Actas de las Jenui, vol. 7. 2022.
Páginas: 111-118

Ayuda para enseñar *Scrum* con o sin programar

Javier J. Gutiérrez Rodríguez
Departamento de Lenguajes y Sistemas Informáticos
Universidad de Sevilla
Sevilla
javierj@us.es

Francisco Gracia Ahufinger
Gerente de Proyectos
Derivco
Córdoba
fran.grahu@gmail.com

Resumen

Scrum es una de las propuestas más utilizadas en el desarrollo de software y forma parte de la mayoría de los temarios de carreras relacionadas con las tecnologías de la información. En el 2020 realizamos un estudio sobre cómo se enseñaba *Scrum* en las universidades a partir de artículos publicados en JENUI. Las conclusiones nos motivaron a presentar las principales dificultades a la hora de enseñar *Scrum*, y un conjunto de puntos y sugerencias para enseñar *Scrum* al alumnado completamente alineada con la documentación oficial de *Scrum*. Además, compartimos nuestras propias experiencias enseñando *Scrum* en más de una década. Los resultados sirven de guía para la preparación de la docencia de *Scrum* en asignaturas que impliquen actividades de programación y en las que no tengan estas actividades.

Abstract

Scrum is one of the most widely used approaches and is part of most IT-related degree programmes. In 2020 we conducted a study on how *Scrum* was being taught in universities based on evidence found mainly in articles published in JENUI. The conclusions motivate us to present, in this paper, the main problems when teaching *Scrum*, and a set of points and suggestions for teaching *Scrum* to students aligned with the official *Scrum* documentation. In addition, we share our own experience teaching *Scrum* over more than a decade.

Palabras clave

Scrum, enseñanza universitaria, *product goal*, inspección, adaptación.

1. Motivación

Scrum continúa siendo la propuesta más utilizada para desarrollo ágil de software [15]. La *15th Annual State Of Agile Report* realizó 1,382 encuestas sobre técnicas y practicas *Agile* durante 2021 [1]. El 66% de sus encuestados utilizaban *Scrum*. Si tenemos en cuenta otras variantes de *Scrum*, por ejemplo, combinado con *Kanban*, este porcentaje sube a 81%.

En el año 2020, los autores realizamos un estudio sobre cómo se enseñaba *Scrum* en asignaturas de las universidades españolas [6]. Mientras preparábamos este estudio, tuvimos la oportunidad de hablar con todos los autores citados, conocer con más profundidad cómo enseñaban *Scrum* en sus asignaturas y descubrir que, en la mayoría de los casos, existían diferencias con la Guía de *Scrum*.

Motivados por este descubrimiento, y por haber incurrido nosotros mismos durante este período en algunas de estas diferencias al aplicar *Scrum*, presentamos en este trabajo un conjunto de principios y sugerencias para la elaboración de prácticas para asignaturas en las que se enseñe en *Scrum.* El objetivo es que permitan al alumnado entender el porqué de esa manera de trabajar, las ventajas que le puede aportar y cómo acercar la práctica de *Scrum* a la realidad de una empresa.

Como complemento a este artículo, presentamos un artefacto que muestra una posible práctica de programación con *Scrum* disponible en [7]. No es objeto del presente artículo presentar una evaluación de los artefactos de Scrum en la práctica de la asignatura para ello consultar [6] y [7], sino el compartir las experiencias extraidas tras este período.

La organización de este trabajo se describe a continuación. En las secciones dentro de esta introducción, se define qué es *Scrum*. La sección 2 expone las principales dificultades que hemos encontrado en nuestra experiencia con prácticas de *Scrum* en el contexto de una asignatura sin programación.

La sección 3 expone los principios a incluir en cualquier práctica sobre *Scrum*. La sección 4 expone nuestro trabajo enseñando *Scrum* sin programación desde 2012. La sección 5 analiza trabajos relacionados. Finalmente, la sección 6 expone las conclusiones.

1.1. ¿Qué es *Scrum*?

La documentación oficial de *Scrum* es la "*Scrum Guide*" [15]. La versión actual define *Scrum* como "un marco ligero que ayuda a las personas, los equipos y las organizaciones a generar valor a través de soluciones adaptativas para problemas complejos". Vamos a analizar tres puntos de esta definición.

El primer punto es que *Scrum* es un marco de trabajo ligero, no una metodología, ni proceso ni conjunto de pasos a seguir. No se puede seguir al pie de la letra, sino que necesita una adaptación. En este artículo exponemos una adaptación para docencia.

El segundo punto es que el objetivo de *Scrum* es generar valor y, si utilizamos *Scrum* para desarrollar software, este valor es aportado por el software. Más adelante se menciona cómo incluir la idea de aportar valor en prácticas de *Scrum*.

El tercer punto es que *Scrum* está orientado a soluciones adaptativas para problemas complejos. Un problema complejo consiste en ambigüedad e incertidumbre, interdependencia, no linealidad, condiciones locales únicas, autonomía, comportamientos emergentes y límites no fijos.

