

Asistentes:

- Fernando Andrés Galindo (inicialmente ausente)
- Jesús Aparicio Ortiz (presentador)
- Matthew Bwye Lera
- Manuel Guerra Cordon
- Fernando José González García
- Francisco Murillo Prior

Agenda:

- Discusión del día semanal para el Daily Scrum
- Revisión y análisis de Product Backlog para la construcción del Sprint Backlog
- ¿Qué entendemos por una tarea terminada?
- Herramientas a utilizar durante el sprint
- Objetivos del sprint

Debate:

Iniciamos la reunión planteando qué cosas deberíamos cubrir, y reflexionando en qué capacidad de cambio en la planificación del sprint tendríamos en los Daily Scrums. Planteamos una serie de preguntas para el profesor acerca de qué cambios son aplicables a la planificación del sprint durante el sprint y qué cosas podrían discutirse en los Daily Scrums, y hablamos de qué día de la semana elegiríamos para Daily Scrums. Entre todos determinamos que el mejor día era los jueves, pero que el día elegido podía ser flexible. **Actualización versión 2:** Después de la primera reunión, entendimos que se haría siempre un Daily Scrum durante la clase práctica de la asignatura. Eso son los viernes. Tener un Daily Scrum un jueves y otro un viernes no tiene sentido ya que en la reunión del viernes habría poco que añadir, por lo que decidimos mover la reunión de los jueves a los domingos, los lunes, o los martes. Ganó la opción de los domingos. Sin embargo, luego hablamos con el profesor, y llegamos a la conclusión de que, en primer lugar, eran demasiadas pocas reuniones a la semana y que, en segundo lugar, colocar dos reuniones en fines de semana y ninguna entre semana haría que los Daily Scrums perdieran utilidad. Por tanto, replanteamos el sistema y decidimos tener un Daily Scrum cada dos días entre semana, es decir, una reunión los lunes a las 17:30, otra los miércoles a las 16:45, y otra los viernes durante la clase, de esa forma podríamos plantear problemas y resolverlos a lo largo de la semana con agilidad.

Jesús inició la presentación del Product Backlog. Discutimos acerca de las distintas tareas, y rápidamente marcamos como hecha la tarea de creación del repositorio. Aprovechamos para introducir un poco la interfaz de Github a Francisco. Acto seguido, empezamos a discutir el reparto de tareas del apartado A2.2.3 del Product Backlog. Tras un corto debate, se decidió que Francisco y Fernando Andrés se encargasen de las tareas c y d, ya que les iban a ayudar a acostumbrarse a los ficheros del framework Spring. El resto de las tareas se repartieron de forma equitativa entre los demás miembros.

Hablamos de los siguientes puntos del producto backlog. Hablamos de la branching strategy que seguiríamos para la realización de las tareas, donde decidimos que aplicaríamos la misma que la que se emplea en la asignatura DP2, ya que así nos estaríamos acostumbrando y experimentando en una estrategia que nos veremos obligados a usar en dicha asignatura,

además de que así no tendríamos que pensar en dos estrategias distintas para dos asignaturas distintas, lo que nos ayudaría a agilizar el trabajo. Mencionamos la política de versiones, aunque aún no hemos llegado a conclusiones.

Para cerrar el punto de la revisión del backlog, revisamos las tareas del apartado A2.3.3. Fernando Andrés y Francisco no cursaron DP1, por lo que tienen menos conocimiento de trabajar en el framework Spring. Ellos trabajaron con Manuel con la asignatura IR, y él sí cursó DP1, por lo que nos pareció razonable hacer un grupo de ellos 3, y que se encargasen de unas tareas que vimos oportunas para que aprendieran a utilizar el framework. La carga de trabajo de este sprint es suficientemente pequeña como para que unir a tres en un mismo grupo fuese problemático. La tarea a fue especialmente difícil de dividir. En primer lugar, determinamos qué atributos y clases nuevas habría que crear. Una vez determinado eso, dividimos las tareas entre la creación de las clases, la creación de una vista y validaciones del formulario de creación, y la creación de las pruebas unitarias de la nueva clase creada. Además, debatimos si debíamos crear una entidad habitación; la conclusión a la que llegamos es que en principio no era necesario, pero que podía hacerse ya que probablemente fuese necesario crearla en algún momento si se decidiera ampliar alguna funcionalidad de este sistema. En conclusión, decidimos que lo mejor era consultar con el Product Owner. Durante el reparto y división de tareas, decidimos usar un criterio de clasificación de tareas para distinguir entre tareas independientes (es decir, tareas que no requieren que se hagan otras tareas de forma previa) y dependientes (tareas que requieren que se haya hecho otra tarea antes de realizarse esta). Marcamos las tareas dependientes con un label en Github llamado "Dependant". Este criterio de clasificación nos ayudó a repartirnos el resto de las tareas entre los miembros de forma que todos los miembros tenían al menos una tarea independiente y al menos una dependiente.

