

Influencia de SCRUM en los plazos de entrega y rendimiento en los proyectos de las asignaturas de Desarrollo de Software

Influence of SCRUM in the delivery times and performance in the projects of the subjects of Software Development

Oscar Efraín Capuñay Uceda¹, Juan Manuel Antón Perez¹

¹ Universidad Tecnológica del Perú, Chiclayo, Perú

c16762@utp.edu.pe, c16712@utp.edu.pe

Recibido: 01/05/2020 | **Corregido:** 27/11/2020 | **Aceptado:** 20/03/2021

Cita sugerida: O. E. Capuñay Uceda and J. M. Antón Perez, "Influencia de SCRUM en los plazos de entrega y rendimiento en los proyectos de las asignaturas de Desarrollo de Software," *Revista Iberoamericana de Tecnología en Educación y Educación en Tecnología*, no. 29, pp. 36-42, 2021. doi: 10.24215/18509959.29.e4

Esta obra se distribuye bajo **Licencia Creative Commons CC-BY-NC 4.0**

Resumen

El objetivo de la presente investigación fue la de determinar la influencia de SCRUM en los plazos de entrega y rendimiento en los proyectos de las asignaturas de Desarrollo de Software de estudiantes de la Universidad Tecnológica del Perú – sede Chiclayo. La investigación fue cuasi experimental, y la muestra de estudio la conformaron 14 estudiantes que recibieron SCRUM (Grupo experimental) y 11 que no lo recibieron (Grupo control), asignados al azar a los grupos. SCRUM fue planificado para la presentación de 3 entregas de informes de los proyectos, evaluándose en cada uno de ellos la eficiencia en el tiempo de entrega, porcentaje del cumplimiento parcial y calificación por entrega; asimismo, se realizó una evaluación final que ponderó las calificaciones parciales por entrega. El resultado principal es que la aplicación de la metodología SCRUM influyó disminuyendo el tiempo promedio de la entrega 3 del informe, facilitó un mayor rendimiento y el desarrollo de un mayor porcentaje promedio en la entrega 3, y en un mayor porcentaje promedio de desarrollo de las 3 entregas. Asimismo, influyó en un mayor rendimiento promedio final de los Proyectos de desarrollo de software.

Palabras clave: SCRUM; Plazos de entrega; Rendimiento.

Abstract

The objective of this research was to determine the influence of SCRUM on delivery times and performance in the projects of the Software Development courses of students of the Technological University of Peru - Chiclayo headquarters. The research was quasi-experimental, and the study sample consisted of 14 students who received SCRUM (experimental group) and 11 who did not receive it (control group), randomly assigned to the groups. SCRUM was planned for the presentation of 3 deliveries of project reports, evaluating in each of them the efficiency in delivery time, percentage of partial compliance and qualification for delivery; likewise, a final evaluation was carried out that weighted the partial grades for delivery. The main result is that the application of the SCRUM methodology influenced by decreasing the average time of delivery 3 of the report, facilitated higher performance and the development of a higher average percentage in delivery 3, and a higher average percentage of development of the 3 deliveries. It also influenced a higher final average performance of the Software Development Projects.

Keywords: SCRUM; Delivery times; Performance.

1. Introducción

Las asignaturas de Desarrollo de Software o Desarrollo de Aplicaciones Empresariales, como se les denomina en la malla curricular de la Carrera Profesional de Ingeniería de Computación e Informática de la Universidad Tecnológica del Perú, tienen como parte de sus competencias el desarrollo de proyectos de software. En este sentido, durante los semestres académicos de los dos últimos años se ha observado poca capacidad para la gestión del tiempo por parte de los estudiantes, este factor influye mucho en la puntualidad y avance presentado en los proyectos de desarrollo de software de la asignatura.