No se pueden aplicar reglas predefinidas ni identificar relaciones causa y efecto [11]. *Scrum* propone inspección y adaptación continuas para buscar soluciones a problemas complejos. Aplicar *Scrum* en problemas simples no requiere aplicar inspección y adaptación.

Por ejemplo, implementar una lista de requisitos ya definida de antemano y que no va a cambiar, no es un problema complejo. Implementar una lista de requisitos y medir el avance del proyecto por el número de requisitos implementado tampoco es un problema complejo.

1.2. ¿Es *Scrum* ágil?

Entre los autores del "Manifesto for Agile Software Development" (llamado Manifiesto a partir de aquí) figuran los dos autores de Scrum. Esto invita a pensar que aplicar Scrum al desarrollo de software implica desarrollar software de manera ágil. Esto no tiene por qué ser cierto.

El Manifiesto son cuatro valores y doce principios. Si dichos valores y principios están presentes en una implantación del marco de referencia *Scrum* entonces sí es correcto afirmar que aplicamos *Scrum* para desarrollo de software ágil, pero si no están presentes, aunque se aplique *Scrum*, no es correcto indicar que se desarrolla software de manera ágil.

Por ejemplo, un valor del manifiesto es "Individuos e iteraciones sobre procesos y herramientas". Si centramos la docencia de *Scrum* en celebrar un conjunto de reuniones tal y como *Scrum* lo indica no estamos aplicando el manifiesto y utilizando *Scrum* en un contexto de agilidad, ya que las reuniones, llamadas eventos en la Guía de *Scrum*, son mecanismos de inspección y adaptación y es tarea de los participantes utilizarlas para tal fin.

Consideramos necesario que cualquier docente que aborde la enseñanza de *Scrum*, haga la reflexión de si va a centrarse únicamente en *Scrum* o va a tener también en cuenta el concepto de agilidad y modificar su docencia al respecto.

2. Dificultades en la enseñanza de *Scrum* en el ámbito universitario

A partir de nuestra experiencia enseñando *Scrum* desde el curso 2012 y conversaciones con otros docentes para preparar [6], hemos sintetizado las cinco dificultades principales que nos ayudan a entender qué es *Scrum* y cómo abordar una práctica de *Scrum*.

La primera dificultad es que *Scrum* no trata sobre reuniones, perfiles y artefactos, sino inspección y adaptación para aportar valor en problemas complejos, tal y como dice la Guía, ya que se basa en una gestión empírica.

Por ejemplo, que un equipo se reúna todos los días quince minutos o trabaje durante unas semanas a las que llamemos *Sprint*, hasta un evento al final de esas semanas al que llamemos *Sprint Review*, no sirve de nada. *Scrum* es un marco de transparencia, inspección y adaptación porque estas herramientas son necesarias para resolver problemas complejos. Aplicar los eventos o artefactos sin entender cuál es el objetivo, es lo mismo que aprender piano memorizando la secuencia de teclas a pulsar.

La segunda dificultad es la atomización del trabajo. Un equipo profesional dedica todos sus días laborables al desarrollo de un proyecto o de un número pequeño de proyectos.

En cambio, nuestro alumnado tiene clase de una asignatura uno o dos días a la semana, durante pocas horas, incrustada en una agenda que obliga al alumnado a cambiar completamente de contexto antes y después. Esto dificulta mucho el trabajo continuo y el poder ver cómo se aplica *Scrum* en el largo plazo.

La tercera dificultad es la dificultad de traer el mundo real a la docencia. No contamos con clientes con necesidades reales, no hay necesidad de desplegar lo hecho, ni con software listo para que el usuario pueda usarlo, etc. Marcos de trabajo, como *Scrum*, orientados a potenciar el valor de lo desplegado y trabajar con clientes reales sufren a la hora de aplicarse en un entorno educativo.

La cuarta dificultad es el número de alumnos y la carga excesiva de trabajo del docente. Si no se tiene un número de alumnos que permita interactuar con ellos, no se puede enseñar *Scrum* de la manera más fiel, y hay que empezar a buscar soluciones de compromiso.

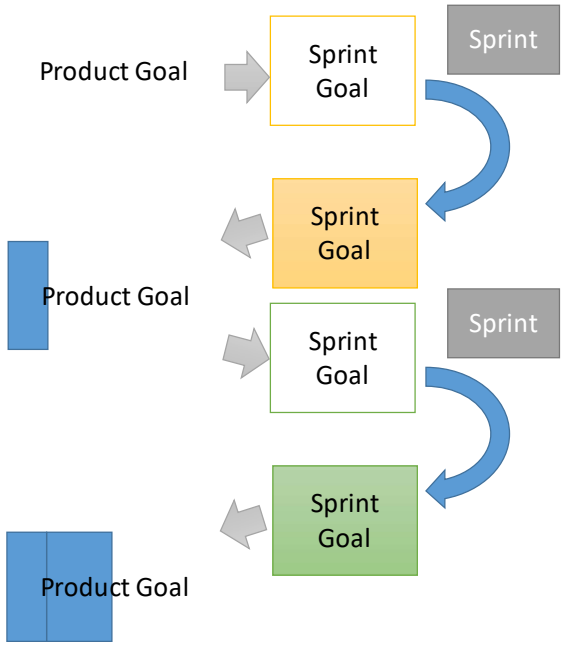


Figura 1: Aportaciones de valor al Product Goal a partir de los Sprints Goals de cada Sprint

Identificamos una dificultad adicional, que surge al enseñar *Scrum* en el contexto de una asignatura en la que no hay programación.

