

Reporte Académico Final

Ingeniería de Software Agil I

2021

Facultad de Ingeniería
Universidad ORT Uruguay

Profesor:
Álvaro Ortas

Iñaki Etchegaray - 241072

Matías González - 219329

Gastón Landeira - 238473

Link al Repositorio de Github:

<https://github.com/GLandeira/mp-M6A-05>

Índice

Introducción	2
Reflexión sobre el Proceso	2
Drop	2
Add	3
Keep	3
Improve	4
Inspección de Resultado y Objetivos Iniciales	5
Lecciones Aprendidas	6

Introducción

Se presenta un reporte académico con las conclusiones por parte de los integrantes sobre el proyecto. Estas incluyen tanto reflexiones sobre las iteraciones pasadas como un análisis sobre el producto final entregado. Se finalizará con un listado de las lecciones aprendidas que llevaremos con nosotros para un siguiente proyecto.

Reflexión sobre el Proceso

Para la realización de este informe, decidimos cada uno escribir sus opiniones y reflexiones sobre el proyecto, luego las compusimos sobre un tablero DAKI. Luego discutimos cada una de las tarjetas agregadas por los distintos miembros del grupo y llegamos a las conclusiones que presentaremos a continuación.

La razón por la cual decidimos utilizar esta metodología es porque ya la veníamos utilizando en las retrospectivas anteriores, y al habernos resultado decidimos utilizarla aca tambien.

Tablero DAKI realizado:

The screenshot shows a Kanban board titled "DAKI Final" with a background image of an iceberg. The board is divided into four columns: "Drop", "Add", "Keep", and "Improve". Each column contains several cards with text reflecting on the project process. The "Drop" column has one card about reviews. The "Add" column has four cards about planning, release planning, sprint quantity, and planning poker. The "Keep" column has four cards about tools like Toggl, sprint planning, Lean coffee, and Trello. The "Improve" column has four cards about requirements, estimation techniques, DAKI practice, Definition of Done, sprint planning time, and daily scrum.

Drop	Add	Keep	Improve
Las reviews dentro del proceso de ingeniería. Mover a sprint review o hacerlo de manera informal entre medio. No ceremonizarlo.	Antes de empezar las iteraciones, haria un planning de proyecto (Estudio de competidores, definicion de solucion, Product backlog)	Toggl, o alguna herramienta de temporizacion, para todo el proceso	Uso de los requerimientos no funcionales
	Luego del "project planning", haria un release planning con todos sus artefactos: date/release driven, de ahi seleccionar del product backlog, hacer las estimaciones, etc	El proceso del sprint planning, la manera en la que se definieron los objetivos, el proceso de eleccion del backlog, etc.	Tecnicas de estimacion desde el principio (Planning Poker, Affinity Estimation, etc)
	Decidiria la cantidad de sprints y su duracion en base al release plan	Adaptacion de las tecnica de Lean coffee	Hacer DAKI y tecnicas de retrospectiva desde el principio.
	Un proyecto como este, que tiene pocas historias de usuario y poca gente, usariamos Affinity Estimation sobre Planning Poker	Trello Como herramienta de organizacion de tareas	Realizacion y uso de Definition of Done
	Experimentar con Burndown chart	Balsamiq para la produccion de prototipos	Tiempo dedicado a sprint planning (nos estaba tomando el doble del "ideal" mencionado en la SCRUM guide.)
		Realizaciones de estudio de competidores y identificacion de stakeholders para asistir la construccion de las historias de usuario.	Definir el daily scrum mejor desde el principio (como regular scrum o otra cosa) tomando en cuenta la naturaleza del proyecto. Hacerlo algo mas hablado.