Los métodos ágiles han demostrado su aplicabilidad y eficiencia en el desarrollo de software. Las ventajas de usar métodos ágiles incluyen tiempos cortos de ciclo de vida de desarrollo; evolución continua, inmediatez y menor costo de cambio [1, 2, 3]. Aunque algunas universidades comenzaron a incluir estos métodos en su plan de estudios, los cursos sobre prácticas para los métodos ágiles en los niveles de pregrado y posgrado son todavía pocos. En las asignaturas se desarrollan proyectos para la construcción de aplicaciones web. Muchos enfoques ágiles se han considerado ampliamente en la industria web real, pero no dentro del sistema educativo universitario [4]. Por tanto, es necesario que los enfoques ágiles deban investigarse de acuerdo con los requisitos educativos específicos para influir positivamente en el proceso de enseñanza. El enfoque ágil en las asignaturas universitarias podría mejorar las habilidades constructivas de comunicación, generar relaciones entre los estudiantes, estimular la participación activa del equipo y el aprendizaje de la motivación. También debe adoptar un desarrollo de software rápido, con alta calidad de la aplicación y gestión de requisitos fácilmente cambiante [4, 5].

En este trabajo, se realiza una investigación para evaluar el impacto del uso de métodos ágiles prácticos como SCRUM en las asignaturas de desarrollo de software.

2. Base teórica

2.1. SCRUM

La metodología de desarrollo ágil se refiere a una estrategia iterativa e incremental que involucra equipos autoorganizados que trabajan en colaboración para crear el software. El término ágil se introdujo en la década de 1990 en referencia a la flexibilidad en los sistemas de producción [6]. El Manifiesto Ágil es una afirmación formal de Cuatro valores clave y Doce principios [7, 8]. La implementación de estos valores y principios durante el desarrollo de la asignatura deben garantizar una interacción adecuada entre el desarrollador y el docente (cliente del software), lo que resultará en involucrar al cliente en la etapa de diseño y desarrollo y, en consecuencia, entregar un software que funcione. Desde la perspectiva del diseño y el desarrollo, las metodologías

ágiles deben garantizar la satisfacción del cliente en las primeras etapas, la aceptación de los cambios en los requisitos, la cooperación entre desarrolladores y clientes, atender las solicitudes de personas motivadas, las comunicaciones del equipo y la simplicidad en el proceso de desarrollo.

Scrum [1, 9, 10] es un marco iterativo e incremental para desarrollo de software, se introdujo por primera vez en 1990. La característica clave es aceptar la escasez de información al comienzo del desarrollo; donde las necesidades no pueden entenderse completamente antes de comenzar el desarrollo, y centrarse en cómo maximizar la capacidad del equipo para entregar rápidamente, responder a los requisitos emergentes y adaptarse a las tecnologías y los cambios en evolución.

El proceso comienza preparando el trabajo atrasado del producto, que es una lista priorizada de ítems necesarios para lograr el objetivo del proyecto, es preparado por el propietario del producto. Cada ítem se conoce generalmente como una historia, que presenta una descripción de la funcionalidad del producto.

Scrum estructura el desarrollo en ciclos o iteraciones llamadas Sprints. El progreso del proyecto es a través de una serie de sprints; al comienzo de cada sprint, el equipo selecciona los requisitos del cliente de una lista priorizada, se comprometen a completar los artículos al final del sprint. Los sprints tienen una duración fija y nunca se extienden, terminan en una fecha específica si el trabajo se ha completado o no. Cada sprint incluye el proceso de planificación, desarrollo y control, durante el sprint, los elementos elegidos no cambian.

2.2. Asignaturas de desarrollo de software

Las asignaturas de desarrollo de software incluyen en su evaluación el desarrollo de una aplicación web robusta. A los estudiantes se les pide que apliquen principios y métodos de análisis aplicables a la ingeniería web, y luego que implementen sus proyectos de aplicaciones web mediante la simulación de un entorno moderno del mundo real. Esto ayudaría a los estudiantes a distinguir entre varios procesos de desarrollo utilizados en asignaturas previas y la actual asignatura, e identificar los posibles inconvenientes y problemas. El proyecto se lleva a cabo durante las 16 semanas del curso, e incluye 3 presentaciones de avance del proyecto y los estudiantes deben trabajar en equipos de dos miembros.