En la ingeniería del software, *Scrum* aporta más valor cuando se usa para desarrollar software. Esta dificultad se aborda con más detalle en la sección 5 dónde se proponen alternativas para asignaturas sin programación basadas en la experiencia de los autores.

A continuación, en la siguiente sección, planteamos una lista de principios que debe tener una práctica basada en *Scrum* que esté alineada con la definición del propio *Scrum*.

3. *Scrum* con perspectiva pedagógica

Hemos mencionado varias veces la palabra valor, pero, ¿qué es valor en *Scrum*? Valor es aquello por lo que el usuario o cliente está dispuesto a pagar dinero, si bien en una organización sin ánimo de lucro, valor puede ser todo aquello que aporte un beneficio a la sociedad.

Para impartir una práctica que transmita la definición de *Scrum* como un marco para aportar valor en problemas complejos, y supere las dificultades vistas en la sección anterior, proponemos cuatro puntos que, como lista de verificación, debe cumplir una asignatura. Estos puntos son:

1. El trabajo a realizar debe contar con un *Product Goal* cuantificable que se vaya completando mediante los *Sprints Goals* alcanzados en cada *Sprint* (figura 1).

2. Cada *Sprint* debe tener un *Sprint Goal*. El grado de cumplimiento del *Sprint Goal* al final del *Sprint* debe contribuir al avance del *Product Goal*.

3. El evento de *Sprint Review* debe orientarse a ver cómo se ha avanzado hacia el *Product Goal* y planificar los próximos *Sprints Goals*.

4. Los grupos de prácticas deben hacer investigación y propuestas de qué desarrollar para alcanzar el *Sprint Goal* y contribuir al *Product Goal*.

A continuación, se da más contexto para los puntos anteriores.

Como se ha visto en la sección 1, *Scrum* se orienta a problemas complejos. En un problema complejo no existe una persona que tenga la solución del problema. Por ese motivo, ni el *Product Owner* debe ser quien diga qué hay que implementar ni se puede tener una lista de requisitos cerrada para trocearla según el número de *Sprints*.

Los conceptos de *Product Goal* y *Sprint Goal* están definidos en la guía de *Scrum*. De manera muy resumida, un "*Goal*" es un estado futuro que aporta valor. El *Product Goal* es una evolución de producto que aporta valor a clientes y usuarios.

El *Sprint Goal* es un incremento de producto que empuja a este a cumplir su *Product Goal*. Por ejemplo, implementar una cantidad de funcionalidad, no es ni un *Product Goal* ni un *Sprint Goal*, por lo que no aporta valor contar requisitos implementados.

Los cuatro puntos anteriores se complementan con un conjunto de sugerencias que se describen en las siguientes secciones. Los ejemplos de estos puntos se incluyen en el artefacto que complementa este artículo [4].

3.1. *Sprint* y entrega de valor continua

Sugerimos no esperar al final del *Sprint* para hacer entregas de valor. En el desarrollo de software el valor se aporta mediante software funcionando que clientes y usuarios pueden utilizar.

Scrum no es un proceso de entrega por lotes, no se prepara un lote de funcionalidad para entregarlo al final del *Sprint*.

Hacerlo de esta manera dificulta la inspección y adaptación y convierte el evento de *Sprint Review* (mencionado al principio de esta sección) en una revisión de pantallas y funcionalidad que no aporta valor.

Funcionalidad terminada que está a la espera del fin de *Sprint* es funcionalidad que no aporta valor y *Scrum* busca aportar valor.

También sugerimos evitar que el *Sprint Goal* y el trabajo del *Sprint* sea simplemente implementar un número de historias o requisitos. Llevar contabilidad de historias, requisitos o puntos de historia implementados apenas aporta valor, y no tiene utilidad en problemas complejos, por lo que pueden omitirse de cualquier práctica.

3.2. *Sprint Planning* y *Sprint Backlog*

El evento *Sprint Planning* consta de tres partes: definición de *Sprint Goal*, selección de ítems del *Product Backlog* que aporten al *Sprint Goal* y definición del *Sprint Backlog*.

Recomendamos que la primera parte se realice en clase y sea realizada principalmente por los docentes en calidad de *Product Owner*, y abriendo la participación al alumnado, ya que ellos deben entender y estar comprometidos con el *Sprint Goal* resultante. También se habla en clase sobre los items para que todos tengan una visión compartida de qué significan y, sobre todo, como aportan valor al *Sprint Goal*.

En un problema complejo, el *Product Backlog* no puede contener todos los ítems necesarios porque no se conocen a priori (o no sería un problema complejo). Existe un espacio de incertidumbre para probar soluciones y utilizar inspección y adaptación. Recomendamos no intentar tener un conjunto completo de items en el *Product Backlog*, sino transmitir a los alumnos que los items son incompletos y ellos tienen libertad para añadir trabajo no relacionado con ningún item del *Product Backlog* si consideran que puede contribuir al *Sprint Goal*. También tienen la posibilidad de proponer items para el *Product Backlog* durante el evento o seguir consultando al profesorado en su papel de *Product Owner*. Un ejemplo de *Product Backlog* está en el artefacto de este artículo [4].

