \leq 280,0$ consigue una calificación inferior a 7,2, lo cual podría ser un predictor muy potente de cara a la monitorización diaria/semanal de los alumnos.

No obstante, los datos de correlación mostrados en el Cuadro 2 también muestran, de nuevo, un comportamiento anómalo de la variable DV, al igual que pasaba en la sección 5.2. En este caso se puede ver que la correlación entre DV y CI es prácticamente inexistente y muy poco significativa ($p >> 0,05$) lo que la inhabilita como elemento en base al cual el profesor pueda tomar decisiones correctoras durante el desarrollo del sprint. En definitiva, parece que las medidas centradas en cuantificar la diferencia entre la curva de desarrollo real y la de desarrollo ideal (como ET, EC o WL) se comportan bien y tienen una relación importante con la calidad del producto final, no siendo así con la medida DV, centrada en medir variaciones de la velocidad de desarrollo, la cual no se comporta de la forma esperada ni tiene relación demostrable con la calidad del producto final.

6. Conclusiones y limitaciones

Los principales resultados positivos de este estudio son muy interesantes y vinculan la introducción de SCRUM a la consecución de objetivos académicos. En primer lugar, los alumnos están realizando una buena gestión temporal, con métricas de elección alternativa, pero que convergen a sus valores óptimos, a pesar de haber realizado sólo dos sprints consecutivos. En segundo lugar, existen evidencias fuertes de la correlación de una buena gestión temporal con la calidad del producto final, medida como la consecución de objetivos académicos, es decir, cuanto mejor se controla el tiempo de un sprint, mejores calificaciones se consiguen. Y todo ello sin que el tamaño del equipo sea una

variable significativa, pudiendo permitir el profesor la creación de grupos diversos que mejor se adecuen a las necesidades de los alumnos.

No obstante, aún se puede realizar un estudio más en profundidad que ponga de manifiesto posibles variables ocultas en el proceso y arroje evidencias más robustas y eficaces para empoderar la labor del profesor durante el seguimiento de las prácticas de programación de la asignatura. Una de estas variables ocultas es la existencia de grupos heterogéneos de alumnos, con un rendimiento diferente cada uno de ellos, con disponibilidades temporales diferentes debido al seguimiento de otras asignaturas y, posiblemente con ambiciones diferentes de cara a la calificación final de la asignatura. Este aspecto está pendiente de un estudio posterior dado que la implantación de SCRUM descrita en [2] permite recoger algunas de estas variables para medir la heterogeneidad en el desarrollo del sprint (equipos con esfuerzo individual no balanceado) y las diferentes calificaciones que los miembros de un mismo equipo obtienen al final del sprint, pues no todos los alumnos de un mismo equipo tienen por qué obtener la misma calificación en el mismo sprint.

Además de esta limitación, existe otra limitación inherente en el propio estudio y es que los miembros del equipo de desarrollo no son profesionales 100% dedicados a un proyecto, sino alumnos universitarios con múltiples asignaturas que superar y a las que dedicar el tiempo. Además de esto, los alumnos sólo pueden dedicar dos sprints dentro de la asignatura al mismo proyecto, lo cual no es mucho, pero es lo habitual en un entorno de docencia universitaria cuyas asignaturas quedan limitadas a un semestre. Esto hace que el comportamiento temporal de los alumnos sea diferente a un programador profesional. Por ejemplo, la Figura 2 y la Figura 3 muestran un esquema de distribución del trabajo lejano a las 8 horas diarias de lunes a viernes de un programador profesional. Esto podría haber influido en el mal comportamiento y la nula correlación de la variable DV dado que los alumnos no siguen un patrón temporal homogéneo ni constante. Esto haría muy difícil sacar estos resultados del mundo académico y extrapolarlos al mundo industrial, en el que este planteamiento temporal es muy diferente e incluso los objetivos que se persiguen también son diferentes.

No obstante, los resultados de este estudio son potencialmente extrapolables a cualquier asignatura en la que el trabajo práctico sea de programación y se siga la metodología SCRUM.

Agradecimientos

El autor quiere agradecer la valiosa ayuda prestada por José Juan Sánchez Cruz, profesor e investigador

de la Escuela Andaluza de Salud Pública de la Junta de Andalucía, durante la preparación y el análisis de los datos recogidos.

Referencias

- [1] Á. Benito-Capa and A. Cruz-Chust. *Nuevas claves para la docencia universitaria en el Espacio Europeo de Educación Superior*. Narcea Ediciones, 2005.
- [2] Luis Castillo. *The use of SCRUM for laboratory sessions monitoring and evaluation in a university course. Enforcing transverse competencies*. SIIE 2014.
- [3] Luis Castillo. *A virtual laboratory for multiagent systems: Joining efficacy, learning analytics and student satisfaction*. SIIE 2016.
- [4] Henrik Kniberg. *Scrum and XP from the Trenches*. Lulu.com Ed, 2007.
- [5] K. Korhonen. *Evaluating the impact of an agile transformation: a longitudinal case study in a distributed context*. *Software Quality Journal* (2013) 21(4), 599-624.
- [6] P. Letelier Torres and M. Penadés Gramaje. *Una estrategia para la enseñanza de metodologías ágiles*. Jornadas de Enseñanza Universitaria de la Informática. Castelló de la Plana. 2013.
- [7] J. Li and N. B. Moe and T. Dyba. *Transition from a plan-driven process to scrum: a longitudinal case study on software quality*. In *Proceedings of the 2010 ACM-IEEE international symposium on empirical software engineering and measurement* (p. 13). ACM.
- [8] C. Loftus and M. Ratcliffe. *Extreme programming promotes extreme learning?* *Proc. Int. the 10th annual SIGCSE Conf. on Innovation and technology in computer science education*. ACM, New York, NY, USA, 2005, pp. 311-315
- [9] V. Mahnič and I. Vrana. *Using stakeholder-driven process performance measurement for monitoring the performance of a Scrum-based software development process*. *Elec-trotechnical Review*, n 72, 2005.
- [10] M. Pinedo. *Scheduling*. Springer, 2015.
- [11] J. M. Such and A. García-Fornes and A. Espinosa and J. Bellver. *Magentix2: A privacy-enhancing Agent Platform*. *Engineering Applications of Artificial Intelligence* (2013) vol. 26, n.1, p. 96-109.
- [12] C. Wallace and S. Mohan and D. Troy and M. E. Hoffman. *Scrum Across the CS/SE Curricula: A Retrospective* *SIGCSE Technical Symposium on Computer Science Education*, 2012, pp. 5-6