

IIC 2143 Ingeniería de Software  
Interrogación 1 - Semestre 1 /2019  
Secciones 01 y 02

*Responda cada pregunta en hoja separada*

*Entregue una hoja con su nombre para cada pregunta aunque sea en blanco*

*Tiempo: 2:00*

*Recuerden que están bajo el código de honor*

**Pregunta 1:** (Procesos de desarrollo de software)

Una problemática presente en numerosas residencias del país es el cómo lidiar con enseres (objetos grandes y abultados que por antigüedad o desperfecto uno se quiera deshacer, como muebles o electrodomésticos). A pesar de que existen servicios que permiten recoger estos productos, la empresa *BestPlaceToWork* estima que sería de interés para la comunidad el ofrecer una aplicación que le permita a la gente ofrecer de manera gratuita o a bajo costo sus enseres a otras personas que les podrían interesar, en vez de simplemente deshacerse de ellos. Para ello, deciden contratar a la empresa local *Initech* con el fin de construir dicha aplicación.

La idea, explica Juan, el gerente de *BestPlaceToWork*, es que las personas puedan subir información de sus enseres y el precio de venta. Los eventuales compradores deben por su lado ser capaces de navegar el listado de enseres disponibles, buscar alguno en particular mediante palabras claves y poder concretar una transacción si encuentra algo que le llame la atención. Juan agrega que como mínimo la aplicación debe soportar *Transbank* como plataforma de pago, pero que *BestPlaceToWork* contrató a una consultora que se encuentra estudiando otros medios de pago y, según el resultado de este estudio, puede que se soliciten nuevos medios más adelante.

Juan explica que por ahora el transporte de los enseres será responsabilidad de, o bien el comprador, o bien el vendedor; pero a largo plazo, buscarán asociarse con una empresa de despacho para incluir el transporte y tracking de estos productos como parte de su servicio. Finalizada la transacción y con el producto en su poder, la aplicación debe permitirle al comprador incluir una reseña que describa su experiencia con el vendedor y la aplicación misma. Por último, Juan enfatiza que este es un proyecto clave para *BestPlaceToWork*, y que es muy importante que la aplicación esté terminada en no más de 1 año, dado que fuentes de la empresa revelan de que una compañía rival está trabajando en una aplicación similar y es imperativo salir al mercado antes que ellos.

*Initech* es una empresa arraigada en la tradición, y como tal, solo tienen experiencia desarrollando proyectos de software utilizando las metodologías cascada, espiral y RUP.

- a) Explique cuál de los tres procesos de desarrollo anteriores sería el más apropiado para este proyecto, detallando también por qué los otros dos serían inadecuados. *Desarrolle su respuesta en no más de 10 líneas.* (2 puntos)
- b) Dado el proceso que eligió en a), realice una planificación gruesa de las actividades a desarrollar durante el año de trabajo para que el proyecto resulte exitoso. No olvide indicar

en qué momento del proyecto se realizaría cada actividad. *Desarrolle su respuesta en no más de 20 líneas. (4 puntos)*

**Pregunta 2:** (Procesos ágiles)

*Lea el siguiente relato y responda las preguntas que vayan surgiendo en no más de 5 líneas cada una.*

La empresa *Initech*, especialista en procesos de desarrollo de software tradicionales, ha tomado la difícil decisión de cambiar sus metodologías de trabajo y unirse al movimiento ágil de desarrollo que se ha vuelto tan popular en el mundo. Para eso, deciden contratarlo a usted como consultor para que capacite a su personal en el proceso de transición. Usted decide comenzar su capacitación explicándole al personal los conceptos esenciales de Scrum, partiendo por el concepto de Sprint.

Los gerentes de *Initech* se empiezan a mostrar inquietos una vez que usted insiste en que estos Sprints deben ser cortos y tener una duración fija. Ellos sugieren que tiene mucho más sentido hacer Sprints largos y de duración variable según las funcionalidades que se estén atacando en un determinado momento.