Por último, sugerimos transmitir a los alumnos que su *Sprint Backlog* tampoco es una lista de tareas por hacer. El objetivo es conseguir el *Sprint Goal* con independencia de que todas las tareas se realicen o queden tareas pendientes. Una vez más, el equipo inspecciona y se adapta. Es conveniente que se vayan de clase con un primer *Sprint Backlog*, necesariamente incompleto porque es un artefacto vivo actualizable día a día, pero desarrollado con el apoyo de los docentes.

3.3. *Product Backlog* e *Ítems*

Sugerimos considerar otros tipos de *Product Backlog* ítem además de historias de usuario (o no usarlas en absoluto). *Scrum* indica que el *Product Backlog* debe contener todo lo necesario para la mejora del producto, pero no menciona las historias de usuario ni estas son obligatorias ni en *Scrum* ni en un desarrollo de software ágil.

Cuando nos referimos a historias de usuario, no nos referimos a ítems como: "incluir una opción de búsqueda por fecha" o "incluir una opción de ordenar resultados por orden alfabético", ni a frases escritas en una tarjeta o en una lista. Nos referimos a historias basadas en el trabajo de los usuarios o las expectativas del usuario, dónde los desarrolladores tengan que investigar y trabajar en buscar la manera de aportarles valor, usando la inspección y adaptación mencionada en la sección 2. Por ejemplo: “como investigadores, una de las necesidades que tenemos es estar al día de los últimos artículos publicados sobre determinados temas porque requiere una cantidad de tiempo considerable tener que hacer búsquedas periódicas por palabras clave o revisar todos los artículos publicados en algunos ‘*journals*’ para ver si descubrimos un artículo nuevo que nos resulte interesante”.

El caso anterior es un buen ejemplo de una historia de usuario; tenemos un usuario y nos cuenta la historia de su trabajo. También podría utilizarse como *Sprint Goal* ya que se podrían cuantificar tiempos, esfuerzos y espacios de búsqueda, y abre la posibilidad a la investigación (¿cómo mejorarlos?). El objetivo del equipo en el *Sprint* no es implementar una funcionalidad tras otras como meros codificadores seudoautomáticos, sino investigar qué alternativas existen, cuáles encajan mejor en el *Sprint Goal* y qué software que funcione pueden poner en producción (o en un entorno que pueda ser usado) durante el *Sprint*.

Un *Product Backlog* debe contener el *Product Goal*, posibles *Sprints Goals*, aspectos del sistema para tener en cuenta, etc. No encaja en una única estructura organizada en forma de tabla con columnas. No recomendamos utilizar una hoja de cálculo, ni ningún otro artefacto tipo tabla de una única dirección para un *Product Backlog*. En su lugar, recomendamos utilizar estructuras de dos dimensiones, como pueden ser un *Story Map* [13], que muestre el contexto del sistema y los usuarios, y se pueda explorar de manera global, considerando todo el trabajo de los distintos usuarios y posibles impactos del *Sprint Goal* o incluso un mapa mental.

3.4. *Product Owner* y *Scrum Master*

Proponemos que las funciones de *Product Owner* sean asumidas por el profesorado. Su principal misión es ayudar al alumnado a entender el *Product Goal*, explicar cómo lo que los alumnos han desarrollado aporta al *Sprint Goal* y al *Product Goal* y trabajar con el alumnado las historias de usuario u otros ítems del *Product Backlog* sobre lo que el usuario quiere hacer, pero no cómo implementarlo en el sistema, para dar al alumnado la posibilidad de que inspeccionen y adapten, y evitar la primera dificultad expuesta en la sección 2.

Si se quiere mostrar cómo *Scrum* puede aplicarse a un desarrollo ágil, es importante transmitir la idea de trabajar en colaboración con clientes y usuarios y que el *Product Owner* no sea un muro que aísle a usuarios de desarrolladores.

En más de una década de trabajo con *Scrum*, la función del *Scrum Master* se ha implementado de

maneras diferentes, aunque no excluyentes. Por ejemplo, el *Scrum Master* puede ser un perfil que trabaje con la organización para maximizar el encaje y rendimiento de equipos *Scrum* o puede ser un experto tecnológico que apoye mediante mentorización a los equipos. Debido a las características y dificultades de la docencia ya comentados en la sección 2 y, en base a nuestra experiencia, este perfil queda muy desdibujado en una práctica universitaria. Por este motivo, proponemos que se aborde desde una perspectiva más técnica que organizacional. Proponemos que las funciones de *Scrum Master* son asumidas por un alumno de un grupo de prácticas, pudiendo rotar este perfil de *Sprint* a *Sprint*. El alumno que asuma este perfil debe continuar colaborando con su grupo, es decir, el trabajo de *Scrum Master* es un complemento a su trabajo de prácticas, y en ningún momento es de dedicación completa.

