

Informe Académico

Equipo 2

Irazoqui - Castiñeiras - Begino - Turchi - Costa

Reflexiones

Sprint Planning Meeting

A la hora de estimar esfuerzo de tareas y SP de las US, al principio no aclaramos bien en base a que estábamos estimando exactamente: si la dificultad del prototipado, o la dificultad de la tarea real. Eso llevó a que algunas tareas al principio tuvieran estimaciones mucho más grandes que otras similares en otras iteraciones

Algo que se notó mucho en las sprint review meeting de la primera iteración es la falta de coordinación en varias cosas, por ejemplo en cuanto a los diseños del prototipado. Lo que nos parece importante a aprender es que las planning meeting no son solo para estimar y completar las salidas del sprint (DoD, Sprint Goal, etc.), sino que también es un momento para discutir otros temas relevantes que involucren a todos los desarrolladores.

Para las estimaciones decidimos usar la herramienta de planning poker. Nos pareció una herramienta muy útil para poder estimar las historias de usuario por varias razones. Primero, nos permitió que las estimaciones de uno no se vean sesgadas por las de otros, ya que la votación es secreta y luego se revela. Segundo, se puede visualizar bien cuales son las distintas votaciones, y cuánta gente votó por ellas. Gracias a esto siempre sabíamos a quién escuchar cuando una de las votaciones variaba mucho con respecto a otras. Tercero, repetir la votación hasta que hubiera una mayoría o decisión unánime, nos aseguro que estuviéramos todos de acuerdo.

Sprint Retrospective Meeting

Para las distintas instancias de Sprint Retrospectives, utilizamos la herramienta Metro Retro, que brinda una sencilla y eficaz forma de realizar esta ceremonia. La posibilidad de poder escribir cada uno lo que piensa, para luego defender su postura y compartirla con los compañeros nos brindó instancias de puesta en común y ver cosas que quizás no todos las habíamos notado.

Metro Retro también permite establecer un marco de tiempo para la ceremonia, lo que nos mantiene enfocados en la tarea en cuestión y no permite distracciones con el fin de realizar correctamente la ceremonia.

Por el contrario, una de las features de esta herramienta que no utilizamos fue el sistema de votación. Sinceramente no recordamos que se encontraba esta funcionalidad, que permite que el equipo distinga cuáles son las “notas” que más importaban. De igual manera, mientras íbamos discutiendo las distintas notas, cada uno daba su opinión y quedaba claro la importancia de cada una.

Sprint Review Meeting

Las review meetings fueron las instancias donde aplicamos en su mayor expresión la diversidad de roles de la metodología ágil. Dividimos el equipo en PO y los desarrolladores, que mostraban los avances de cada iteración. Quizás en estas instancias no se preparó como correspondía dicha ceremonia, ya que por ejemplo, en una de las iteraciones, notamos que no había uniformidad en el diseño de la aplicación, desde botones de distinto color, hasta las fuentes del texto del prototipo. Cabe aclarar que esto sucedió principalmente en la segunda iteración, donde se prototipó por primera vez. Luego de recibir el feedback por parte de los docentes fue que notamos estas inconsistencias y se tomó acción inmediatamente para solucionarlas para la siguiente iteración.

Para hacer las review meeting también decidimos que haya una persona que es la que maneja el prototipo para mantener el orden

Algo que nos gusto mucho, es que a medida que cada US se presentaba, las persona que había trabajado en ella hablaba un poco de los requerimientos, como hizo para lograrlos, etc. Esto hizo que todos los desarrolladores siempre tuvieran una parte en la meeting, y además brindarán buenos comentarios ya que cada uno sabía bien en que había trabajado y por qué de esa forma.

Tablero

En el curso vimos dos tipos de tablero: tablero sustentable y tablero ágil. Al principio comenzamos utilizando el tablero ágil, pero ya para la segunda iteración comenzamos a usar el tablero sustentable siguiendo BDD. Con el tablero sustentable se observa más claramente en qué etapa del desarrollo se encuentra dicha story. Además, el tablero es fácil y entendible ya que las columnas describen claramente lo que se está realizando con la tarjeta. Sin embargo, el hecho de que hubieran tantas tarjetas significó un mayor número de movimientos, lo que hizo que hubiera algunos errores de comunicación. De todos modos, nos parece la mejor opción para ser uno de nuestros primeros proyectos ágiles.