Drop

Consideramos innecesaria la etapa de Review dentro del proceso de ingeniería. El equipo estaba realizando una etapa de review al final de cada user story donde el developer le mostraba el trabajo realizado para la misma al product owner y este recibe un feedback al respecto. Esta etapa le sacaba significado al Sprint Review final ya que él mismo terminaba

siendo simplemente las review de cada user story conjunta. Además el hacer una review individual para cada historia de usuario implicaba una mayor pérdida de tiempo y la necesidad de contactar muchas veces con el product owner, algo que no siempre es posible. Por lo tanto si tuviésemos que hacer de nuevo este proyecto la review en el proceso de ingeniería se eliminaría por completo o se redefiniría a ser una tarea individual para que otro developer revise el trabajo.

Add

Creemos que sería vital sumar dos etapas de planning previas a comenzar con cualquier proyecto. Primero un Project Planning en el cual se haga el estudio de competidores y la definición de la solución, que abarcaría un determinado número de iteraciones, repitiendo después de haberlas finalizado. Y un Product Backlog que abarcaría la totalidad del proyecto.

Luego, un Release Planning por cada conjunto de funcionalidades que se realice del proyecto. En estas instancias se definirá si la misma será date/release driven, hacer la estimación de las user stories planeadas para la misma, la cantidad de iteraciones, su duración y la cantidad de user stories que formarán las mismas.

Para la estimación de story points nosotros utilizamos Planning poker. Si bien la misma nos resultó útil, nos parecería interesante experimentar con el uso de Affinity Estimation. Esto se debe a que si bien esta escala pobremente, en un grupo con pocos integrantes y una baja cantidad de user stories como el nuestro podría probar ser muy eficiente.

Finalmente, creemos importante el comenzar a utilizar burndown charts durante las iteraciones para llevar un registro del progreso del equipo. Esta técnica no se utilizó en un comienzo dado el desconocimiento del equipo sobre ella y su utilidad pero teniendo ahora más práctica con la metodología creemos esta herramienta sería de gran utilidad.

Keep

Durante las distintas iteraciones nos dimos cuenta que las herramientas que dispusimos para trabajar fueron el mejor aspecto del proyecto. Agradecemos haber elegido utilizar Toggl como herramienta de temporización y sin duda la utilizaremos para todos los proyectos por venir que requieran de seguimientos parecidos.

Similarmente, Balsamiq para el prototipado y Trello como herramienta de realización de taskboards, son lo suficientemente profesionales como para considerarlas para proyectos futuros.

A la hora de realizar las retrospectivas, nosotros utilizamos Lean Coffee. Lean Coffee es una muy buena técnica de retrospectiva para desarrolladores sin experiencia como nosotros. Sin embargo, muchas de sus características parecían adaptarse a equipos más grandes, con gente que no necesariamente posee tanta confianza entre ellos (como en muchos ambientes de trabajo) y en ambientes físicos. Dada las circunstancias en las que nos encontrábamos, muchas de las características no cumplían con nuestro grupo:

- Somos un grupo compuesto de amigos de la vida que tienen confianza entre ellos.

- Somos un grupo chico de tres personas.
- Estábamos trabajando en remoto.

De esta manera, decidimos adaptar Lean Coffee a una experiencia más cercana, en donde opinamos sobre el proceso en un ambiente de confianza. Por lo tanto las retrospectivas consistieron en turnarnos y trabajar sobre un tablero con listas DAKI y algunas de las ceremonias determinadas por Lean Coffee. Ceremonias que se enfocan más a que los integrantes se suelten no las tomamos en cuenta por lo mencionado anteriormente. Tampoco ceremonias que necesitan interacción física entre integrantes al estar en un ambiente remoto. Si tomamos ceremonias que buscan el pensamiento individual e imparcialidad para lograr un mejor espectro de ideas.

Creemos que esto resultó en una adaptación de Lean Coffee que era favorable para nuestra situación y que dio buenos resultados. Por lo tanto, si nos encontráramos en un proyecto con características similares, lo aplicaríamos.