3.5. *Daily* Meeting y *Retrospectives*

Scrum está orientado a un único equipo trabajando en un único proyecto, con el mismo horario laboral de manera colaborativa. En la docencia universitaria esto no sucede, como se mencionó en la sección 2.

El evento *Daily* Meeting no tiene por qué ser una reunión, ni tiene por qué limitarse a contestar un conjunto de preguntas predefinidas. La *Daily* es una herramienta de inspección y adaptación diaria, principalmente orientada a que el trabajo fluya para conseguir el *Sprint Goal* y se detecten impedimentos a este flujo lo antes posible para poder abordarlos.

La *Daily* con un grupo de alumnos con varias prácticas, asignaturas y horarios y hábitos diferentes es difícil de aplicar y muy difícil de aplicar con resultados positivos. Nuestra recomendación es transmitirle que tienen que estar en contacto periódico para hacer visible el avance del proyecto, esto pueden hacerlo por salas de chat asíncronas (Telegram, Slack, etc,). Durante las retrospectivas se puede ayudar a madurar esta práctica.

Las retrospectivas son una de las principales herramientas para el equipo, pero, por nuestra experiencia, apenas aportan valor al alumnado, por las características particulares del alumnado que ya hemos comentado en la sección 2.

Llevamos realizando retrospectivas en clase desde el curso 2018 - 19 y hemos presentado nuestros resultados en JENUI [8].

En nuestra experiencia la autonomía y capacidad de evolución de grupos de usuarios es baja, por ese motivo recomendamos que las retrospectivas se celebren en horario de clase, sean facilitadas por los docentes de la asignatura, y se orienten a seguir entendiendo el por qué de la manera de trabajar con *Scrum* y a potenciar que los alumnos apliquen introspección y adaptación.

3.6. Evaluación de prácticas de *Scrum*

Una evaluación directa se puede hacer por el valor aportado, de la misma manera que se evaluaría en el mundo real. Sin embargo, no recomendamos que esto sea el único mecanismo de evaluación. En el momento en el que nos enfrentamos a problemas complejos en los que hay que emplear inspección y adaptación, que puede verse similar a prueba y error, existe la posibilidad de que el trabajo no converja a buenos resultados. Y eso, no es un fracaso, no debe ser punible en un entorno real y no debe tener un impacto en la nota en una evaluación.

Desde nuestra primera práctica en 2012, nos dimos cuenta que necesitábamos conocer la evolución del *Product Backlog* o lo que había sucedido en el evento *Sprint Review* para evaluar la práctica adecuadamente.

Esto nos motivó a pedir a los alumnos un diario de desarrollo (como el de la figura 2) en el que incluyeran todo el trabajo hecho que no fueran los resultados pedidos en la práctica. Los alumnos era libres de elegir el formato digital en el que realizar dicho diario (documento compartido, página web tipo wiki, canal en herramienta de mensajería, etc.). Además, cabe puntualizar, que dicho artefacto no era objeto de evaluación.

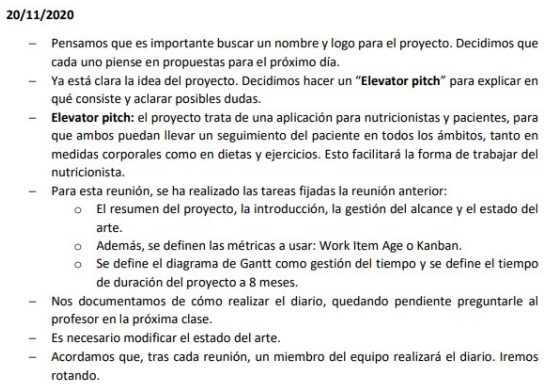
Este diario, igual que un diario común, tiene sus fechas, resumen del trabajo hecho, artefactos de gestión, reuniones que hayan podido hacer, descartes, etc. Este diario nos da el contexto del trabajo y podemos entender el trabajo realizado y evaluarlo con independencia de los resultados obtenidos.

Con los puntos y sugerencias vistas en esta sección, se ha cubierto la práctica totalidad de la guía de *Scrum* de una manera en la que se puede aplicar para la definición de prácticas y ejercicios.

4. Prácticas de *Scrum* sin programación

El origen de *Scrum* fue el desarrollo de productos electrónicos innovadores y, actualmente, se documenta su uso en contextos distintos del desarrollo del software [4]. Por tanto, sí debería ser posible desarrollar prácticas de *Scrum* sin programar siempre que se cumplan los puntos vistos en la sección 3. En el artefacto que acompaña a este artículo se muestra en detalle un ejemplo de práctica de programación con *Scrum*.

Esta sección explora otras alternativas sin programación, cómo utilizando dinámicas relacionadas con el juego, los alumnos pueden experimentar aplicar *Scrum* en el contexto de avolucionar un producto, sin que este implique la elaboración de un software y mediante el trabajo colaborativo en equipo.



