

PLATAFORMA DE APRENDIZAJE COLABORATIVO BASADO EN LEARNING ANALYTICS

G. Viguera¹, F. Rodríguez², D. Viguera³, M. Cerrato⁴, G. Román¹, A. Herranz¹
M. Jiménez¹, V. Rampérez¹, J. Soriano¹

1: Dpto. de Lenguajes y Sistemas Informáticos e Ingeniería del Software

Escuela Técnica Superior de Ingenieros Informáticos

Universidad Politécnica de Madrid

e-mail: {gviguera, groman, aherranz, mjimenez, vramperez, jsoriano}@universidad.es

2: Fundación IMDEA Software

e-mail: francy.rodriguez@imdea.org

3: Dpto. de Psicología y Sociología

Universidad de Zaragoza

e-mail: diegojviguera@gmail.com

4: Facultad de Educación

Universidad Internacional de Cataluña

e-mail: mcerratolara@uic.es

Resumen. *Las metodologías ágiles para el trabajo colaborativo, utilizadas en la industria, han llevado a adoptar estas metodologías en la enseñanza de Ingeniería de Software. El plan de estudios ha evolucionado recientemente para introducir el uso del desarrollo ágil, de modo que los estudiantes practiquen su uso y capaciten las habilidades necesarias para el trabajo colaborativo basado en proyectos. Las metodologías ágiles se caracterizan por ser iterativas e incrementales, con ciclos cortos, entregas constantes y un alto nivel de interacción entre los miembros del equipo. De acuerdo con estas características, se ha desarrollado un entorno de aprendizaje colaborativo con soporte informático (CSCL) para ayudar a los académicos a evaluar y proporcionar comentarios a los estudiantes. El entorno CSCL se basa en la plataforma colaborativa GitLab, adaptada para implementar conceptos de la metodología SCRUM, permitiendo además recopilar automáticamente información sobre el trabajo individual y en equipo de los estudiantes.*

Palabras clave: Learning Analytics, Aprendizaje Colaborativo, Evaluación del Aprendizaje, Psicología del Aprendizaje, Uso de las TIC.

1. Introducción

El éxito del uso de metodologías ágiles para el trabajo colaborativo en la industria ha llevado a adoptar estas metodologías para enseñar Ingeniería de Software. El plan de estudios ha evolucionado en los últimos años para introducir el uso del desarrollo ágil, de modo que los estudiantes practiquen su uso y capaciten las habilidades necesarias para el trabajo colaborativo basado en proyectos. Las metodologías ágiles se caracterizan por ser iterativas e incrementales, con ciclos cortos, entregas constantes y un alto nivel de interacción entre los miembros del equipo. Estas características constituyen un desafío para los educadores y estudiantes, ya que, en cortos períodos de tiempo, es necesario evaluar y proporcionar retroalimentación al trabajo individual y grupal, con respecto a aspectos como el uso de la metodología, la

gestión de herramientas y la colaboración dentro del equipo [1, 2]. Por esa razón, se ha desarrollado un entorno de aprendizaje colaborativo con soporte informático (CSCL) para ayudar a los académicos a evaluar y proporcionar comentarios a los estudiantes. El entorno CSCL se basa en la plataforma colaborativa GitLab, que se ha adaptado para implementar conceptos asociados a SCRUM, una metodología ágil ampliamente adoptada. Además, el uso de GitLab permite recopilar automáticamente información sobre el trabajo individual y en equipo de los estudiantes. Utilizando los datos recopilados de GitLab, se ha desarrollado una plataforma de Learning Analytics para analizar el trabajo grupal e individual durante la ejecución de proyectos de estudiantes utilizando SCRUM. El objetivo es determinar si SCRUM ayuda a los estudiantes a elaborar un mejor software, evaluando la adopción de la metodología y la calidad del software resultante. Un prototipo de la plataforma fue desarrollado y utilizado en un curso de pregrado de Ingeniería de Software en una universidad española, en el que 79 estudiantes divididos en grupos de 3-4 personas desarrollaron dos proyectos independientes. Los resultados preliminares muestran que el entorno CSCL propuesto ayuda a proporcionar información para evaluar y dar retroalimentación a los estudiantes. Además, los datos recopilados por el entorno CSCL mostraron una buena correlación de la adopción de SCRUM por parte de los estudiantes y la calidad del software resultante.

2. Estudio realizado

El entorno CSCL propuesto se ha evaluado durante las prácticas de una asignatura relacionada con Ingeniería del Software. Para ello se propuso a los alumnos la realización de dos proyectos utilizando SCRUM. En cada proyecto, se desarrolló un producto de software diferente. Se formaron grupos de 3-4 personas, para asignar los diferentes roles SCRUM. El rol del propietario del proyecto (PO), enfocado en el negocio, está a cargo de transmitir la visión del proyecto al equipo, formalizando y priorizando los requisitos. El rol de SCRUM Master (SM), cuyos objetivos son: liderar el equipo, garantizar que se cumplan las reglas y el proceso, gestionar los riesgos y trabajar con el PO para maximizar la calidad del producto. Finalmente, el rol de desarrollador está a cargo de las tareas de desarrollo con el conocimiento técnico necesario para generar el producto.