En la primera iteración, nosotros realizamos ciertos documentos de identificación de stakeholders y un estudio de competidores. Estos documentos fueron sumamente útiles a la hora de realizar las historias de usuario y definir el product backlog ya que nos responden tres preguntas claves: **que**, **como** y **quien**. El estudio de competidores junto a la especificación del producto nos ayudó a identificar **qué** funcionalidades son un estándar para este tipo de aplicaciones y **como** un usuario tipo espera que estas funcionalidades se dispongan. Luego la identificación de stakeholders nos ayuda a responder la cuestión de **quién** usaría nuestra aplicación y porque, que tipo de funcionalidades debemos de priorizar y fue sumamente útil para asistir con él 'Como' de las historias de usuario. En un proyecto futuro, sin duda realizaremos estos documentos nuevamente.

Improve

Entre las cosas señaladas a mejorar, nos encontramos con el uso y el manejo de las historias de usuario que definen requerimientos no funcionales. Las mismas fueron realizadas y tomadas en cuenta, pero de una manera mental. Es decir, al realizar los prototipos y las historias de usuario, siempre se tomó en cuenta los requerimientos no funcionales, pero creemos que su manejo no fue ideal. La historia de usuario se dejó en el product backlog y, al no traerla a los sprint backlogs para tenerla en cuenta, al final del proceso el product backlog quedaba con los requerimientos no funcionales, lo cual es raro y no sentimos que encaje.

Para la próxima vez, incluiremos los elementos no funcionales del product backlog en los sprints en forma de una lista de requerimientos no funcionales, o siendo parte del definition of done. Como ejemplo, en un pizarrón o cualquier lugar fácil de acceder por todos los miembros del equipo.

Muchas de las cosas que identificamos para mejorar trataban sobre el proceso en el tiempo. Nosotros aprendimos a usar scrum en el camino a realizar este proyecto, por lo que hubieron muchas cosas que nos hubiera gustado haber hecho desde el principio:

- La realización de planificaciones sobre los elementos del product backlog desde un principio.

- El uso de técnicas de estimación de esfuerzo de tareas o tamaño de historias de usuario como Planning Poker o Affinity Estimation desde el principio del proyecto.
- El uso de técnicas de retrospectiva como DAKI a la hora de hacer retrospectivas.

Otra de las cosas que identificamos para mejorar del proceso es el tiempo dedicado a la Sprint Planning. Parte de esto puede ser por la falta de experiencia de las metodologías, pero en varias ocasiones, nos encontrábamos doblando la duración recomendada por la Scrum Guide. Para un proyecto futuro, haría bien dejar en claro qué elementos deben surgir de la Sprint Planning y realizarla de una manera más guiada y grupal con metodologías investigadas de planificación.

Al principio de la iteración, nosotros realizamos Daily Scrums, pero rápidamente nos dimos cuenta de que esto no encajaba en la naturaleza del proyecto por una razón muy simple: nosotros no trabajamos todos los días. Por lo que decidimos cambiar el nombre a Regular Scrum. Pero estos problemas siguieron surgiendo, ya que al ser regular no era una buena manera de describir que hizo uno 'ayer', sino la última vez. Además, estas reuniones las realizamos de una manera más escrita y no tan verbal. Para la próxima vez, procuraremos definir mejor los días de trabajo y en base a ello definir el esquema de Daily Scrum. Además, debemos hacer de esta ceremonia algo más interactivo, verbal y grupal en vez de simplemente escribirlo en una hoja como hacíamos.

Inspección de Resultado y Objetivos Iniciales

Concluimos las tres iteraciones con un prototipado en balsamiq de todas las historias de usuario que se pudieron realizar, conectadas entre sí con su caso de uso común. Además, se agregaron los escenarios no principales a cada una de ellas en sus respectivos archivos de balsamiq también.

Nuestro grupo cree que este prototipo generado cumple con los objetivos instaurados. El mismo siendo el prototipo de un Minimum Viable Product que cumple con las funcionalidades instauradas en la especificación del proyecto. A pesar de que nos faltaron dos historias de usuario por completar, las mismas fueron identificadas como de mínima prioridad, por lo que las funcionalidades principales fueron todas representadas en la prototipación.