Si bien seguimos la metodología de BDD, hay muchas cosas en el ciclo de BDD que no se aplican en nuestro proyecto. Por lo tanto, tuvimos que hacer varios cambios en el tablero sustentable para que se ajuste al contexto de nuestro proyecto. El ejemplo más claro es que nosotros solo prototipamos, por lo que todo lo que las partes de implementación en código y testing automatizado se sacaron o cambiaron de nombre a “prototipado”

Prototipado

Al comienzo del proyecto, la mayoría del equipo nunca había realizado un prototipado de una app. Nos generó mucha incertidumbre y creíamos que iba a ser una tarea muy compleja, más aún al leer todas las plataformas de prototipado que existen. Temíamos que al elegir la herramienta equivocada podría generarnos problemas a corto y mediano plazo.

Afortunadamente los docentes introdujeron Framer, la herramienta de prototipado que se utilizó para este proyecto. Sin duda que la herramienta fue vital para que el prototipado haya sido un proceso entretenido y eficiente. El aventurarse en las distintas funcionalidades fomentó la creatividad en las funcionalidades y las distintas ideas del equipo. Al ser tan intuitiva, el prototipado se logró realizar de buena manera, siempre con cosas a mejorar, pero aquello a lo que tanto le temíamos al comienzo resultó ser de las actividades más entretenidas del proyecto.

El prototipado en sí, como se mencionó anteriormente, provocó que en la segunda iteración se sobreestime el tiempo y trabajo que realmente iba a llevar. Esto, en nuestra opinión, sucedió debido a que tenemos asociada la imagen de prototipado con desarrollo, por lo que las funcionalidades a implementar parecían muy demandantes. Con el aprendizaje durante el curso y las demostraciones de los distintos prototipos, notamos que es un gran procedimiento para aprender y practicar. Además de eso, el equipo se adaptó muy rápido a la herramienta y, al poder trabajar en línea, cada uno podía ver el progreso de los demás compañeros e incluso analizar temas de diseño.

Proyecto en general

Una instancia importante que tuvimos fue la de validación del producto con usuarios. Por un lado, estas estuvieron muy buenas ya que nos dieron información importante del producto y de mejoras que podríamos tener en cuenta. Nos mostró lo importante que es tener a un usuario “real” que pueda utilizar el producto, ya que hay muchas cosas que nosotros los desarrolladores pasamos por alto. Lo que deberíamos mejorar es tener estas instancias un poco antes en la etapa de desarrollo. Si bien la información fue valiosa, como tuvimos estas instancias al final de la iteración, hay cosas que nos dijeron que no pudimos mejorarlas.

Conclusiones

SCRUM

En el proyecto pudimos aplicar la metodología ágil de SCRUM (con algunas modificaciones debido a los horarios que manejaba el equipo) y aprendimos mucho de ella.

Aprendimos a funcionar como un equipo auto-gestionado, decidiendo internamente quien hace que, lo que nos dio más sentimiento de propiedad (*ownership*) sobre el proyecto; y por otro

lado, trabajar en un equipo “cross-functional” nos enseñó a confiar en nuestros compañeros y poder ayudarnos unos a otros.

Basándonos en el empirismo y *lean thinking*, nos aseguramos tanto de que nuestro conocimiento y decisiones surjan de lo observado, y que no haya desperdicio al concentrarse en lo esencial. Todo esto, y siguiendo un acercamiento iterativo incremental pudimos maximizar la predictibilidad y control de riesgo. Los problemas que surgían en un momento dado del proyecto, podíamos encararlos rápidamente en la siguiente iteración.

Concentrarnos en un Sprint nos mostró directamente los beneficios de un proceso iterativo y corto. Tener un Sprint Goal siempre en la mira nos ayudó a concentrarnos en lo esencial para aumentar el valor del producto. Tener siempre una *Definition of Done* hizo que estuviéramos seguros siempre de cuándo una historia se consideraba terminada y evitar desacuerdos.

Para aprender cómo funciona SCRUM fue esencial aplicar bien las ceremonias más importantes: Sprint Planning Meeting, Sprint Retrospective Meeting y Sprint Review Meeting.

En la planning meeting aprendimos a estimar, discutiendo entre los desarrolladores por qué una historia podría costar más o menos, y llegar a una conclusión al final. Al final de la meeting, todos siempre tuvimos claro que es lo que se debía hacer en el Sprint, cual era el objetivo, y cuanto nos llevaría hacer el trabajo.

La Sprint Review Meeting, si bien fue una de las menos “reales” (pues no hubo PO real ni cliente real), de todos modos nos enseñó a presentar nuestro trabajo a un cliente para ver el progreso de nuestro producto. Discutimos de que fue logrado, que había que mejorar, y eso lo llevábamos al siguiente Sprint.

