**Feedback primera entrega 2023: Joaquin Sommer – 184441, Sofia Begino – 230763, Romina Giaccio - 206127. 15/04/23.**

1. Sobre el proceso de gestión.
2. En realidad, las ceremonias de Kanban son dos: Daily Standup (como su nombre lo dice, es diario) y retrospective (una vez por semana y no una vez por entrega). Deberían rever esto.
3. Recuerden que el proceso debe indicar en cada paso cuál era su resultado (¿Qué obtengo), el rol (¿Quién lo hace?), el momento en que se dispara la actividad (¿Cuándo se hace?) y, finalmente, que métodos y herramientas se usan para construir el resultado (¿Cómo se hace?).
4. El tablero es un reflejo visual de lo que es el proceso. Por tanto, no deben mezclar tablero con proceso.
5. En el caso particular del “tablero.png” no se corresponde con lo que definieron como proceso de gestión. Las columnas que tiene ese tablero no son pasos del proceso de gestión.
6. Además, el tablero debería estar en “Projects” de Github.
7. Sobre la Pequeña guía de "Definición/uso del proceso de ingeniería en el contexto de KANBAN".
8. Uds. lo denominaron “ProcesoDeIngenieria”. Se agradece que mantengan el nombre “Pequeña guía de "Definición/uso del proceso de ingeniería en el contexto de KANBAN", ya que nos facilita encontrar los ítems en el repo. Así hagan con todos los ítems del obligatorio.
9. Para la primera entrega, solo se esperaba lo que Uds. denominan “Tareas sin user stories”. Ya que solo tenían que encontrar bugs (testear) y encontrar problemas de codificación (analizar código). Lo que denominan “Tareas con user stories” es recién para la tercera entrega. Sin embargo, para la segunda entrega, si tiene que cambiar el proceso de ingeniería, con algo mas complejo que el de la primera entrega y menos que el de la tercera. Son 3 versiones de proceso de ingeniería.
10. Lo que Uds. llaman “Tareas sin user stories” se podría haber separado el “In Progress” en dos pasos: Testing y Análisis de Código.
11. También, se pretendía que indicaran en cada paso cuál era su resultado (¿Qué obtengo), el rol (¿Quién lo hace?), el momento en que se dispara la actividad (¿Cuándo se hace?) y, finalmente, que métodos y herramientas se usan para construir el resultado (¿Cómo se hace?).
12. El doc. que armaron como “Roles del Equipo” va dentro del quien en el proceso de ingeniería (punto anterior). Tengan cuidado de agregar roles que después no los usan, dado que tienen que aparecer todos en los tableros e informes de esfuerzo.
13. La idea es que después puedan instanciar/convertirlo en un tablero.
14. “tablero.png” tampoco coincide ni con “Tareas sin user stories” ni con “Tareas con user stories”.
15. Recuerden que el proceso es previo a la construcción del tablero y que el tablero es un reflejo visual de lo que es el proceso. Tómenlo como una lección aprendida para la próxima entrega.
16. Les sugiero que para la tercera entrega revean los artefactos porque en ningún caso aparece un “Proceso de gestión”, sino el “enmarcado” en un proceso de gestión.
17. Fue una excelente idea mostrar evidencias de ejecución del proceso en el tablero en el documento “StandUp.md”.
18. Sobre el documento “Templates”.
19. Me pareció una excelente solución. Aunque no se pedían todos ya en esta entrega, les va a servir para todas las entregas.
20. Lo que les recomendamos es que agreguen dos atributos: fecha de inicio y fecha de fin, para poder medir el lead time y cycle time.
21. Lo que les faltaría es la clasificación de bugs. Vean Anexo 1.
22. Pequeña guía de "Creación y posterior mantenimiento del repositorio: elementos que contiene y cómo los van a versionar".  
    Se espera un análisis más detallado. ¿Por qué toman esas decisiones? ¿Realizarán Pull Requests? Recordar que esto puede cambiar en el futuro.
23. Sobre el “Análisis de deuda técnica”.  
    Realizan un buen análisis de deuda técnica y se organizan para el mismo. ¿Usaron herramientas para el análisis estático de código? Siguen correctamente el formato y templates para la creación de las user Stories, tasks y bugs.
24. Sobre el “Detalle de registro de esfuerzo por tipo de tarea, según las actividades del proceso de ingeniería o del tablero, si se prefiere” y los “Totales de registro de esfuerzo por la entrega.”
25. Están muy bien.
26. De todas maneras, los totales de las tareas deberían de alguna forma relacionarlo con las fechas de inicio y fin.
27. Recuerden que estas no son iteraciones (no tienen un largo fijo), son entregas aleatorias completamente. No están usando Scrum, están usando Kanban.
28. A futuro deben reportar el esfuerzo de acuerdo al nombre de la tarea del proceso. Por ej.: Requirements Definition = x horas-personas, Application Implementation = y horas-personas, etc. Además de otras tareas que no las tengan en el proceso como puede ser la instalación de algo.
29. Sobre el “Informe de la retrospectiva.”
30. Los post-its están muy chiquitos y casi no se pueden leer. Por favor busquen una solución para que sean más legibles.
31. Algunas de las acciones planteadas aparecen como muy confusas y ambiguas. Por ej.: ¿Qué quiere decir “Dejar muchas tareas para los últimos días”? ¿Qué tipos de tareas? ¿Quién debería informa esto? ¿A quién?

Otro ejemplo: “Fechas fijas para reunirse.”: ¿Para reunirse para desarrollar, para planificar, para retrospectiva? Revisen que casi todos los post-its están ambiguos.

1. Algunas cosas las pudimos inferir leyendo las conclusiones, pero ese no es el objetivo de las conclusiones.
2. Por lo que alcanzamos a leer, vemos que plantearon solo problemas de gestión (“Dejar muchas tareas para los últimos días”), pero ninguna de desarrollo o ingeniería o técnicas. ¿Creen Uds. que están todos los temas importantes para que un equipo reflexione y mejore? Les menciono algunas que pueden (no estoy afirmando, esto lo deben determinar Uds.) haber omitido: métricas, estimaciones, riesgos, DoD, capacitación, cuellos de botella, entre otros. También pueden agregar otros.
3. Antes de llenar los cuatro cuadrantes les sugerimos que utilicen algún método para que no se olviden u omitan lo mencionado anteriormente y que además busquen una manera de asegurarse que el post-it que pongan es útil. Por ej.: preguntarse ¿por qué voy a poner este post-it? ¿Qué problema busco resolver? ¿El problema es de parte del equipo o de todo el equipo? ¿El problema se da siempre o a veces? ¿Es un problema realmente importante? Etc.
4. Las conclusiones están un poco pobres (serian los Actions Items).
5. De la DAKI no solo salen conclusiones sino lecciones aprendidas que las pueden ir anotando en un backlog para ir mejorando en sucesivas entregas y para hacerlas referencias en la entrega final. Pueden salir también dificultades y desafíos. Ídem en un backlog.
6. Les dejamos este link que es el método para retrospectivas que vimos el semestre pasado: <https://retromat.org/blog/best-retrospective-for-beginners/>. Si tienen dudas, lo vemos en clase.

Sugerencia: quizás podrían poner un pequeño Readme apenas se entra al repositorio para que el lector tenga una idea con que se va a encontrar. Con 5 o 6 líneas alcanza.