En SCRUM, un sprint es la unidad temporal para representar un ciclo de desarrollo o iteración. En nuestro caso, cada sprint tiene una duración de una semana. Por otro lado, cada requisito de software consiste en una descripción no formal, llamada historia. Antes de cada sprint, el PO presenta las historias priorizadas, luego el equipo decide colectivamente con cuántas y con cuáles pueden comprometerse. En base a esto, el equipo realiza la planificación de la tarea requerida para completar las historias y se reúne regularmente para sincronizar, verificar el proceso de desarrollo y analizar posibles problemas. Debido al período de tiempo inicial del curso, el número de sprints para cada proyecto se establece en 5 para el primero y en 4 para el segundo.

En este estudio se ha medido el nivel de adopción de SCRUM y se ha evaluado el trabajo de los alumnos. La evaluación de dicho trabajo se realiza asignando los siguientes pesos a cuatro aspectos diferentes: metodología (40%), funcionalidad (50%), calidad del código (5%) y usabilidad (5%). Además, se definen cuatro categorías para la evaluación de la metodología: uso adecuado de roles, retroalimentación, confiabilidad de la planificación y actividades de desarrollo. A su vez, cada categoría se evalúa en función de las métricas, cuyos valores se obtienen automáticamente a través de la plataforma de análisis de aprendizaje al consultar el marco de GitLab.

Por otro lado, el nivel de adopción de SCRUM se obtiene a través de la Plataforma de Análisis de Aprendizaje. La información recopilada automáticamente y las gráficas

generadas por la plataforma, ayudan en la tarea de evaluar la adopción de la metodología de desarrollo. Concretamente, la actividad de desarrollo recopilada por la plataforma se combina para generar tres tipos de gráficos: la actividad individual de cada alumno dentro de su grupo, el nivel de adopción de los roles SCRUM por parte de los alumnos y la distribución temporal y la medición cualitativa / cuantitativa de cada grupo actividad. Debido a limitaciones de espacio, solo se muestra un ejemplo del último tipo de gráfico en la Figura 1. Así, la Figura 1 muestra la distribución temporal de la actividad de desarrollo para cada grupo al trazar, por sprint, los elementos de desarrollo y SCRUM actividad (es decir, reuniones, historias de usuarios, características, errores y otros).

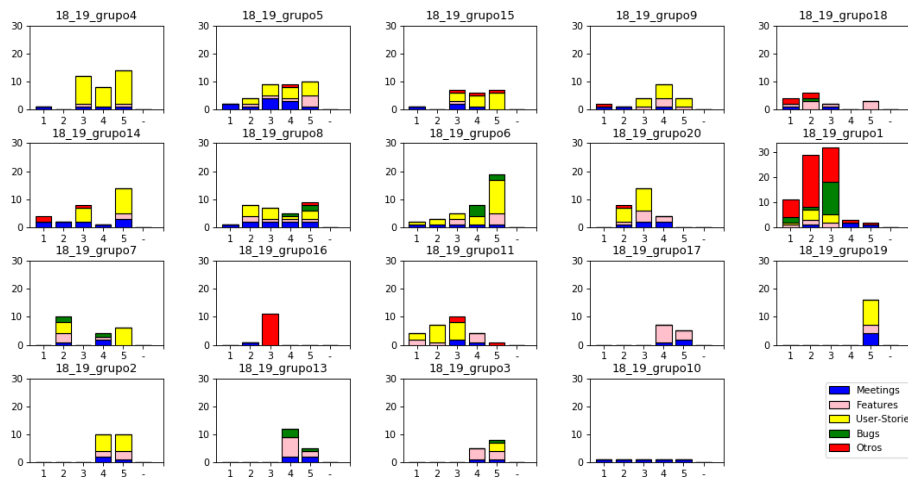


Figura 1. distribución del tiempo y datos cualitativos/cuantitativos de la actividad de grupo.

3. Resultados

Utilizando los datos de GitLab recopilados a través de la Plataforma de Learning Analytics, el análisis estadístico se ha basado en un estudio de correlación al comparar el nivel de adopción de SCRUM por los estudiantes con la calidad del software resultante desarrollado por los estudiantes. Los resultados se muestran en forma de gráfico de correlación. La Figura 2 muestra la correlación obtenida para los dos proyectos desarrollados durante el curso. El diagrama de la izquierda corresponde a los resultados del primer proyecto y el diagrama de la derecha corresponde a los resultados del segundo proyecto. Los diferentes puntos en la Figura 2 representan un grupo de prácticas. La etiqueta asociada a cada punto es el número de grupo, que se muestra con fines de seguimiento. Además, los gráficos en la Figura 2 se complementan con una línea, para mostrar la tendencia de correlación.