Luego debatimos sobre qué entendíamos por tarea terminada. Primero se encaminó a que un trozo de código supere las pruebas unitarias que le corresponden, pero luego decidimos ampliar un poco el concepto. Decidimos crear una nueva columna en nuestro proyecto de github, llamado "Untested", por el que debería pasar cualquier tarea no trivial antes de ser considerado "Done". Las tareas que se encuentran en esa columna necesitan ser revisadas por otros miembros. El cómo dejar constancia de esa revisión se hablará en otro momento. Decidimos que, de forma irregular, los distintos miembros del equipo debían revisar el proyecto en busca de fallos, por ejemplo, al realizar pull. Una vez las tareas están acabadas y los miembros del equipo se disponen a realizar la documentación relacionada con el sprint, se debe realizar una revisión global del proyecto en busca de tareas inacabadas. **Actualización versión 2:** A raíz de una recomendación del profesor, concretamos un poco la revisión de las tareas, de tal forma que ya no debe ser irregular, ya que esto causaría bastantes problemas y podría terminar en no revisándose nada nunca, o tener a pocos miembros revisando siempre el trabajo de los demás, además del suyo propio. Para cada tarea del Product Backlog, se creará una tarea de comprobación que será asignada a alguien que no haya trabajado en la tarea principal. Además, hemos decidido anticipar la revisión global, no se realizará al final del todo, sino varios días antes del fin del sprint, de tal forma que dé tiempo a revisar el sprint entero y a aplicar cambios si fuese necesario. Lógicamente, esto implica que se deben terminar todas las tareas antes del final oficial del sprint, por lo que debemos adelantar el deadline para que dé tiempo a revisar bien las tareas y a aplicar cambios.

A continuación, hablamos un poco de las herramientas de comunicación que usaríamos durante el sprint. Durante la reunión informal de presentación que tuvimos para conocernos, se habló de usar Slack, pero tras documentarse, el grupo entero decidió abandonar la idea ya que Slack no soporta canales de voz como Discord. La decisión fue utilizar Discord tanto para conversaciones por voz como por escrito, ya que se pueden organizar las conversaciones bastante bien en distintos canales. Dentro del entorno de trabajo, se tiene habilitado un grupo de WhatsApp para contactar en momentos puntuales, pero la principal intención del grupo es para hablar en carácter informal.

Fernando Andrés llegó a la reunión y se le informó de lo hablado, haciendo recopilación de todo lo que se ha dicho, que ayudó a completar el boceto del acta.

Para terminar la reunión, se habló de los objetivos que perseguíamos en este sprint. En un principio no pensamos más allá que en completar todas las tareas, pero luego se añadieron una serie de objetivos que se habían hablado en momentos puntuales de esta reunión y la previa. La conclusión final fue que los objetivos eran completar todas las tareas, y acostumbrarnos y sacar conclusiones de las estrategias empleadas, de las herramientas usadas, del aprendizaje del grupo, y del trabajo en equipo que hiciéramos. Estos objetivos son especialmente importantes en este sprint ya que tiene menos peso que los que están por venir, por lo que queremos darle especial énfasis a nuestra mejoría como equipo en varios sentidos.

Conclusiones:

- Decidimos realizar un Daily Scrum semanal los jueves a las 5 de la tarde, que es el día que mejor le viene a la mayoría. **Actualización versión 2:** Decidimos ampliar el número de Daily Scrums, en primer lugar, todos los viernes se harían Daily Scrums durante la hora de clase y, además, decidimos añadir dos más, uno el lunes a las 17:30, y otro el miércoles a las 16:45.
- Se hizo un reparto de tareas teniendo en cuenta especialmente qué aspectos deberían fortalecerse en el grupo.
- Decidimos emplear la branching strategy que se emplea en DP2, porque así no tenemos que cambiar de esquema de trabajo entre dos asignaturas, lo que reduce el esfuerzo del grupo, y porque así nos acostumbramos más rápidamente a este esquema de trabajo para ambas asignaturas. Además, queremos usar este primer sprint para ver si nuestro grupo trabaja bien con esta estrategia.
- Usaremos desde ahora la etiqueta "Dependant" para tareas que dependan de otras, y trataremos que, en la medida de lo posible, las tareas dependientes e independientes estén equitativamente distribuidas entre los miembros del grupo.
- Para verificar que una tarea está terminada, hemos decidido crear una nueva columna en el proyecto de github, llamada "Untested", por la que deben pasar todas las tareas no triviales antes de considerarse terminadas. Para considerarse terminada, debe superar una prueba de usuario y, en caso de que sea posible, una prueba unitaria. **Actualización versión 2:** Se ha concretado un poco más en qué consiste la prueba que deben superar tareas no triviales. Posicionamos un sprint deadline propio al menos

Acta de Sprint Planning

11/03/2021 (Versión 1)

Versión 2

13/03/2021 (Versión 2)

media semana antes del deadline oficial, para dejar esos días para analizar las tareas y aplicar cambios que consideremos oportunos. Además, se añadirá una tarea de comprobación para cada tarea del Product Backlog al realizar el sprint planning.

- Hemos contemplado 4 herramientas de comunicación: Discord, WhatsApp, Slack y Gitter. Hemos ponderado la utilidad que ofrecen cada una de ellas respecto a la comodidad que ofrece su uso. Discord será la herramienta que más utilizaremos, aunque sabemos que puede inducir a más distracción que las demás, pero ofrece sistema de voz, cosa que los otros sistemas no ofrecen (o al menos no ofrecen con tal calidad).
- La herramienta de gestión que usaremos es Github. Usaremos la branching strategy que se utiliza en DP2.
- Los objetivos de este sprint son:
 - Completar todas las tareas del Sprint Backlog.
 - Acostumbrarnos y sacar conclusiones de:
 - Las estrategias empleadas, para determinar si usarlas en futuros sprints o no, o para pensar en mejores formas de aplicarlas.
 - Las herramientas usadas, para determinar si replantearnos el uso de dichas herramientas o en formas de optimizar su efectividad.
 - Trabajo en equipo, para ver con qué miembros trabajamos mejor, comprobar de primera mano los puntos fuertes y débiles de cada miembro y conocernos mejor.

Duración total de la reunión: 2 horas

Duración total de replanning: 1 hora