Cada entrega tiene historias de usuario que los equipos deben desarrollar y presentar. En el caso de la primera entrega, el estudiante debe diseñar la interfaz de usuario, utilizando herramientas de diseño web. En esta entrega no se incluye ningún componente de correspondencia al desarrollo de la lógica de la aplicación. En la segunda entrega, los equipos desarrollan el módulo de administración de la aplicación, en el que al menos deben desarrollarse las páginas de gestión de registros de la base de datos de la aplicación web y la autenticación de usuarios administradores. En la tercera y última entrega,

se solicita el desarrollo de la parte lógica de la aplicación, con prioridad en la implementación del proceso principal de la aplicación y las funcionalidades complementarias.

En esta investigación se tomaron equipos de manera aleatoria, a los que les tocó trabajar bajo el marco Scrum, se les entrenó en la metodología bajo este marco, a los que no les tocó trabajar con Scrum se les pidió trabajar bajo el procedimiento tradicional.

2.3. Impacto de los métodos ágiles en la educación

Las prácticas ágiles parecen ser apropiadas para el desarrollo de software realizado en universidades y colegios como curso o proyectos de graduación de último año. Los estudiantes en cursos de programación desarrollan programas para problemas específicos; Además, en la mayoría de los proyectos universitarios es imprescindible contar con un software de trabajo de buena calidad que se entregue en poco tiempo. En estos escenarios, las prácticas ágiles ayudan al enfocarse en la programación rápida de software que cumple con los requisitos especificados.

La integración de la metodología ágil en lo académico ayudará a comprender mejor la importancia y los roles de los conceptos y prácticas ágiles [11]. En realidad, los estudiantes que trabajan juntos con una buena comunicación e interacción pueden operar a niveles más altos que cuando usan sus talentos individuales, los equipos de proyectos ágiles se centran en aumentar las competencias individuales y los niveles de colaboración.

Algunos estudios observan las prácticas ágiles utilizadas en la educación y exploran su impacto en el aprendizaje, los resultados indican una influencia positiva en la habilidad de los estudiantes, especialmente en los estudiantes con habilidades de programación débiles. Además, tiene un gran efecto en la variación promedio entre las habilidades de sus compañeros de proyecto. Además, influye positivamente en las puntuaciones más altas para la codificación y las evaluaciones finales de sus proyectos en promedio [12].

En una investigación realizada en la Universidad de Kasel, Alemania [13], los autores recomiendan la enseñanza de las metodologías ágiles utilizadas ampliamente en la industria, simulando ambientes de trabajo del mundo real, esto debido a que, trabajaren equipo durante un período de tiempo limitado y entregar un producto de alta calidad son algunas de las habilidades que los estudiantes deben adquirir durante sus estudios en la universidad.

Al monitorear un estudio de la aplicación de SCRUM en un curso de Ingeniería de Software [14], el maestro obtiene una retroalimentación mucho mejor sobre lo que han aprendido los estudiantes. Los resultados empíricos pueden ayudarlo a analizar las percepciones de los estudiantes y la capacidad de aprendizaje de los métodos ágiles, encontrar mejores formas de impartir los conceptos más importantes e identificar áreas que merecen más atención o mejora. Los maestros que planean impartir un

curso similar pueden beneficiarse de evidencia empírica bien establecida. Además menciona además, que un estudio bien diseñado, puede proporcionar a los estudiantes no solo una simulación realista de la experiencia profesional, sino que también puede introducir la ingeniería de software empírica como parte de la enseñanza de la ingeniería de software. Los estudiantes pueden aprender la importancia de la evaluación basada en evidencia de los métodos ágiles para identificar sus fortalezas y debilidades antes de implementarlos en entornos de software industrial. Al considerar cuidadosamente las diferencias entre el aula y la industria, los resultados del estudio también pueden ayudar a la industria a proporcionar evidencia preliminar del comportamiento de los equipos de desarrollo que utilizan Scrum por primera vez.

