



Pontificia Universidad Católica de Chile  
Escuela de Ingeniería  
Departamento de Ciencia de la Computación

## IIC2143 – Ingeniería de Software

### Interrogación 1 - Solución

#### Instrucciones:

- Sea preciso: no es necesario escribir extensamente pero sí ser preciso.
- En caso de ambigüedad, utilice su criterio y explicita los supuestos que considere convenientes.
- Responda y entregue cada pregunta en hojas separadas. Si no responde una pregunta debe entregar de todas formas la hoja correspondiente a la pregunta.
- Indique su nombre en cada hoja de respuesta.
- Esta interrogación fue diseñada para durar 120 minutos.

1. (0.5 pts) Explique brevemente 4 desafíos de la ingeniería de software.

- Resolver necesidades reales
- Valor > Costo
- Resultados confiables y seguros
- Cumplir con plazos y metas
- Procesos de desarrollo replicables

2. (0.7 pts)

a. Explique qué es un desarrollo iterativo.

Desarrollos parciales sobre todas las funcionalidades de manera simultáneamente.

b. Explique qué es un desarrollo incremental.

Desarrollos completos de las funcionalidades de manera secuencial.

- c. Con un ejemplo concreto en el ámbito del *software*, compare cómo se abordaría un mismo proyecto de forma iterativa contrastado a una forma incremental.

En el caso del proyecto del curso (creación de un foro), si se realizara de forma iterativa se irían realizando distintas funcionalidades pero con un manejo básico al principio. Por ejemplo poder publicar un *post* pero que solo permita formato de texto en un comienzo, acompañado de otras funcionalidades sin un manejo completo. Luego en iteraciones siguientes se completarían todas funcionalidades hasta llegar a las metas deseadas.

Para el caso incremental esta misma funcionalidad se realizaría hasta el final, es decir de forma completa para luego realizar otra. El usuario podría subir *post* escritos acompañados de imágenes, videos, encuestas, etc...

3. (0.6 pts) Explique la práctica conocida como *pair programming*, detallando sus características principales, beneficios y desventajas.

*Pair programming* es una técnica de desarrollo potenciada por las metodologías ágiles. En esta práctica dos desarrolladores trabajan juntos en una estación de trabajo, tomando los roles de “navegante” y “conductor”. La persona que toma el rol de “navegante” tiene la responsabilidad de revisar el código que se desarrolla, junto con guiar el trabajo. El “conductor” es el encargado de plasmar en código el trabajo realizado por ambos. Es importante que ambas personas estén concentradas en sus labores, lo que se potencia con un intercambio de roles frecuente.

#### Beneficios:

- Mejora la calidad del resultado al ser discutido y comentado por dos desarrolladores
- Aumenta la satisfacción frente al trabajo y potencia una cultura de equipo, al enfrentar problemas de manera conjunta
- Facilita la transferencia de conocimiento entre personas del equipo y el sentimiento de *code ownership*

#### Desventajas:

- Requiere más “horas hombre” para desarrollar código, que si no se utilizan de manera adecuada pueden ser un desperdicio de recursos (por ejemplo, realizar tareas mecánicas y de baja dificultad en pareja).
- Para ser efectiva requiere que ambos participantes tenga una participación activa. De no ser así no se podrá obtener ningún beneficio asociado a esta práctica.
- Debe existir respeto y comunicación efectiva entre la pareja para que el trabajo sea efectivo.

4. (0.6 pts)

- a. Defina los roles (junto con sus responsabilidades) de un equipo SCRUM.

Product owner: representa a los *stakeholders* y es el encargado de que el producto genere valor para los usuarios. Dentro de sus responsabilidades está proponer el alcance de los *sprints*, definir los *releases* del proyecto, comunicar el avance del proyecto y negociar con el cliente.

SCRUM master: facilita la correcta implementación de las prácticas de SCRUM dentro del desarrollo. Algunas de sus responsabilidades son ayudar al *product owner* a controlar el avance del proyecto, aclarar los requisitos al equipo, guiar y resolver problemas del equipo de desarrollo.

Development team: son los miembros del equipo responsables de desarrollar el producto. Es un equipo auto-gestionado, donde sus integrantes definen cómo lograrán las metas de cada *sprint*.

- b. Describa qué es un *scrum meeting*, cuáles son sus requisitos mínimos y qué beneficios aporta en el desarrollo de un proyecto.

*Daily scrum meeting* es una pequeña reunión diaria del equipo de desarrollo (no más de 20 minutos). En esta reunión cada integrante responde las preguntas:

- ¿Qué trabajo realicé ayer?
- ¿Qué trabajo planeo realizar hoy?
- ¿Qué problemas veo para ello?