La retrospectiva fue de las más importantes, pues nos dio la oportunidad de debatir con nuestros compañeros desarrolladores que salió bien, qué problemas hubieron, etc.

Algo importante a destacar también de las ceremonias es que conocimos y pudimos utilizar múltiples herramientas que nos ayudaron a cumplir los objetivos de las ceremonias (planning poker en azure, metroretro). Quizás es menor, pero siempre es útil contar con herramientas que nos ayuden a organizarnos, y conocerlas puede hacernos destacar.

Todo esto nos hizo entender muy bien, por lo menos a nivel básico, el marco de gestión ágil. Todos fuimos PO en algún momento, todos fuimos desarrolladores, y en conjunto todos fuimos Scrum Master. Sentimos que entendemos mejor qué significan estos roles y cómo su trabajo afecta al progreso del equipo en el proyecto.

BDD

Seguir la metodología de BDD nos dio una manera eficiente de seguir el ciclo de desarrollo de una US. Siguiendo los pasos (que fueron modificados para ajustarse a nuestro contexto) siempre fue claro que se debía hacer, tanto para nosotros como para el resto del equipo que podía ver el estado de la tarjeta en el tablero.

Esto nos aseguró por ejemplo, no comenzar a prototipar y que luego al terminar nos diéramos cuenta que las cosas estaban mal porque no habíamos relevado los requisitos, gracias a que ese siempre fue el primer paso. De la misma manera, nunca nos faltó testear el prototipo, que en nuestro caso también incluía testing cruzado entre varios desarrolladores, logrando siempre una validación segura del trabajo.

Una de las cosas más importantes que aprendimos de BDD, fue cómo escribir una buena historia. Con títulos, escenarios, narrativas, etc. Luego de aplicar un poco de esto en el proyecto, entendimos por qué es tan importante escribir buenas historias. Todos entendemos lo que está ocurriendo, especialmente el cliente, para evitar situaciones como “!Esto no es lo que yo pedía!”

BDD Nos asegura que nuestra idea pueda transformarse en “código” implementado, testeado y pronto para ser puesto en producción.

Prototipado

El prototipado fue lo que sustituyó al código en nuestro proyecto, pero eso no afectó de manera negativa nuestro aprendizaje básico de los marcos de gestión ágil (aunque tuvimos que ajustar BDD, lo importante lo aprendimos).

En cierto sentido, prototipar nos dio un acercamiento a un área del marco de gestión ágil que pocos o ninguno conocíamos. Leyendo sobre *Dual Track Agile* aprendimos cómo una forma eficaz de asegurarse de que lo que entra a desarrollo es lo que el cliente quiere, es con prototipos. De esta forma, se adelanta el momento en el que el cliente ve el trabajo realizado: ya no es al final del desarrollo, en cuyo caso si estuviera mal, eso sería trabajo perdido. En cambio, en Dual Track Agile ese momento se hace en un ciclo separado al de desarrollo, y funciona como requerimiento para que un ítem se agregue al backlog.

Por lo tanto, prototipar nos mostró el poder que tiene para agilizar el proyecto un simple prototipo. Utilizando solamente los prototipos, pudimos mostrarle al cliente como nuestra aplicación cumpliría los requerimientos una vez implementados. Al final de la iteración 2, una nota importante fue la de mantener la uniformidad de la interfaz de usuario. Si en vez de estar prototipando estuviéramos codificando, sería muy desmotivante al final de la iteración darse cuenta de que la mitad de lo que hicimos está mal y hay que diseñarlo de vuelta. Luego de esa situación, rápidamente cambiamos el diseño para que sea más uniforme. Situaciones como

esta (no así de simples, claramente), son las que entendemos que ocurren en proyectos reales con clientes reales.

Trabajo como equipo

A la hora de organizar y realizar la documentación, no todos los integrantes tomaron el mismo nivel de iniciativa, por lo que no todos los integrantes aportaron lo mismo. Sin embargo, a la hora de prototipar y de realizar el trabajo asignado, consideramos que se realizó un trabajo satisfactorio ya que todos los integrantes cumplieron en tiempo y forma con lo que se le había asignado.