- a) Explique el motivo detrás de la duración fija de los Sprints y por qué lo que proponen los gerentes es una mala idea. (1 punto)

Los gerentes se espantan cuando usted trata de explicarles de que no es necesario acordar mediante contrato la totalidad de los requisitos de un proyecto antes de que este comience. En particular, les preocupa que un cliente pueda aprovecharse de la empresa y agregar requisitos en forma descontrolada a medida que avanza el proyecto.

- b) Explique por qué en los procesos ágiles en general descartan la idea de acordar todos los requisitos de antemano y cómo es que Scrum aborda el control de cambios y evita el problema enunciado por los gerentes. (1 punto)

Usted trata de explicarle a la empresa que Scrum favorece la creatividad del equipo de desarrollo dándole la libertad de implementar las soluciones de la manera que ellos estimen más conveniente. Los gerentes arguyen que un enfoque de esa naturaleza dificulta el seguimiento del avance del proyecto y puede producir que individuos se aprovechen y eludan sus labores.

- c) Explique qué mecanismos ofrece Scrum para enfrentar estas dos situaciones. (1 punto)

A los gerentes les agrada la figura del ScrumMaster en Scrum, ya que permite designar a un encargado con autoridad para zanjar los desacuerdos, fijar las prioridades del Backlog y mantener la disciplina en el equipo.

- d) Comente esta aseveración de los gerentes. (1 punto)

Los gerentes se muestran dubitativos y solicitan conocer otros enfoques ágiles para contrastar con Scrum. Así es como usted procede a hablarles de Kanban. Es entonces que los gerentes notan que un posible riesgo de Kanban es abultar el tablero con tareas “En Progreso”, sin ir cerrando features que le sean relevantes al cliente.

- e) Explique cómo es que Kanban promueve el cierre de features y que todo el equipo esté constantemente trabajando. (1 punto)

Al finalizar su disertación, muchos gerentes parecen quedar con la idea de que Kanban es casi lo mismo que Scrum.

f) Explique las similitudes y diferencias entre Kanban y Scrum. (1 punto)

### **Pregunta 3:** (Relatos de usuario)

Una municipalidad de Santiago ha decidido contratar a la empresa *Ozom* para desarrollar una aplicación móvil-web que permita a los habitantes de la comuna participar en foros, realizar denuncias, facilitar el pago de multas y obtener información de interés promovida por la municipalidad (eventos, campañas publicitarias e instructivos). La municipalidad le explica a *Ozom* que tenga en cuenta las siguientes consideraciones:

- La aplicación debe ser de utilidad para los miembros de la comuna y promover la comunicación. En especial, debe poder ser utilizada por aquellos miembros de la tercera edad o personas con discapacidades.
- Los foros requieren de moderadores que no necesariamente tienen que ser personal de la municipalidad.
- Una denuncia puede ser desde un inconveniente relativamente menor (ej. un agujero en los caminos) hasta un hecho grave (ej. un robo en progreso).
- En donde se pueda, sería interesante permitir a los usuarios subir imágenes o videos.
- Para realizar pagos, sería ideal soportar más de un medio de pago.
- La aplicación debe ofrecer confianza a sus usuarios e idealmente ofrecer un medio eficiente para atender consultas acerca de problemas al usar la herramienta

- a) Dada la descripción anterior, identifique los principales roles que pueden asumir los usuarios de la aplicación. (1 punto)
- b) Haga un levantamiento de requisitos mediante relatos de usuario. Si lo estima conveniente, puede organizar sus relatos en forma de épicas, pero no es obligatorio. Al menos 3 de los relatos que defina deben incluir condiciones de satisfacción. (4 puntos)
- c) Mencione brevemente qué requisitos no-funcionales debe soportar la aplicación. (1 punto)