Las funcionalidades que creemos se cumplieron satisfactoriamente:

- El prototipo brinda métricas del estado de la enfermedad relevantes para la población como casos activos, cantidad de vacunados, tests realizados y camas de CTI, el primero con la opción de dividirlos por departamentos, y el último dependiendo la distancia a la ubicación actual.
- El prototipo brinda un formulario al usuario con preguntas referentes a los síntomas de diagnóstico de Covid19. Una vez completado este indica probabilidad de contagio y la posibilidad de agendar un hisopado.
- El prototipo modela alertas de exposición al virus y agrega el valor de aquellas.
- El prototipo se compone de una interfaz apta para dispositivos móviles, cumpliendo el requisito no funcional determinado en los objetivos.
- Otorgamos customización de privacidad y manejo de la información de los usuarios, por lo que la privacidad es un atributo de alta prioridad de la aplicación.

- El prototipo integra funcionalidades de la competencia como el agendado de hisopados, el otorgamiento del registro de vacunaciones y la implementación de un botón de emergencia para gente mayor.
- El prototipo logra cumplir con el objetivo de otorgar una experiencia fácil y placentera para todo tipos de usuarios al proveer una guía de uso de la aplicación en caso de que algo no se entienda por el usuario, además se agregó una funcionalidad de multi usuario, así los miembros menos hábiles de una familia pueden ser asistidos por aquellos que sí son hábiles en su uso.

No se cumplieron directamente:

- Registro de las dosis restantes de la población en los diferentes vacunatorios con sus horarios disponibles.

Si bien en la pestaña de información en Vacunaciones se ven la cantidad de dosis restantes para el día actual, esta no se dividió por diferentes vacunatorios ni se mencionó sus horarios disponibles de ellos. Esto sucedió por un desvinculamiento entre las historias de usuario y el objetivo del proyecto. Se enfocó tanto el grupo en seguir las ceremonias de Scrum que esto pasó por alto aunque hubiese dado tiempo de sobra para implementarla.

A continuación dejamos el link al video en el cual se presenta el producto: [linkadink]

Lecciones Aprendidas

A modo de conclusión, decidimos realizar una lista de lecciones aprendidas que resume el documento de manera rapida y facil de entender:

1. Es importante disponer de buenas herramientas para trackear el tiempo dedicado a cada tarea y el progreso de las mismas.
2. Una buena herramienta de creación de tableros, de su manejo y que sea fácil de usar e interactiva entre muchos usuarios puede asistir mucho a la hora de realizar taskboards para el seguimiento del proyecto.
3. Aplicar todos los elementos de Scrum desde el principio, al tener ahora el conocimiento completo de la metodología.
4. Agregar ceremonias de planificación de proyecto y de planificación de Releases.
5. En los mismos, utilizan ampliamente técnicas para la estimación del tamaño de las historias de usuario y del esfuerzo de las tareas.
6. De manera similar, adoptar desde el principio del proceso las técnicas de retrospectivas aprendidas en el curso, ya que estas favorecen mucho la interacción entre los integrantes del equipo, resultando en mejores retrospectivas y mejores aprendizajes.
7. Es importante realizar una identificación de stakeholders y un estudio de competidores para luego realizar historias de usuario de una manera más fácil y más acertada a la realidad.
8. Etapas definidas a la hora de trabajar en cada user story permite un mejor control sobre las tareas y el avance de cada integrante.

9. Las ceremonias de scrum son adaptables a cada equipo de trabajo. Es importante empezar con las bases de scrum y después ir las adaptando y mejorando constantemente.
10. No importa la cantidad de planning, siempre surgirán situaciones inesperadas. El grupo muchas veces planeaba de más dedicando demasiado tiempo a ello. En conclusión la lección sería, no planear de más. Las metodologías ágiles por naturaleza buscan tomar las decisiones lo más tarde posible.
11. No es malo intentar adaptar las técnicas de Scrum y que no resulte como se esperaba (Experimentar). Siempre se puede volver a lo que resultó más eficiente.