Comunicación y organización

En términos de comunicación consideramos que se mejoró en el correr del proyecto y se llegó a un nivel satisfactorio. Pasamos de trabajar 2 personas en la misma US por falta de comunicación a dividirnos las tareas a hacer en función de los horarios de cada integrante, ya que en muchos casos no coincidían. También la mejora en la coordinación se vio reflejada en la uniformidad del diseño de la aplicación, ya que en la primera iteración de prototipado cada integrante desarrolló la interfaz de usuario basándose simplemente en sus gustos, por lo que naturalmente no había consistencia alguna en el uso de colores por ejemplo. Esto se observó en la sprint review y fue corregido para la siguiente iteración, ya que se discutió y se llegó a un consenso en el uso de colores y de íconos.

Lecciones aprendidas

IF....THEN → Anecdota

Si creen que les va a dar el tiempo los últimos tres días ENTONCES están equivocados.

Esta lección la aprendimos en la segunda iteración, debido a que por distintas entregas de otras materias y horarios laborales, el equipo estaba sobrecargado de horarios y terminamos postergando el progreso para los últimos días. De no haber hecho la Sprint Planning Meeting en la fecha correspondiente, no hubiésemos logrado realizar la iteración en el tiempo estimado. Es muy importante progresar, al menos un poco, desde los primeros días de las iteraciones.

Si creen que tienen todo lo necesario para dar por finalizada una ceremonia ENTONCES deberían revisar las distintas planillas con los entregables de cada una de ellas.

Muchas veces creíamos que ya habíamos discutido/hecho todo lo necesario para cada ceremonia, y nos encontrábamos con que algún entregable nos faltaba. Principalmente nos pasó al principio, ya que cuando recibimos el feedback de la primera entrega notamos que estos archivos/planillas no los habíamos tenido en cuenta. Agradecer a los profesores por tener dicho material para que podamos guiarnos y realizar la iteración de la mejor manera posible.

Si van a estimar las tareas de prototipado ENTONCES deberían considerar que la complejidad no se acerca a la que llevaría desarrollar dicha feature.

En nuestro caso, como se mencionó en las conclusiones y reflexiones, sobre-estimamos el tiempo que nos llevaría prototipar debido a la falta de experiencia en este campo. Para nuestra sorpresa, el prototipado con Framer es bastante intuitivo y fácil. La complejidad de cada feature es lo que puede hacer más o menos demandante el prototipo, como también la creatividad del equipo.

Si van a tener instancias de validación con usuarios ENTONCES asegúrense de tenerlas, a lo sumo, a mediados de la etapa de desarrollo

Para la tercera iteración nosotros tuvimos estas instancias con usuarios. Fueron múltiples y nos dieron información importante. Sin embargo, como las tuvimos ya finalizando la etapa de desarrollo, hay cosas que no podríamos haber cambiado si las hubiéramos hecho antes.

Si van a utilizar metodologías ágiles ENTONCES siempre recuerden de ajustarlas a su contexto

No todas las cosas se pueden aplicar a todos los proyectos. Las metodologías son guías. En nuestro caso, hacer Dailys de SCRUM era imposible por ser estudiantes. Y también al no haber código, hacer testing automatizado en BDD no tiene sentido. Siempre hay que considerar el contexto del proyecto y ajustarse a ello.

Si van a utilizar tableros ENTONCES definan en equipo cual se va a usar

Nosotros tuvimos un par de situaciones donde nos confundimos y en vez de usar el tablero sustentable usamos el tablero ágil. En una ocasión un compañero estaba trabajando y moviendo las tarjetas en el tablero ágil, mientras que el resto estábamos en el tablero sustentable.

Si van a utilizar herramientas para la retrospectiva ENTONCES usen metroretro

Es una herramienta muy fácil de usar y simple para entender. Brina una forma fácil de llevar a cabo esta ceremonia, concentrándose en qué cosas hay para mejorar, qué problemas surgieron, etc.

Video

Problema a resolver

Proveer una interfaz amigable para facilitar que los usuarios puedan tener acceso a la información acerca de las líneas de ómnibus en Uruguay y brindar herramientas útiles para el traslado en las mismas.

Valor entregado por el producto

Mediante esta app un usuario puede acceder a un amplio conjunto de herramientas que le serán de gran utilidad a la hora de usar el transporte público uruguayo. Entre ellas destacamos:

- Hacer seguimiento de las líneas (actualizaciones en tiempo real)
- Ver información de las líneas (congestión de la ruta, fotos de la parada, estimaciones de tiempo de demora y llegada, cantidad de personas, etc.)
- Recibir notificaciones varias relevantes para los usuarios (retrasos, llegadas, etc.)
- Compartir viajes control usuarios (por diferentes medios -whatsapp, messenger)
- Ver historial de viajes
- Tener la posibilidad de guardar locaciones y líneas en favoritos para tener fácil acceso a ellas

[Link al video](#)