### **Pregunta 4:** (Aspectos generales de ingeniería de software)

*Responda las siguientes preguntas en no más de 5 líneas cada una:*

- a) Enuncie y explique dos puntos del Agile Manifesto. (1 punto)
- b) Si la construcción de software implica en general solamente mano de obra, ¿por qué los proyectos de software suelen ser tan caros? (1 punto)
- c) Hoy en día, sabemos que Cascada o Waterfall tiene una alta probabilidad de llevar a proyectos fallidos o “challenged”. ¿Por qué entonces esta metodología fue tan popular en los 90 e incluso el día de hoy se usa en algunas partes? (1 punto)
- d) Explique los conceptos de “Lead Time” y “Cycle Time” en Kanban. (1 punto)
- e) ¿Qué es el *WIP* en Kanban y cuál es la razón de poner un límite máximo? (1 punto)
- f) Explique la diferencia entre “output” y “outcome” y por qué los procesos ágiles enfatizan el segundo. (1 punto)

### **Pauta Pregunta 1:**

- a) Explique cuál de los tres procesos de desarrollo anteriores sería el más apropiado para este proyecto, detallando también por qué los otros dos serían inadecuados. (2 puntos)

Del texto sabemos que los requisitos son inciertos dado que nuevos medios de pago pueden agregarse en el futuro. Además, sabemos que el proyecto es clave para la empresa, por lo que reducir la ansiedad del cliente es importante. También sabemos que el proyecto no puede tardar más de 1 año, por lo que necesitamos una metodología que sea estricta en los plazos.

Cascada resulta inadecuado ya que no admite cambios en los requisitos a mitad de proyecto. Además, el proyecto es bastante largo (1 año) y es probable que se presenten muchos cambios.

Espiral resulta inadecuado ya que las constantes evoluciones del producto impiden estimar bien la duración del proyecto, con lo cual el cumplimiento del plazo de 1 año resulta bastante incierto.

RUP permite satisfacer todas las condiciones anteriores: se enfoca en la reducción del riesgo mediante una planificación temprana coordinada con el cliente, admite cambios en los requisitos durante el avance del proyecto, la división en iteraciones con demos al cliente permite reducir la ansiedad, y es estricta en el cumplimiento de los plazos.

**Nota de pauta:** Se espera que el alumno indique 2 motivos a favor de RUP, 1 motivo en contra de espiral y 1 motivo en contra de cascada. Asignar 0.5 puntos por cada uno.

**Nota de pauta:** Si un alumno elige un modelo distinto a RUP, omitir la evaluación del inciso b) y calificar todo este ejercicio con 0 puntos.

- b) Dado el proceso que eligió en a), realice una planificación gruesa de las actividades a desarrollar durante el año de trabajo para que el proyecto resulte exitoso. (4 puntos)

Los alumnos deben describir las 4 fases de RUP:

Inception: Se identifican los requisitos principales y las necesidades del cliente. Se estudia viabilidad del proyecto. Se estiman recursos necesarios.

Elaboration: Se trabaja en un listado de requisitos más acabado, se trabaja en arquitectura y diseño (modelado). Se arma un plan de trabajo más detallado. Pueden incluirse pruebas de concepto tempranas.

Construction: Se programa el software, incluyendo el testing. Se divide en iteraciones. Requisitos pueden ir levemente adaptándose según feedback del cliente entre iteraciones. Posibles iteraciones: 1) subir enseres por parte de los vendedores, 2) buscar y comprar enseres por parte de los compradores, 3) reseñas de la experiencia.

Transition: Se hace el deployment de la solución. Se realizan los ajustes finales. Se entrega documentación final y se hace capacitación al cliente de la solución. Puede incluir una marcha blanca.

**Nota de pauta:** RUP admite iteraciones en sus distintas etapas, pero en esta evaluación, solo se exige que los alumnos precisen iteraciones en construcción. Se deja libertad para precisar las actividades a realizar en cada iteración mientras sean razonables. Restar 1 punto si no ponen ninguna iteración en construcción. No se acepta como parte de las actividades propuestas temas en relación con el despacho, dado que se explicita que se trata de un plan a futuro. Restar 0.5 puntos si el alumno alude a esta actividad.