En Eslovenia, se hizo un trabajo de investigación [15] similar al descrito anteriormente, en este se indica que la enseñanza de metodologías ágiles en el ámbito académico presenta muchas dificultades y limitaciones. Este artículo describe un estudio de caso en el que se adoptó un enfoque innovador para la enseñanza de tecnologías de desarrollo de software. En este enfoque, todo el curso se diseñó para ajustarse a los principios de Scrum, de modo que los estudiantes pudieran aplicarlos mientras lo estaban aprendiendo. Además, el proyecto principal del curso se iba a desarrollar en sprints, como se propone en Scrum. Después de casi dos años utilizando este enfoque, en este artículo describimos nuestra experiencia y realizamos un análisis crítico. Observamos algunos puntos positivos, como el carácter práctico del aprendizaje con el ejemplo, y una mejor preparación de los estudiantes en cuanto a metodologías ágiles. Como puntos negativos, se destaca la imposibilidad de entregar productos completos en sprints anteriores, y algunas dificultades de interacción y colaboración. La principal conclusión del estudio es que, para que el enfoque funcione en escenario académico, fue necesaria una versión modificada de la metodología Scrum.

3. Metodología

La investigación fue de tipo cuasi experimental, constituida por 14 estudiantes cuyos proyectos de las asignaturas de Desarrollo de Software fueron desarrollados bajo la metodología ágil SCRUM que constituyeron el Grupo Experimental, además es importante remarcar que el docente hizo las veces de Product Owner en cada equipo, y 11 estudiantes que lo hicieron sin SCRUM, y que formaron el Grupo Control. La asignación de los estudiantes fue realizada al azar.

La población de estudio son los estudiantes de la Universidad Tecnológica del Perú – sede Chiclayo que llevan las asignaturas de Desarrollo de Software.

La muestra estuvo constituida por los estudiantes matriculados en el semestre 2019 – I y II, siendo un total de 25 estudiantes, que formaron parejas y evaluados individualmente; 1 estudiante trabajó sin pareja y solo uno

de ellos no presentó informe en la entrega 2. Los estudiantes que conforman la muestra, han cursado previamente asignaturas de análisis, diseño y desarrollo de software; también, tienen conocimiento de metodologías tradicionales para el desarrollo de software y de al menos dos lenguajes de programación y bases de datos, pues son parte de los contenidos de asignaturas que son requisitos para cursar la asignatura en la que se realiza el experimento.

En este experimento el docente determinó las herramientas de desarrollo, la arquitectura de la aplicación y los patrones de diseño desde el inicio de cada proyecto.

Se utilizó la prueba de Shapiro Wilk para determinar la normalidad de las variables de estudio al 99% de confiabilidad por ser la prueba más poderosa para todos los tipos de distribución y tamaño de muestra [16,17] y la prueba "t" de student para comparar promedios al 95%, considerando la variabilidad de la varianza. Cabe indicar que esta prueba corresponde a tratamientos de muestras pequeñas [18,19], como en el caso de este trabajo. Se utilizaron los programas: R y SPSS.

El enfoque de investigación inductiva se ha adoptado en este estudio, comenzando por la recopilación de datos y observaciones de casos individuales para generar ideas y teorías generales amplias. El estudio involucró las siguientes fases:

1. Realización de simulación empresarial en la vida real dentro del laboratorio de computación. A los equipos de estudiantes de las asignaturas de desarrollo de software se les asignó el desarrollo de aplicaciones web dentro de un plazo estricto, los miembros de la mitad de los equipos han aprendido los conceptos básicos de Scrum, cada equipo trabajó por separado en la construcción de la aplicación web requerida.
2. Preparar y difundir un cuestionario a los participantes; el objetivo del cuestionario es medir el esfuerzo basado en horas de diseño y programación.
3. El análisis de los participantes responde para alcanzar una comprensión más profunda y más amplia que eventualmente traería un impacto positivo en las asignaturas de desarrollo de software.
4. Se analiza los resultados obtenidos en cada una de las entregas realizadas a los largo de las 16 semanas.

