IIC 2143 Ingeniería de Software
Interrogación 1 - Semestre 1 /2020
Secciones 01 y 02

En caso de responder este enunciado en papel, responda cada pregunta en una hoja separada.
Recuerden que están bajo el código de honor.

Pregunta 1:

La empresa Initech se encuentra realizando un post-mortem de un proyecto de software fallido. Con el objetivo de entender qué salió mal, se le contrata a usted como consultor para analizar el proceso con que se llevó a cabo el proyecto. El gerente encargado de este comparte con usted el siguiente relato:

"En Initech nos enorgullecemos de seguir las mejores prácticas de desarrollo y de estar siempre a la vanguardia de los procesos modernos de manejo de proyectos, por lo que decidimos adoptar Scrum desde el primer día. Al inicio del proyecto, el cliente nos entregó un documento con los requisitos funcionales del proyecto y procedimos a plasmarlos en relatos de usuario. En base a ellos, construimos una carta Gantt para tener una aproximación inicial de cómo ir avanzando en el proyecto. Dividimos nuestro equipo de trabajo en 5 grupos, cada uno a cargo de un módulo distinto de la aplicación. Cada grupo consistía en 5 desarrolladores y 1 supervisor. Toda comunicación entre equipos debía hacerse a través de los supervisores para optimizar la toma de decisiones. Todos los días, realizamos un stand-up meeting entre yo y los supervisores para ir monitoreando avances. Estas reuniones se limitarían a resumir los sucesos de interés de la jornada anterior y no solían durar más de una hora. Siguiendo las buenas prácticas de Scrum, dividimos el proyecto en Sprints cortos de entre una y tres semanas de duración según la tarea planificada para el momento. Las tareas por realizar en cada Sprint se desprendieron de la carta Gantt inicial. Al finalizar cada Sprint, se le enviaba al cliente un reporte exhaustivo de todos los avances del proyecto en un documento oficial. La retroalimentación del cliente era analizada por gerencia quien estaría a cargo de destilarla en requisitos adicionales. De ser aprobados, estos requisitos serían derivados a los respectivos equipos de desarrollo para incorporarlos a la lista de tareas planificadas para el Sprint actual de manera a no generar atrasos en la planificación."

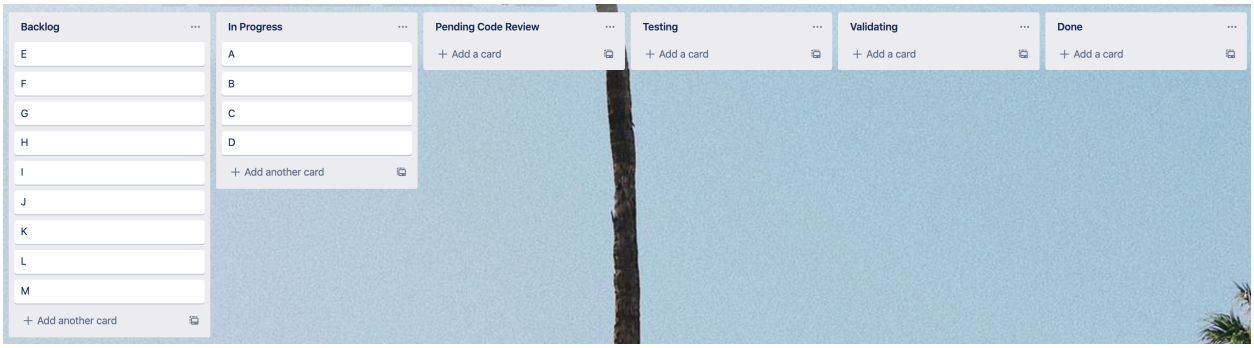
En base a sus conocimientos de ingeniería de software, ¿qué estima que se hizo mal?

Pregunta 2:

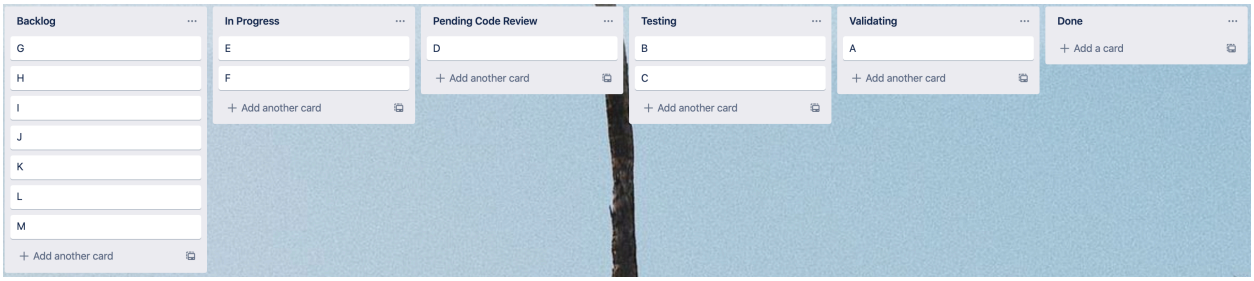
A continuación, se muestra el estado del tablero Kanban de un proyecto de software en distintos momentos de su ciclo de desarrollo. Cada tarjeta representa un relato de usuario distinto donde la letra representa su prioridad (la tarea A tiene mayor prioridad que la tarea B). Elabore una propuesta de cómo se hubiese podido llevar a cabo el mismo proyecto utilizando la metodología Scrum. Su propuesta debe incluir las actividades a realizar semana por semana desde el inicio del proyecto hasta su conclusión. Justifique su respuesta.

Asuma que los días a continuación denotan días hábiles desde que se comenzó a programar y que no hay feriados entre medio (i.e. el día 6 denotaría la segunda semana de desarrollo). Deje claramente establecido cualquier supuesto que estime necesario.

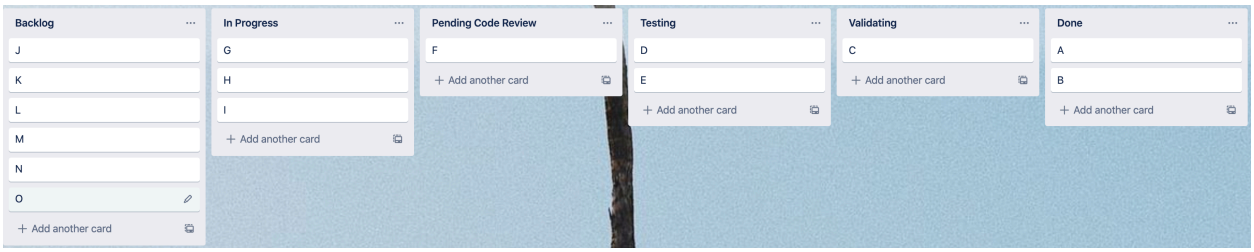
Día 1:



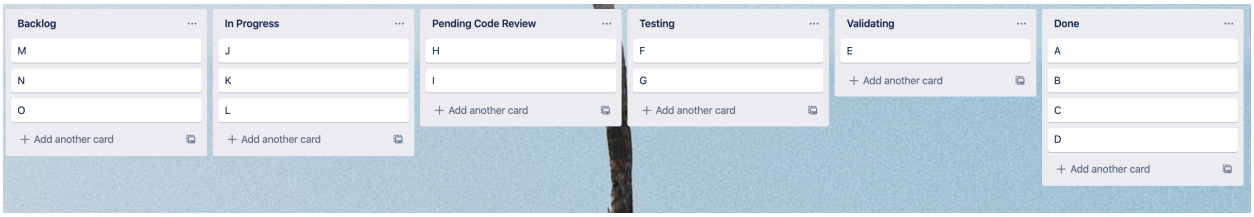
Día 5:



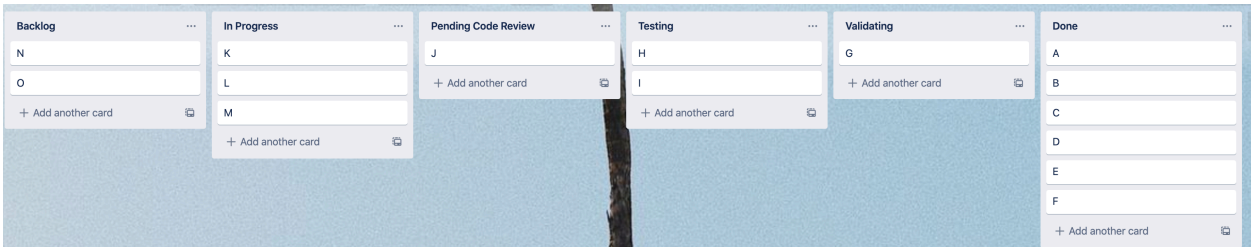
Día 8:



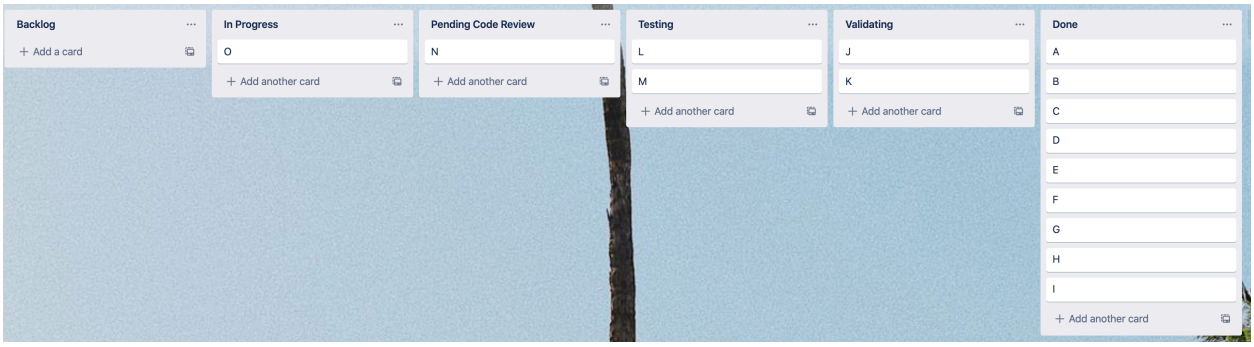
Día 12:



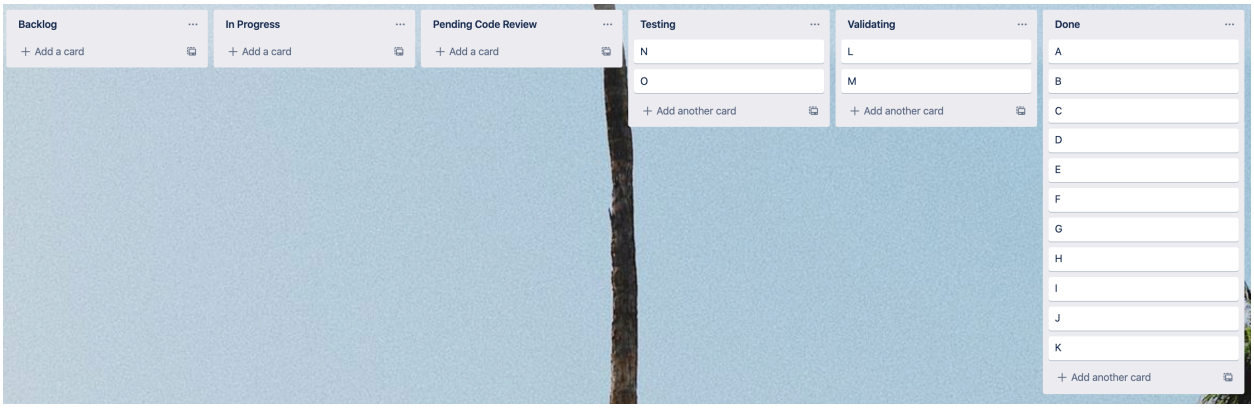
Día 17:



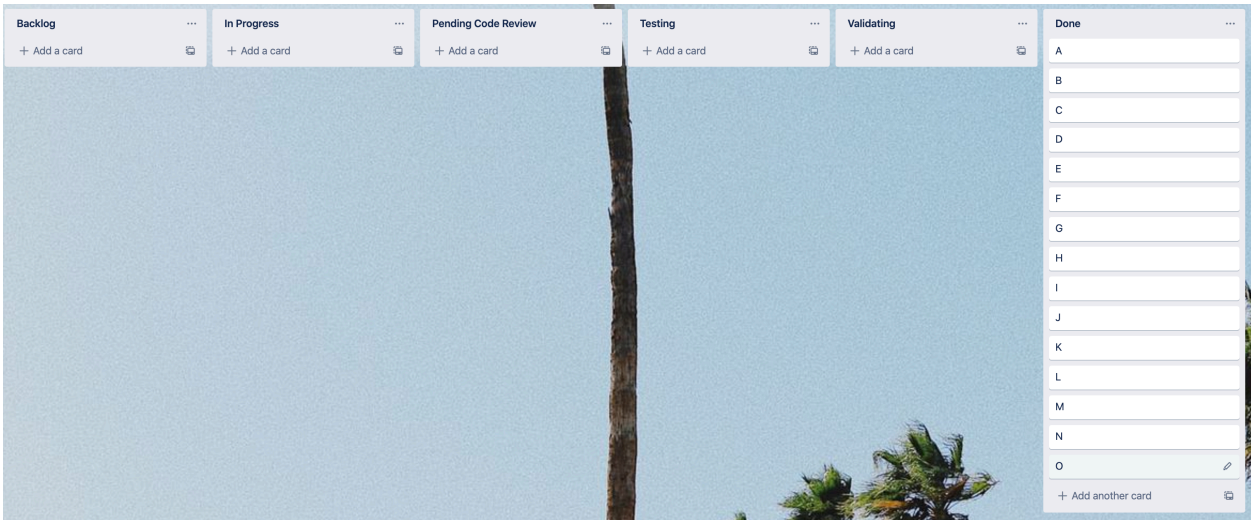
Día 22:



Día 25:



Día 28:


Backlog
Add a card
In Progress
Add a card
Pending Code Review
Add a card
Testing
Add a card
Validating
Add a card
Done
A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
Add another card

Pregunta 3:

A raíz de la pandemia de Covid-19 que afecta al mundo entero, un grupo de estudiantes de la UC quiere proponer una iniciativa para desarrollar una app que pueda ayudar a reducir la tasa de contagio. Aprovechando la alta tasa de penetración de smartphones del país se pretende que cada persona mantenga una aplicación que les ayude en diversos aspectos:

- Información sobre distribución de las personas que han sido confirmados como positivas.
- Información sobre locales de comida y farmacias mas cercanas.
- Información sobre restricciones vigentes (cordones sanitarios, cuarentenas).
- Información estadística sobre número de positivos, fallecidos y hospitalizados; en las últimas 24 horas, en la última semana, etc. (gráficos).

Se han levantado en total 26 relatos de usuario de los cuales 10 son simples, 6 son simples a intermedio, 8 son intermedio a complejo y 2 son complejos. No se definieron épicas ni temas. El desarrollo parte hoy 5 de Mayo, y el proyecto debe estar terminado a más tardar el Martes 2 de Junio (4 semanas). Se cuenta con un equipo de desarrollo que se estima puede completar en promedio 14 puntos de relato en una semana de trabajo (rango 12 a 18).

Realice una planificación inicial del primer release previsto para el 2 de Junio. Deje claramente establecido cualquier supuesto que estime necesario.

Pauta Pregunta 1:

Se pueden señalar los siguientes inconvenientes en el relato del gerente:

- Los relatos de usuario deben ser generados por el cliente, no deben ser generados por el equipo de desarrollo en base al análisis de un documento de requisitos.
- El forzar la comunicación a través un canal oficial particular (los supervisores) e impedir que los desarrolladores de distintos equipos se comunicaran directamente dificulta la coordinación entre equipos.
- Los stand-up meetings deben realizarse entre todos los miembros del equipo y no durar más de 15-20 minutos. Una hora es excesivo y sugiere que se estaban haciendo tareas administrativas adicionales que no corresponden.
- La duración de los Sprints en Scrum es fija, no debe ser variable.
- El basar el sprint backlog en una carta Gantt impide adaptarse a errores en la planificación, retrasos y cambios en el proyecto.
- No se aprecia una sesión formal de Sprint Review al final del Sprint. El enviar un documento escrito de avances no es un sustituto a una reunión presencial con demos.
- La retroalimentación debe ser analizada por los mismos desarrolladores y en procesos ágiles estos mismos deben estar empoderados para tomar las decisiones que estimen más convenientes. La estructura organizaciones vertical que se desprende del relato es ajena a las buenas prácticas promovidas por los procesos ágiles de desarrollo.
- El incorporar un requisito nuevo a un Sprint debe necesariamente estar acompañado de una replanificación para no agobiar al equipo de trabajo.

Nota de Pauta:

Cada una de las ideas expresadas en esta pauta debe figurar de alguna forma en la respuesta del alumno. Asignar 0.75 puntos por cada una.

Los ayudantes pueden descontar hasta 1 punto por aspectos formales (ortografía, gramática…)

Pauta Pregunta 2:

Una posible planificación de actividades considerando sprints de 2 semanas sería la siguiente:

Semana 1:

Presentación del proyecto
Elaboración de relatos de usuario
Product planning

Semana 2 - 3:

Primer sprint, tareas A a D.
Justificación: las tareas A a D se encuentran todas cerradas en el día 12. Dado que hasta ese momento se trabajó también en paralelo en las tareas E a L, es razonable asumir que de enfocar todos los esfuerzos solamente en estas tareas se pueden concretar en 10 días hábiles.

Semana 4 – 5:

Segundo sprint, tareas E a I.
Justificación: las tareas E y F se comenzaron a trabajar el día 5 y terminaron el día 17 (12 días hábiles de trabajo). Las tareas G a I se comenzaron el día 8 y terminaron el día 22 (14 días hábiles de trabajo). Nuevamente considerando todas las demás tareas que se hacían en paralelo, es razonable asumir que de enfocar todos los esfuerzos en estas tareas podemos cerrarlas en 10 días.