Los beneficios que aporta son:

- Mantiene sincronizado al equipo con el trabajo que todos están realizando
- Transparenta los problemas que existen, y genera instancias para colaborar sobre estos
- Permite detectar riesgos del *sprint* tempranamente y corregir el rumbo

5. (0.4 pts) Imagine que forma parte del equipo de desarrollo de una nueva plataforma de foros de discusión. Basados en la metodología *SCRUM*, su *product owner* le pide que el trabajo se gestione a través de un tablero *Kanban*.

a. ¿Qué beneficios presenta esta herramienta para el desarrollo del proyecto?

- Visualizar el trabajo
- Limitar el trabajo simultáneo
- Potenciar el flujo de tareas

b. Nombre y describa las columnas que utilizaría en su tablero.

- *To do*: tareas faltantes para completar el proyecto
- *Current sprint*: tareas que se realizarán durante el *sprint* actual
- *In progress*: tareas que el equipo está haciendo en este momento
- *Done*: tareas completadas

6. (1.2 pts) Responda si las siguientes afirmaciones son verdaderas (V) o falsas (F). Justifique todas las respuestas.

a. F Las metodologías de desarrollo ágiles presentan más riesgo para un proyecto de *software* comparadas con otras metodologías ya que son menos estructuradas.

El desarrollo ágil permite anteponerse a los riesgos de forma oportuna y abordarlos de manera más flexible.

b. F Según la metodología RUP, en la fase de *Elaboration* se definen los casos de uso del sistema.

*Elaboration* corresponde a la fase de planificación y detección de riesgos de los casos de uso definidos en la fase de *Inception*.

c. F Tanto casos de uso como historias de usuarios describen cómo un actor logra una meta a través del sistema.

Ninguno de los 2 se centran en cómo se logran las metas, sino que en definir cuáles son estas metas.

d. F El concepto *stakeholders* representa a las entidades con poder de decisión sobre un proyecto.

El concepto de *stakeholder* es más amplio: no solamente incluye a las personas con poder de decisión sobre el proyecto, sino que a las entidades interesadas o afectadas por este.

- e. F Un diagrama de casos de uso gráfica las interacciones entre los procesos de un sistema y sus datos.

Un diagrama de casos de uso gráfica las interacciones de los actores sobre el sistema, sin considerar los datos.

- f. F Durante el *sprint review* se reflexiona sobre el trabajo realizado, para así mejorar el proceso del siguiente *sprint*.

*Sprint review* corresponde a la revisión de las funcionalidades desarrolladas junto con una demo. En esta instancia participa todo el equipo y los representantes del cliente.

## 7. (2 pts) Análisis de proyecto

En la actualidad existen variadas aplicaciones de mensajería para teléfonos inteligentes (*WhatsApp, Telegram, Facebook Messenger, Allo*, entre otras), pero ninguna consolidada en el mercado. Ante este escenario, se le solicita que defina los principales requisitos funcionales para una aplicación que quisiera ofrecer este mismo servicio, pero ser atractiva para los usuarios.

Para ello se le pide que escriba 2 épicas distintivas del problema, cada una con 4 historias de usuarios relevantes, acompañadas de sus condiciones de satisfacción. Considere que estas historias de usuarios ayudarán a que la nueva aplicación sea atractiva en el mercado de la mensajería. Escoja nombres auto-explicativos para identificarlas.

Ejemplo de épica: Como un usuario común quiero poder realizar una videollamada para comunicarme con otras personas de manera más enriquecedora.

1. Como usuario común quiero poder utilizar filtros sobre la imagen o mi rostro durante una videollamada con el fin de hacer más lúdica la conversación.
  - a. Que un usuario pueda seleccionar un filtro de imagen.
  - b. Que se reconozca el rostro del usuario y se aplique un filtro especial sobre él.
2. Como usuario común quiero poder compartir la pantalla de mi celular durante una videollamada con el fin de mostrarle a la otra persona el contenido de esta.
  - a. Que el usuario tenga un botón que le indique que puede compartir su pantalla.
  - b. Que al compartir pantalla esta se vea en la pantalla del celular de la otra persona (o personas) con la que se está hablando.
3. Como usuario común quiero poder hacer videollamadas grupales con más de 1 persona con el fin de compartir o conversar con varias personas a la vez.
  - a. Al iniciar una videollamada con 1 persona tener un botón para agregar más gente.

- b. Al seleccionarlo, se muestran los usuarios disponibles para unir a la conversación.
  - c. Seleccionar al nuevo usuario y llamarlo.
  - d. Agregarlo cuando acepte la llamada.
- 4. Como usuario común quiero poder descargar la videollamada con el fin de tenerla guardada para cuando quiera volver a verla.
  - a. Al finalizar la llamada dar la opción de descargarla a través de un botón.
  - b. Al seleccionarlo el video se descargará y guardará en la galería del teléfono como un video.