Se establecieron tiempos esperados en horas para culminar cada una de las 3 entregas o informes parciales de los proyectos de desarrollo de software, se midieron las horas de trabajo que realizaron los estudiantes, los porcentajes de cumplimiento de las entregas y se calificaron cada entrega en la escala vigesimal. Se definió como Eficiencia en el tiempo de entrega a la diferencia del tiempo esperado para entregar el 100% de la entrega con el tiempo utilizado o requerido para completar el 100% de la

entrega, ambos en horas; si esta diferencia es cero, significa que la entrega fue presentada en el tiempo esperado establecido. Si es positiva, significa que la entrega se produjo antes del tiempo esperado; si es negativa, significa que hubo retraso en el tiempo esperado establecido.

Finalmente se midió el rendimiento o calificación final como el promedio ponderado de las 3 calificaciones por entrega, con pesos del 30% para las 2 primeras entregas y el 40% para la 3ra entrega.

4. Resultados

Las variables se ajustaron a distribuciones normales de acuerdo a la Prueba de Shapiro Wilk para muestras pequeñas ($p < 0.01$).

En la Tabla 1 y Figura 1, respecto al tiempo programado, se aprecia que los estudiantes del grupo experimental presentaron eficiencias promedio en las entregas 1 y 2 de $-2,42 \pm 4,25$ horas (promedio \pm desviación estándar) y $-3,95 \pm 6,53$ horas respectivamente, estadísticamente similares a los tiempos promedio de entrega observados en los estudiantes del grupo control que fueron de $-0,38 \pm 4,73$ y de $-4,24 \pm 8,55$ horas respectivamente (p -valor $> 0,05$). La eficiencia promedio de los tiempos de las 3 entregas fue menor en los estudiantes del grupo experimental ($-3,84 \pm 6,56$ h) respecto a lo observado en los estudiantes del grupo control ($-11,36 \pm 22,10$ horas), aunque esta diferencia no fue significativa. Sin embargo, la eficiencia promedio en la entrega 3, fue significativamente mayor en los estudiantes del grupo experimental ($-5,15 \pm 8,43$ horas), respecto a lo observado en los estudiantes del grupo control ($-28,83 \pm 30,13$ horas) (p -valor = $0,0276$).

Tabla 1. Eficiencia en los tiempos promedio de las 3 entregas parciales de los Proyectos de desarrollo de software respecto al tiempo programado. UTP Chiclayo

Eficiencia promedio en los tiempos de entrega (horas)	Grupo control (sin SCRUM)	Grupo experimental (con SCRUM)	p-valor
Entrega 1	-0,3752	-2,4152	0,268380
ds	4,73342	4,24511	(No sig)
Entrega 2	-4,2417	-3,9489	0,924955
ds	8,54784	6,53117	(No sig)
Entrega 3	-28,8191	-5,15195	0,027639
ds	30,12781	8,427636	(Sig)
Total	-11,36104	-3,83869	0,070738
ds	22,1035	6,5608	(No sig)

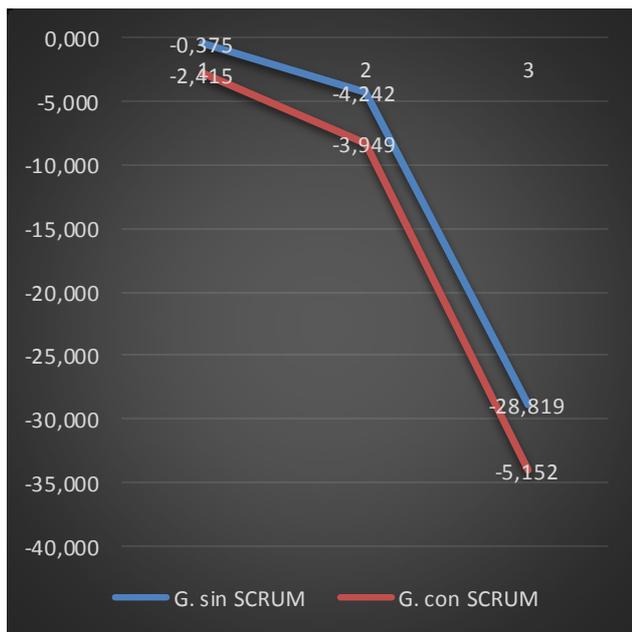


Figura 1. Tiempos promedio de las 3 entregas parciales del G. sin SCRUM vs G. con SCRUM