**Nota de pauta:** La transición no incluye mantención de la solución. Restar 0.5 puntos si un alumno indica lo contrario. No obstante, se admite que un alumno diga que después del primer año, puede incluirse una etapa de mantención adicional.

**Nota de pauta:** Los alumnos deben especificar la duración de cada fase. Admitir cierta flexibilidad en las estimaciones de los alumnos. Los siguientes intervalos, en relación con la duración total del proyecto, son aceptables: inception 5-15%, elaboration 10-30%, construction 50-70%, transition 5-15%. Para la duración de las iteraciones de la construcción, se admiten duraciones del orden de meses. Si el alumno no pone duraciones, restar 1 punto. Si el alumno pone duraciones muy alejadas de lo aquí propuesto, restar 0.5 puntos.

## **Pauta Pregunta 2:**

- a) Explique el motivo detrás de la duración fija de los Sprints y por qué lo que proponen los gerentes es una mala idea. (1 punto)

Motivos a favor de la duración fija de los sprints:

- Obliga a priorizar
- Obliga a ir cerrando features
- Demuestra progreso del proyecto
- Evita perfeccionismo innecesario
- Mejora predictibilidad
- Mantiene la motivación del equipo alta
- Permite ir replanificando frecuentemente

Sprints demasiado largos son difíciles de planificar y puede promover el pasar mucho más tiempo del necesario en un feature. Al ser más largo, también reduce el número de sprints de un proyecto, con lo cual se reduce el número de instancias para obtener feedback y el número de instancias para hacer ajustes o replanificaciones del proyecto.

Sprints de duración variable desfavorece la disciplina en el equipo de trabajo y dificulta la planificación a futuro.

**Nota de pauta:** Se espera que el alumno entregue al menos 2 motivos a favor de la duración fija de los Sprints y de un motivo en contra de que sean demasiado largos, y un motivo en contra de que tengan duración variable.

- b) Explique por qué en los procesos ágiles en general descartan la idea de acordar todos los requisitos de antemano y cómo es que Scrum aborda el control de cambios y evita el problema enunciado por los gerentes. (1 punto)

Se descarta el planificar todo de antemano porque:

- Es imposible predecir todo lo que se va a tener que realizar durante el transcurso de un proyecto muy largo
- Es extremadamente caro el plasmar todos los requisitos de antemano en un documento
- No permite ir realizando ajustes al proyecto, sea por cosas omitidas por el cliente en un comienzo o por novedades en la industria
- Es susceptible de promover interpretaciones erradas de los requisitos si no hay comunicación

Control de cambios se aborda en los sprint reviews. En esta instancia se demuestran avances al product owner y el puede ofrecer feedback, solicitar correcciones o cambios.

Para cambios muy grandes, el problema enunciado por los gerentes se soluciona mediante una renegociación la cual puede implicar aumentar el presupuesto del proyecto, o bien, más

comúnmente, ofrecerle al cliente sacar features menos prioritarios para mantener el esfuerzo estimado del proyecto.

**Nota de pauta:** Se espera que el alumno entregue al menos 2 motivos en contra de planificar todo de antemano.

c) Explique qué mecanismos ofrece Scrum para enfrentar estas dos situaciones. (1 punto)

Scrum ofrece dos mecanismos para hacer trazabilidad de los proyectos: el daily meeting/stand-up meeting y el sprint review.

Daily meeting es la reunión diaria (o cada X días) en donde cada miembro del equipo debe responder brevemente qué hizo, qué problemas encontró y qué es lo que va a hacer. El sprint review es donde se inspecciona el producto y incluye una demo con el cliente para recibir retroalimentación.

El daily meeting en particular permite evitar de que un miembro del equipo no esté haciendo nada dado que lo obliga a exponer en frente de todo el equipo de trabajo qué está haciendo.