20/11/2020

- Pensamos que es importante buscar un nombre y logo para el proyecto. Decidimos que cada uno piense en propuestas para el próximo día.
- Ya está clara la idea del proyecto. Decidimos hacer un "**Elevator pitch**" para explicar en qué consiste y aclarar posibles dudas.
- **Elevator pitch:** el proyecto trata de una aplicación para nutricionistas y pacientes, para que ambos puedan llevar un seguimiento del paciente en todos los ámbitos, tanto en medidas corporales como en dietas y ejercicios. Esto facilitará la forma de trabajar del nutricionista.
- Para esta reunión, se ha realizado las tareas fijadas la reunión anterior:
  - El resumen del proyecto, la introducción, la gestión del alcance y el estado del arte.
  - Además, se definen las métricas a usar: Work Item Age o Kanban.
  - Se define el diagrama de Gantt como gestión del tiempo y se define el tiempo de duración del proyecto a 8 meses.
- Nos documentamos de cómo realizar el diario, quedando pendiente preguntarle al profesor en la próxima clase.
- Es necesario modificar el estado del arte.
- Acordamos que, tras cada reunión, un miembro del equipo realizará el diario. Iremos rotando.

Figura 2: Frangmento de diario de desarrollo del curso 2020 - 21

4.1. Experiencia de los autores

La experiencia en prácticas de *Scrum* de los autores ha sido en asignaturas sin programación. En el curso 2012 – 13 nuestra práctica consistía en simular el trabajo de un *Sprint* pidiéndole a los alumnos que escribieran requisitos funcionales detallados a partir de historias de usuario. Esta propuesta, estaba muy alejada de los puntos vistos en la sección 3 por lo que transmitía de manera muy pobre los conceptos de la Guía de *Scrum*. En estos primeros años, durante el evento *Sprint Review*, pedíamos a los alumnos alguna variación de los resultados entregados en el *Sprint*. En años posteriores cambiamos el enfoque ya que entendemos que la *Sprint Review* no tiene como objetivo revisar los artefactos hechos en el *Sprint*, sino inspeccionar la consecución de objetivos y planificar cómo seguir para alcanzar los objetivos planteados.

En el curso 2016 - 17, se cambió la práctica para que el trabajo no fuera escribir requisitos funcionales, sino realizar mock-ups de interfaces gráficas. Este cambio no tuvo ningún impacto positivo ya que seguíamos repitiendo los errores ya comentados en la sección 2. En los cursos 2019 – 20, 2020 – 21 y en el 2021 - 22, se eliminó la práctica de *Scrum* como una práctica independiente. En su lugar, se propuso que el alumnado utilizara *Scrum* para redactar un pliego de memoria técnica para una contratación pública. Este no es el contexto más adecuado para mostrar *Scrum* ya que, por ejemplo, falta la interacción con usuarios.

4.2. Prácticas con LEGO®

En cursos antes del COVID, se le ha propuesto al alumnado la posibilidad de hacer prácticas de *Scrum,* fuera de horario lectivo, utilizando LEGO® y las dinámicas definidas en [9]. LEGO® permite trabajar de una manera similar al código fuente, creando nuevos elementos en un *Sprint*, modificándolos y recombinándolos, descartándolos si ya no se necesitan, etc, con el objetivo de crear un producto final.

En nuestras prácticas, les pedimos al alumnado que construyera un campus universitario de alto rendimiento. Como preparación, les presentábamos potenciales usuarios del campus, principalmente alumnos y profesores, indicando sus estudios, actividades, aficiones, peticiones, etc. En cada *Sprint* les pedíamos que potenciaran un aspecto del campus. A continuación, se explica cómo se aplican los 4 puntos expuestos en la sección 3 a esta práctica de *Scrum* con LEGO®. El Product Goal cuantificable es la cantidad de alumnado del campus. Para ello, el campus debe tener servicios que atraigan a distintos tipos de estudiantes. Cada Sprint Goal puede estar orientado a hacer el campus más atractivo para un tipo de estudiante. Según el incremento o decremento de estudiantes del campus, el alumnado conoce si lo que ha investigado funciona o no.

Los *Sprints Review*s están orientados a explicar qué aportan al *Sprint Backlog* los edificios desarrollados. Por ejemplo, en el *Sprint Review* de un *Sprint* con *Sprint Goal* de atraer a estudiantes deportistas, un equipo había dedicado una gran cantidad de tiempo y recursos a desarrollar una piscina cubierta y otro equipo había desarrollado un óvalo.

En el *Sprint Review* se comprobó que la piscina cubierta era algo que el equipo pensaba que iba a funcionar, pero no había alumnos nadadores entre los alumnos potenciales que se usaron de referencia. El óvalo resultó ser una pista de atletismo que sí encajaba con los deportes practicados por la mayoría de alumnos. Sin embargo, su nivel de desarrollo era insuficiente, por lo que se planteó la posibilidad de evolucionarlo en el siguiente *Sprint*.