En la Tabla 2 y Figura 2, se observa que los estudiantes del grupo experimental presentaron calificaciones promedio en las entregas 1 y 2 de $18,21 \pm 2,75$ y $16,21 \pm 4,25$ puntos respectivamente, estadísticamente similares a las calificaciones promedio observados en los estudiantes del grupo control que fueron de $17,55 \pm 3,27$ y de $13,20 \pm 4,05$ puntos respectivamente (p -valor $> 0,05$). Sin embargo, la calificación promedio en la entrega 3, fue significativamente mayor en los estudiantes del grupo experimental ($16,07 \pm 3,81$), respecto a lo observado en los estudiantes del grupo control ($11,64 \pm 6,15$) (p -valor = $0,03686$).

Tabla 2. Calificaciones promedio de las 3 entregas parciales de los Proyectos de desarrollo de software. UTP Chiclayo

Calificaciones promedio por entrega (sistema vigesimal)	Grupo control (sin SCRUM)	Grupo experimental (con SCRUM)	p-valor
Entrega 1	17,55	18,21	0,58364
ds	3,267	2,751	(No sig)
Entrega 2	13,20	16,21	0,095
ds	4,050	4,246	(No sig)
Entrega 3	11,64	16,07	0,03686
ds	6,153	3,812	(Sig)

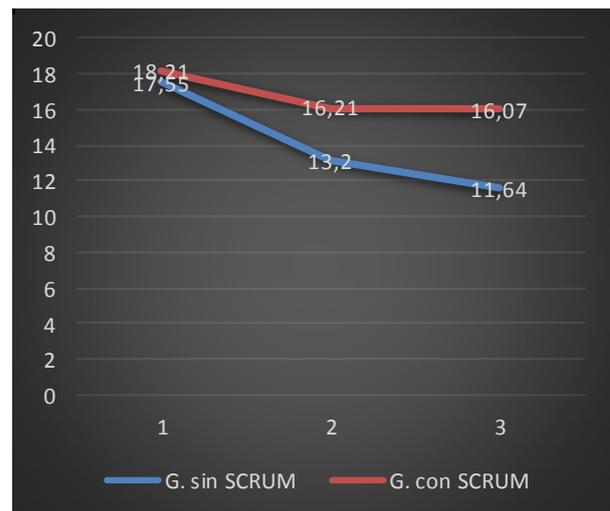


Figura 2. Calificaciones promedio de las 3 entregas parciales del G. sin SCRUM vs G. con SCRUM

En la Tabla 3 y Figura 3, se reporta que los estudiantes del grupo experimental presentaron porcentajes promedio de cumplimiento de las entregas 1 y 2 de $89,64\% \pm 17,04\%$ y $78,93\% \pm 25,21\%$ respectivamente, estadísticamente similares a los porcentajes promedio de cumplimiento observados en los estudiantes del grupo control que fueron de $86,36\% \pm 19,12\%$ y de $62,5\% \pm 24,08\%$ respectivamente ($p > 0,05$). Por otro lado, el porcentaje promedio de cumplimiento de la entrega 3, fue significativamente mayor en los estudiantes del grupo experimental ($78\% \pm 23,02\%$), respecto a lo observado en los estudiantes del grupo control ($54,09\% \pm 34,99\%$) (p -valor = $0,041132$). También se reporta que el porcentaje promedio de cumplimiento de las 3 entregas fue significativamente mayor en el grupo experimental ($82,19\% \pm 22,14\%$) respecto a lo observado en los estudiantes del grupo control ($67,81\% \pm 29,67\%$) (p -valor = $0,0195$).