Se pueden extraer diferentes conclusiones de la Figura 2. Por un lado, se puede observar el nivel de correlación entre el uso de SCRUM, medido como grado SCRUM, y la calidad del software desarrollado, medido como grado de proyecto. Al observar ambas gráficas, existe una correlación casi perfecta, ya que los equipos con un bajo nivel de adopción de SCRUM han dado como resultado un software con mala calidad, en términos de funcionalidad, calidad de código y usabilidad.

Por otro lado, la Fig. 2 también muestra más relaciones entre grupos con un mal y buen uso de SCRUM. Los grupos de estudiantes se pueden clasificar en dos conjuntos: uno con una calificación SCRUM menor o igual a 2 (considerado malo) y otro con una calificación SCRUM mayor a 2 (considerado bueno). Por lo tanto, el conjunto de grupos con mala calificación SCRUM para el primer proyecto sería: 3, 13, 14 y 18 (ver diagrama de la izquierda en la Figura 2). El mismo conjunto para el

segundo proyecto estaría compuesto por grupos: 1, 2, 13, 14 y 18. Por otro lado, el conjunto de grupos con buena calificación SCRUM para el primer proyecto sería: 1, 4, 5, 6, 7, 8, 9 y 15. El mismo conjunto para el segundo proyecto estaría compuesto por grupos: 4, 5, 6, 7, 8, 9 y 15.

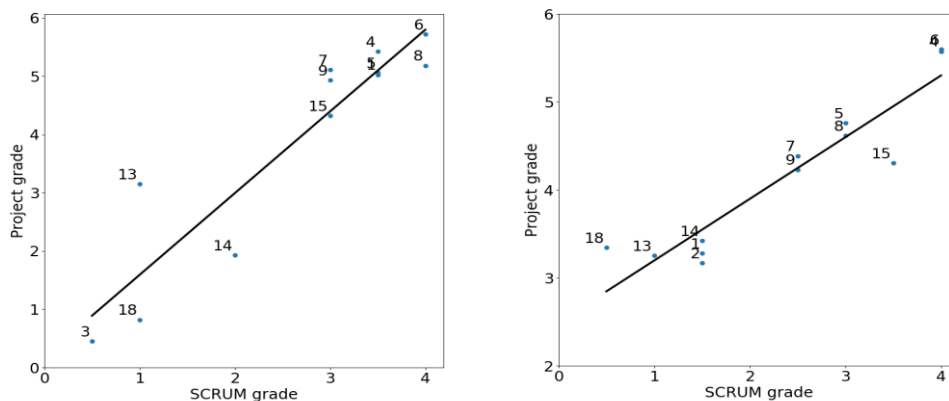


Figura 2. Correlación entre los resultados de SCRUM y del código desarrollado.

Desde una perspectiva cuantitativa y de acuerdo con los conjuntos descritos anteriormente, la proporción de grupos con calificación SCRUM mala: buena sería 4: 8 para el primer proyecto y 5: 7 para el segundo proyecto. El hecho de que estas proporciones sean similares en todos los proyectos muestra que el número de grupos con un buen uso de SCRUM en el primer proyecto, continuaron utilizando la metodología en el segundo proyecto mientras obtenían una buena calificación del proyecto.

El análisis cualitativo y cuantitativo de la Figura 2 también muestra que los grupos que usan SCRUM no aumentaron del primer proyecto al segundo proyecto. Además, de los veinte grupos formados para participar en el curso, solo doce grupos participaron realmente en el primer y segundo proyecto. Dejamos como trabajo futuro el análisis de las posibles razones detrás de este comportamiento. Sin embargo, como varios estudiantes declararon informalmente, el calendario y las tareas académicas, realizadas simultáneamente con este curso, fueron una limitación importante para usar SCRUM correctamente.

4. Conclusiones

En este trabajo se ha propuesto un entorno de aprendizaje colaborativo con soporte informático (CSCL) para ayudar a los académicos a evaluar y proporcionar comentarios a los estudiantes. El entorno CSCL se basa en la plataforma colaborativa GitLab, adaptada para implementar conceptos de la metodología SCRUM, permitiendo además recopilar automáticamente información sobre el trabajo individual y en equipo de los estudiantes. Los datos recopilados han permitido constatar una buena correlación positiva entre el uso de SCRUM en el aula y el desempeño de los alumnos.

Referencias

- [1] Anslow, C., & Maurer, F. (2015). An Experience Report at Teaching a Group Based Agile Software Development Project Course, (Cmmi), 500–505. <https://doi.org/10.1145/2676723.2677284>
- [2] Mahnic, V. (2012). A capstone course on agile software development using scrum. IEEE Transactions on Education, 55(1), 99–106. <https://doi.org/10.1109/TE.2011.2142311>