Otro ejemplo, a veces, grupos de alumnos incluyen una cafetería como parte de lo que es imprescindible para impartir clase y, en posteriores *Sprints*, dicha cafetería se mejora o incluso se divide en varios edificios (cafetería, comedor, heladería) para potenciar el *Sprint Goal* de dicho *Sprint*. Estás prácticas son largas, mínimo 2 horas, y fácilmente pueden llegar a 6 horas. Hemos probado a parar la práctica y retornarla otro día dejando el aula las construcciones LEGO® ya realizadas, y ha funcionado bien. Esta manera de trabajar no la podemos aplicar a nuestra docencia reglada ya que no podemos hacerlo con el número de alumnos matriculado en la asignatura, lo cual se ha mencionado en la sección 2 como una de las principales dificultades a la hora de plantear una práctica de *Scrum*.

5. Análisis de trabajos relacionados

El artículo [12] plantea la dificultad de formar buenos *Scrum Master* y su experiencia con un curso

de un semestre orientado a formar en este rol de *Scrum*. Una de sus principales recomendaciones es contar con el apoyo de profesionales. Esa recomendación la hemos aplicado los autores de este trabajo incorporando como coautor a un perfil profesional con amplia experiencia. Aunque en este trabajo no recogemos explícitamente el contar con profesionales con experiencia, es sin duda un gran valor añadido y sugerimos que se aplique siempre que sea posible.

El propósito de [5] es analizar *Scrum* en aprendizaje basado en proyectos en la educación superior. La asignación de tarea, el monitoreo del desempeño, la gestión visual y la retroalimentación (*feedback*) regular se consideraron las principales ventajas, y *Scrum* tuvo un impacto positivo en el desempeño de los estudiantes. Los estudiantes reconocen el rol del *Scrum Master* y *Product Owner* como vital para guiar a los equipos de una manera sostenible.

En este artículo defendemos una aproximación diferente en el que el rol de *Scrum Master* y *Product Owner* trabaje con los equipos para que ellos sean autogestionados (en el sentido que tengan libertad para decidir cómo aportar valor a los usuarios del sistema) y puedan abordar retos complejos de manera autónoma aportando valor, en lugar de un equipo que se limite a implementar el requisito pedido por el *Product Owner*. Como se vio en la sección 1, esta segunda manera de trabajar está más alineada con la definición y objetivos de *Scrum*.

El artículo [14] presenta un marco para la enseñanza del curso de gestión de proyectos de tecnologías de información con contenido tradicional basado en la propuesta de gestión de proyectos del *Project Management Institute* (sin desarrollo de software) al tiempo que presenta *Scrum* como la lógica organizativa para realizar el trabajo del curso. Este marco adapta las prácticas de *Scrum*. Este artículo está muy alejado de lo que proponemos en esta propuesta ya que aplica *Scrum* en un entorno predictivo, en el que los *Sprints* están compuestos de listas de requisitos a implementar, estimaciones del número de requisitos que caben en un *Sprint*, etc., mientras que en este trabajo defiende aplicar inspección y adaptación para abordar problemas complejos.

La sección 4 ha expuesto que simulaciones de *Scrum* con LEGO® son una buena alternativa para plantear prácticas en las que se practique *Scrum* cuando no hay posibilidad de programar. Existen muchas referencias sobre LEGO® y *Scrum* en la docencia, por ejemplo, a continuación, se citan 3 publicadas entre 2020 y 2021. El artículo [3] expone una experiencia con 198 participantes que la valoraron como significativa, relevante y valiosa. Además, cuatro *Scrum Master*s indicaron que esta propuesta proporciona una representación realista de proyectos *Scrum* del mundo real; que sea dinámico, complejo, desafiante y motivador, y que se aprende haciendo. La referencia [10] presenta un juego de LEGO® para presentar conceptos y principios básicos de *Scrum* durante una única clase. La referencia [2] presenta el caso real de una actividad en el aula para enseñar conceptos de *Scrum* con LEGO®. Al final de las clases, los resultados mostraron que los juegos dinámicos y las actividades palpables son más efectivas que las lecciones teóricas o en video.

6. Conclusiones

Este artículo ha aportado un enfoque realista de la aplicación de *Scrum* en el contexto docente de realización de prácticas sin programación. Al igual que en el mundo profesional, *Scrum* no se aplica de la misma manera en una empresa u otra, incluso en el desarrollo de dos productos distintos dentro de la misma empresa, nuestra experencia nos dice que la aplicación de *Scrum* en el contexto docente tampoco será homogénea y que, por tanto, tal y como la Guía nos indica, la base de *Scrum* será la inspección y adaptación al contexto concreto de aplicación y se fundamentará en tres pilares; la experiencia del *Product Owner*, la experiencia del *Scrum Master* y el nivel de autoorganización y de autogestión del *Scrum Team*. Por tanto, nuestra recomendación para llevar a cabo una buena práctica de *Scrum* es seguir las guías que hemos presentado.

Scrum puede intentar aplicarse en desarrollos en los que exista un conjunto de requisitos definidos de antemano, se establezca un tiempo y coste fijo, existan penalizaciones si no se entrega una cantidad de funcionalidad cada cierto tiempo, y se apliquen otras restricciones muy habituales, por ejemplo, en contratación pública o en contratos llave en mano. Y funciona porque *Scrum* está construido alrededor de un ciclo de vida iterativo e incremental. Sin embargo, en ese contexto, las ventajas de usar *Scrum* son muy similares a no usarlo y una mala aplicación de *Scrum* o aplicarlo sin entender por qué es así y por qué se hace lo que se hace tiene un impacto muy pequeño.