Tabla 3. Porcentajes promedio del cumplimiento de las entregas parciales de los Proyectos de desarrollo de software. UTP Chiclayo

Porcentajes promedio de cumplimiento por entrega	Grupo control (sin SCRUM)	Grupo experimental (con SCRUM)	p-valor
Entrega 1	86,36%	89,64%	0,654869
ds	19,117%	17,037%	(No Sig)
Entrega 2	62,50%	78,93%	0,123749
ds	24,181%	25,205%	(No sig)
Entrega 3	54,09%	78,00%	0,041132
ds	34,987%	23,018%	(Sig)
Total	67,81%	82,19%	0,019546
ds	29,674%	22,137%	(Sig)

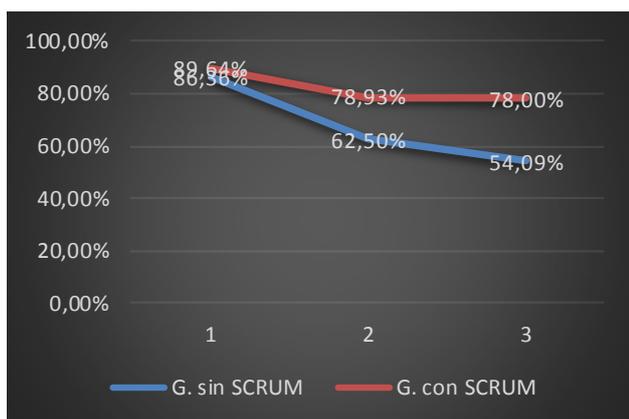


Figura 3. Porcentajes promedio del cumplimiento de las 3 entregas parciales del G. sin SCRUM vs G. con SCRUM

En la Tabla 4, se reporta que el rendimiento promedio (calificaciones finales promedio) en los estudiantes del grupo experimental ($16,86 \pm 2,32$) fue significativamente mayor a lo observado en el grupo control ($13,73 \pm 3,77$) (p-valor=0,0175).

Tabla 4. Calificaciones finales de los Proyectos de desarrollo de software. UTP Chiclayo

Rendimiento (Calificación final promedio)	Grupo control (sin SCRUM)	Grupo experimental (con SCRUM)	p-valor
□	13,73	16,86	0,017533
Ds	3,771	2,316	(Sig)

Conclusiones

SCRUM influyó significativamente, logrando una mayor eficiencia en el tiempo de la entrega 3 de los proyectos de desarrollo de software, aunque en ambos grupos se mostraron retrasos promedios en las tres entregas, siendo mayor en el grupo sin SCRUM.

SCRUM influyó significativamente en una mayor calificación y porcentaje del cumplimiento de la entrega 3 de los proyectos de desarrollo de software, que incluye la parte más especializada y extensa como lo es el desarrollo de la parte lógica de la aplicación, con prioridad en la implementación del proceso principal aplicativo y las funcionalidades complementarias.

SCRUM también influyó significativamente en el mayor porcentaje promedio de cumplimiento de las 3 entregas.

SCRUM es una metodología ágil que logró mejorar significativamente el rendimiento o calificación final de los proyectos de desarrollo de software.

Referencias

[1] A. K. Kamepally and T. Nalamothu, "Agile Methodologies in Software Engineering and Web

Engineering," *I.J. Education and Management Engineering*, vol. 5, pp. 1-14, 2016.

[2] A. McDonald and R. Welland, *Agile Web Engineering (AWE) Process*. 2001.

[3] B. V. Ramana, M. Salman, S. Abdul Moiz and M. Sharfuddin, "Applying Agile And Web Engineering Technique For Successful Web Management," in *Proceedings of the 4th National Conference*, INDIACom-2010, 2010.

[4] R. V. Roque Hernández, J. A. Herrera Izaguirre, A. López Mendoza and J. M. Salinas Escandón, "A Practical Approach to the Agile Development of Mobile Apps in the Classroom," *Innovación Educativa*, vol. 17, no. 73, 2017.

[5] T. Besley and M. A. Peters Eds., *Re-imagining the Creative University for the 21st Century*. Sense Publishers, 2013.

[6] V. M. M. Thilak, S. R. Devadasan and N. M. Sivaram, "A Literature Review on the Progression of Agile Manufacturing Paradigm and Its Scope of Application in Pump Industry," *The Scientific World Journal*, 2015.