**Nota de pauta:** No basta enumerar Daily meeting y Sprint Review, hay que explicar en qué consisten.

d) Comente esta aseveración de los gerentes. (1 punto)

Los gerentes malentienden el rol del ScrumMaster. Él no es un jefe que decide por el equipo. Su rol es el de apoyar al equipo de desarrollo ofreciendo coaching, moderando discusiones y promoviendo el llegar a un consenso. En particular se encarga de facilitar de que todos estén siguiendo lo mejor posible las prácticas ágiles. Adicionalmente, el ScrumMaster no puede priorizar las tareas del backlog. Sí puede ayudar al Product Owner en esta tarea y sugerir técnicas para manejar de mejor forma el backlog.

e) Explique cómo es que Kanban promueve el cierre de features y que todo el equipo esté constantemente trabajando. (1 punto)

Kanban consigue lo anterior limitando el número de tareas que pueden estar en un determinado conjunto de columnas. Si no se pueden mover features desde una columna hacia otra debido a que el equipo de esta última está atascado, Kanban promueve la ayuda inter-equipo, con lo cual aquellos que no tienen en qué trabajar pueden asistir a sus colegas para cerrar features más rápido.

f) Explique las similitudes y diferencias entre Kanban y Scrum. (1 punto)

Similitudes:

- Los dos son procesos ágiles
- Los dos pueden visualizar tareas en un tablero



- Los dos promueven el ir cerrando features para promover el avance del proyecto
- Los dos otorgan independencia a los desarrolladores para implementar los features

Diferencias:

- Scrum limita el número de tareas en una iteración, Kanban el número de tareas en un determinado estado
- En Scrum los equipos son polifuncionales, en Kanban esto es opcional
- Scrum es más prescriptivo, Kanban es más bien un conjunto de principios
- En Scrum el tablero se resetea entre los sprints, en Kanban es global
- En Scrum existe un ScrumMaster que actúa de facilitador y mediador, no existe un rol análogo en Kanban

**Nota de pauta:** Se espera que el alumno entregue al menos 2 similitudes y 2 diferencias.

### **Pauta Pregunta 3:**

- a) Dada la descripción anterior, identifique los principales roles que pueden asumir los usuarios de la aplicación. (1 punto)

Posibles roles de los usuarios son:

- Residente de la comuna
- Moderador (de los foros)
- Funcionario de la municipalidad (definen la información que se muestra en la página)
- Administrador

**Nota de pauta:** Se admite que los alumnos definan más roles si están bien justificados. Al menos debiesen identificar 3 roles distintos.

- b) Haga un levantamiento de requisitos mediante relatos de usuario. Si lo estima conveniente, puede organizar sus relatos en forma de épicas, pero no es obligatorio. Al menos 3 de los relatos que defina deben incluir condiciones de satisfacción. (4 puntos)

Ejemplos de relatos de usuario:

- **Participar en foro:** Como residente de la comuna, quiero acceder a un foro, leer su contenido y enviar mensajes para poder participar en tópicos que sean de mi interés.
- **Moderar foro:** Como moderador, quiero poder suprimir un mensaje ofensivo o inapropiado para evitar conflictos entre los residentes de la comuna.
- **Realizar denuncia:** Como residente de la comuna, quiero denunciar un evento potencialmente peligroso para que la municipalidad realice gestiones para solucionar el problema.
- **Emitir alerta:** Como residente de la comuna, quiero realizar una alerta de un delito en progreso para dar advertir a mis vecinos y promover una rápida respuesta de carabineros.
- **Ver denuncias:** Como residente de la comuna, quiero ver el listado de denuncias publicados por mis vecinos para saber de qué me tengo que cuidar.
- **Resolver denuncia:** Como funcionario de la municipalidad, quiero marcar una denuncia como resuelta luego de confirmar que fue atendida para hacerle saber a la comunidad que el problema ya fue resuelto.
- **Publicar eventos:** Como funcionario de la municipalidad, quiero publicar relacionada a eventos para invitar a miembros de la comunidad a asistir.
- **Publicar campañas:** Como funcionario de la municipalidad, quiero publicar información ligada a campañas publicitarias para informar a la comunidad de información que tienen que saber.
- **Ver multas:** Como residente de la comuna, quiero ver mi registro de multas impagas para saber si debo cancelar algo.
- **Pagar multa:** Como residente de la comuna, quiero poder pagar una multa impaga a mi nombre para saldar mis deudas con la municipalidad.