Scrum está pensado para aplicar inspección y adaptación como manera de abordar problemas complejos, por lo que para aprenderlo y practicarlo tenemos que crear espacio para inspeccionar y adaptar y tenemos que buscar un problema complejo. Al enfrentarnos a un problema complejo, descubrimos que no aporta valor un *Product Backlog* lleno de requisitos para implementar, no aporta valor un *Product Backlog* que contenga información detallada de qué deben implementar los equipos, no aporta valor celebrar eventos *Scrum Daily* en las que se pregunta a cada miembro del equipo de manera

mecánica qué hizo ayer, qué está haciendo hoy y si tiene algún problema y no aporta nada llevar una contabilidad de historias, puntos de historia, horas ideales, etc., ya que el objetivo es aportar valor. *Scrum* ni propone ni defiende nada de lo anterior.

Los nuevos retos en ingeniería informática, desde computación cuántica, a criptomonedas, *blockchain*, web 3.0, metaversos, etc., encajan en problemas complejos dónde está todo por descubrir, por lo que una inspección y adaptación rápida en ciclos es imprescindible.

Como continuación de este trabajo, esperamos escalar la inspección y adaptación más allá de *Scrum* y del contexto de una asignatura hasta un nivel de plan de estudio.

Agradecimientos

Esta investigación ha sido apoyada por la Agencia Estatal de Investigación (AEI) a través del proyecto NICO PID2019-105455GB-C31 del Ministerio de Economía y Competitividad y Competitividad y a través del proyecto SmartAuditor (P20_00644) de la Consejería de Consejería de Economía, Conocimiento, Empresa, y Universidades de la Junta de Andalucía.

Referencias

[1] *15th Annual State of Agile Report. 2021.* https://digital.ai/resource-center/analyst-reports/state-of-*Agile*-report

[2] Douglas Augusto Barcelos Bica y Carlos Alexandre Gouvea da Silva. *"Learning process of Agile Scrum methodology with lego blocks in interactive academic games: Viewpoint of students." IEEE Revista Iberoamericana de Tecnologías del Aprendizaje* 15.2 (2020): 95-104.

[3] Simon Bourdeau, Alejandro Romero-Torres, and Marie-Claude Petit. *"Learning Scrum: A LEGO®-Scrum Simulation." Agile Scrum* Implementation and Its Long-Term Impact on Organizations. IGI Global, 2021. 169-189.

[4] Emilio López Cano, Juan Manuel. García-Camús, Javier Garzás, Javier Martínez Moguerza, y Noemí Navarro Sánchez. *A Scrum-based framework for new Product development in the non-software industry. Journal of Engineering and Technology Management* - JET-M, 61, julio 2017. https://doi.org/10.1016/j.jengtecman.2021.101634

[5] Sandra Fernandes, José Dinis-Carvalho y Ana Teresa Ferreira-Oliveira. *"Improving the performance of student teams in Project-based learning with Scrum." Education Sciences* 11.8 (2021): 444.

[6] Javier Jesús Gutiérrez. *Scrum se Escribe Scrum (no SCRUM), y Otras Ideas para Mejorar su Docencia.* 2020. https://personal.us.es/javierj/articulos/files/2020-JavierGutierrez-*Scrum*.pdf

[7] Javier Jesús Gutiérrez. Artefacto: https://personal.us.es/javierj/articulos/2022jenui.html

[8] Javier Jesús Gutiérrez, José González Enríquez, Virginia Cid-de-la-Paz, Leticia Morales, Andrés Jiménez. *Retrospectivas en el aula. Una experiencia práctica*. XXV Jornadas Sobre La Enseñanza Universitaria de La Informática (JENUI), 4, 295–302. Junio 2019.

[9] Alexey Krivitsky. *"Lego4Scrum 3.0: A Complete Guide to Lego4Scrum - A Great Way to Teach the Scrum Framework and Agile Thinking.".* Amazon Digital Services LLC (2017).

[10] Stan Kurkovsky. *"A simple game to introduce Scrum concepts."* Proceedings of the 51st ACM Technical Symposium on Computer Science Education. 2020.

[11] Andreas Nachbagauer. *Managing complexity in Projects: Extending the Cynefin framework. Project* Leadership and Society, 2, 100017. 2021. https://doi.org/10.1016/j.plas.2021.100017

[12] Maria Paasivaara. *"Teaching the Scrum Master Role using Professional Agile Coaches and Communities of Practice."* 2021 IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET). IEEE, 2021.

[13] Jeff Patton y Peter Economy. *User story mapping: discover the whole story, build the right Product.* " O'Reilly Media, Inc.", 2014.

[14] Daniel Evan Rush y Amy J. Connolly. *"An Agile framework for teaching with Scrum in the IT Project Management classroom." Journal of Information Systems Education* 31.3 (2020): 196-207.

[15] Ken Schwaber y Jeff Sutherland. *The Scrum Guide. The Definitive Guide to Scrum: The Rules of the Game*. Noviembre 2020. https://*Scrum*guides.org/