Semana 6 – 7:

Tercer sprint, tareas J a O.
Justificación: la tarea J se comenzó a trabajar el día 12 y terminó el día 25 (13 días hábiles de trabajo). El resto empezó a lo menos el día 17 y terminó a lo más el día 28 (11 días hábiles de trabajo). De focalizar los esfuerzos, se pudieron haber cerrado en 10 días.

Nota de Pauta:

Las justificaciones asumen que de focalizar esfuerzos en un conjunto reducido de tareas se podrían haber terminado más rápido. En caso de no considerar lo anterior, es perfectamente válido que la respuesta del alumno contemple sprints de 3 semanas de duración (ninguna tarea excede los 15 días hábiles de trabajo).

Cada sprint incluye actividades de sprint planning, sprint review y retrospective. Toda tarea abarcada en un sprint incluye desde su codificación hasta su validación.

Nota de Pauta:

La propuesta anterior es solo un ejemplo, existe mucha libertad para que el alumno elabore su respuesta. Para propósitos de corrección, se deben verificar los siguientes elementos en la respuesta del alumno:

- Actividades previas al primer sprint (1 punto)
    - Esto incluye al menos la comunicación con el cliente, la elaboración de los relatos de usuario y el product planning.
    - Esto debe figurar en la respuesta del alumno dado que el enunciado dice claramente explicitar las actividades desde el inicio del proyecto y por el lado del Kanban se explicita que los días señalados corresponde a aquellos transcurridos desde que se comienza a programar.
- División del proyecto en sprints (1 punto)
    - Se debe señalar la semana en que cada sprint comienza
    - Cada sprint debe tener una duración fija
    - La duración de los sprints puede ser de entre 2 y 4 semanas (1 semana es muy poco dado que hay tareas que requieren más tiempo que eso para cerrarse completamente).
- Desglose de tareas en cada sprint (3 puntos)
    - Se debe respetar la prioridad de las tareas
    - El desglose exacto dependerá de la duración de sprint elegida
    - El desglose elegido debe estar debidamente justificado
- Otras actividades de los sprints (1 punto)
    - Se debe mencionar a grandes rasgos que los sprints incluyen al menos sprint planning, review y retrospective

Los ayudantes pueden descontar hasta 1 punto por aspectos formales (ortografía, gramática, mala presentación…)

Pauta Pregunta 3:

Interesa más el criterio y la forma de enfrentarlo que los números

Se dispone solo de 4 semanas por lo que si queremos un desarrollo ágil, capaz de corregir desvíos e introducir cambios sería conveniente hacer sprints de 1 semana. Si consideran sprints de 2 semanas no está mal, pero a lo menos deberían hacer esta consideración.

Si usamos una escala exponencial (1 simple, 2 simple a intermedio, 4 intermedio a complejo y 8 complejo), tenemos un total de :
Puntos de Relato = 10 x 1 + 6 x 2 + 8 x 4 + 2 x 8 = 70 sp

El equipo es capaz de entregar entre 14 y 18 sp por semana. Luego el número de semanas necesario iría desde:
Mejor Caso: 70/18 = 3.9 semanas
Peor Caso: 70/12 = 5.8 semanas

Plan de Desarrollo del Primer Release

- Dado que por lo general la realidad se acerca más al peor caso es mas probable que el proyecto completo requeriría 6 semanas para completarlo.
- Desarrollo en 4 sprints de 1 semana cada uno
- No hay posibilidad de ajustar por tiempo por lo que si existe ajuste será por alcance.
- Elegir cuidadosamente los relatos a desarrollar en cada sprint de modo de evaluar la prioridad con los interesados.
- Ajustar la productividad del equipo (sp/week) al término de cada sprint. De acuerdo a ello alertar tempranamente a los interesados cuales relatos no estarán presentes en el primer release.

Nota de Pauta:

- Es aceptable el uso de otra escala (fibbonaci) para las 4 categorías de complejidad de los relatos
- Es importante considerar mejor y peor caso en las decisiones
- Es aceptable hacer 2 sprints de dos semanas pero debe estar la reflexión de que eso quita agilidad
- Debería descontarse cualquier complicación de cálculo con días en lugar de semanas porque lo que se conoce es productividad por semana
- Si hay ajuste DEBE ser por alcance porque se decía que el plazo de 4 semanas es inamovible
- Debe aparecer la necesidad de priorizar cuidadosamente