[7] H. K. Flora and S. V. Chande, "A review and analysis on mobile application development processes using agile methodologies," *International Journal of Research in Computer Science*, vol. 3, no. 4, pp. 9–18, 2013.

[8] Manifesto for Agile Software Development. [Online]. Available: <http://agilemanifesto.org>.

[9] M. Almseidin, K. Alrfou, N. Alnidami and A. Tarawneh, "A Comparative Study of Agile Methods: XP versus SCRUM," *International Journal of Computer Science and Software Engineering (IJCSSE)*, vol. 4, no. 5, 2015.

[10] V. Gunga, S. Kishnah and S. Pudaruth, "Design of A Multi-Agent System Architecture For The Scrum Methodology," *International Journal of Software Engineering & Applications (IJSEA)*, vol.4, no.4, 2013.

[11] J. Bergin, J. Bergin, T. Reichlmayr, J. Caristi, G. Pollice, "Agile development in computer science education: practices and prognosis," in *Proceedings of the 36th SIGCSE Technical Symposium on Computer Science*, 2005.

[12] G. I. U. S. Perera, "Impact of using agile practice for students software projects in computer science education," *International Journal of Education and Development using Information and Communication Technology (IJEDICT)*, vol. 5, no. 3, pp. 85-100, 2009.

[13] A. Scharf and A. Koch, "Scrum in a software engineering course: An in-depth praxis report," in *2013 26th International Conference on Software Engineering Education and Training (CSEE&T)*, San Francisco, CA, 2013, pp. 159-168, doi: 10.1109/CSEET.2013.6595247.

[14] V. Mahnic, "A Capstone Course on Agile Software Development Using Scrum," in *IEEE Transactions on*

Education, vol. 55, no. 1, pp. 99-106, Feb. 2012, doi: 10.1109/TE.2011.2142311.

[15] S. D. Zorzo, L. de Ponte and D. Lucrédio, "Using scrum to teach software engineering: A case study," in *2013 IEEE Frontiers in Education Conference (FIE)*, Oklahoma City, OK, 2013, pp. 455-461, doi: 10.1109/FIE.2013.6684866.

[16] N. Mohd and Y. Bee. "Power comparisons of Shapiro- Wil, Kolmogorov-Smirnov, Lilliefors and Anderson Darling tests," *Journal of Statistical Modeling and Analytics*, vol.2, no.1, pp. 21-33, 2011.

[17] B. Yazici and S. Yolacan, "A comparison of various tests of normality," *Journal of Statistical Computation and Simulation*, vol. 77, no. 2, pp. 175-183, 2007, doi: 10.1080/10629360600678310

[18] D. Anderson, D. Sweeney and T. Williams, *Statistics for administration and economy*. CENGAGE Learning, 2008, pp. 402-403.

[19] J. Freund, I. Miller and M. Miller, *Estadística Matemática con aplicaciones*. Pearson Education: México, 2000, pp.284-285.

Información de Contacto de los Autores:

Oscar Efraín Capuñay Uceda

Esquina Prol. Augusto B. Leguía con Av. Herman Meiner
Chiclayo
Perú

C16762@utp.edu.pe

ORCID iD: <https://orcid.org/0000-0002-4145-6309>

Juan Manuel Antón Pérez

Urb. La Purísima Mz. 31 Lote 18
Chiclayo
Perú

C16712@utp.edu.pe

ORCID iD: <https://orcid.org/0000-0002-9665-779X>

Oscar Efraín Capuñay Uceda

Nació en Monsefú, Lambayeque, Perú. Se graduó como Ingeniero de Sistemas en la Universidad Nacional Pedro Ruiz Gallo. Es docente investigador en la Facultad de Ingeniería de la Universidad Tecnológica del Perú.

Juan Manuel Antón Pérez

Maestro en Estadística Aplicada, docente universitario de pre y pos grado en cursos de Estadística y de Metodología de la Investigación Científica. Ponente de temas estadísticos en eventos nacionales e internacionales.