- **Publicar multa:** Como funcionario municipalidad, quiero emitir una multa a nombre de un residente de la comuna para que esta pueda gestionar su pago o apelación vía la plataforma.
- **Designar funcionario municipal:** Como administrador, quiero poder definir qué usuarios van a poder tener el acceso de un funcionario municipal a la plataforma para controlar quién puede acceder a las funciones más avanzadas de la plataforma.

**Nota de pauta:** En esta pauta se mencionan solo algunos ejemplos, hay muchos relatos posibles más que pueden concebir los alumnos. Se considerarán válidos siempre y cuando sigan el formato de “Quién – Qué – Por qué” y sean atingentes al enunciado del ejercicio. Se espera que los alumnos identifiquen al menos 8 relatos de usuario. Verificar que al menos 3 incluyan criterios de satisfacción. Descontar 0.5 puntos por cada relato de usuario mal formulado o no atingente al enunciado. Si ponen menos de 8 relatos, descontar 0.5 puntos por cada relato faltante. Descontar 0.5 puntos por cada conjunto de criterios de satisfacción mal enunciado. Si ponen menos de 3 relatos con criterios de satisfacción, descontar 0.5 por cada uno que falte.

c) Mencione brevemente qué requisitos no-funcionales debe soportar la aplicación. (1 punto)

La aplicación debe poder ser utilizada por miembros de la tercera edad, así que **usabilidad** es importante. También debe poder ser usada por personas con discapacidad, por lo que **accesibilidad** es otro requisito no-funcional. Debe ofrecer confianza, por lo que debe tener **seguridad** y debe respetar la **privacidad** de los usuarios. Por último, debe ofrecer **serviceability** para asegurar una respuesta rápida y efectiva frente a problemas con la herramienta.

**Nota de pauta:** Basta con que los alumnos identifiquen 2 requisitos no-funcionales para obtener todo el puntaje. Si describen algo que no califica de requisito no-funcional, descontar 0.5 puntos.

#### **Pauta Pregunta 4:**

a) Enuncie y explique dos puntos del Agile Manifesto. (1 punto)

- Individuos e interacciones por sobre procesos y herramientas: el trabajo en equipo y la buena convivencia es más importante que simplemente seguir ciegamente un conjunto estructurado de reglas.
- Software funcional por sobre documentación exhaustiva: es más importante avanzar en un software que ofrezca funcionalidades acabadas que dedicarle mucho tiempo a documentar cada aspecto del software.
- Colaboración con el cliente por sobre negociación de contrato: es más importante integrar de alguna forma al cliente al equipo de trabajo e ir obteniendo constantemente retroalimentación de su parte que el decidir de antemano todos los aspectos de un proyecto y fijar lo anterior bajo un contrato inmodificable. Así es más probable llegar a un resultado que sea satisfactorio para el cliente.
- Responder al cambio por sobre seguir un plan: es más importante tener la capacidad de adaptarse a la mitad de un proyecto frente a imprevistos o cambios en los requisitos que planificar todo mediante una carta Gantt exhaustiva en un inicio.

b) Si la construcción de software implica en general solamente mano de obra, ¿por qué los proyectos de software suelen ser tan caros? (1 punto)

Para asegurar la calidad de un software, gran parte del trabajo está dedicado a fases de análisis, diseño, testing y validación, además de la codificación. Todo lo anterior requiere de un equipo multidisciplinario a quienes hay que pagarles el salario durante todos los meses que dure un proyecto. Esto cobra mayor relevancia si consideramos el gran costo que puede tener un bug (desde leaks masivos de datos hasta pérdida completa de hardware). Por sobre lo anterior, es necesario además cobrar un margen adicional que cubra eventuales riesgos de un proyecto, los costos fijos de una empresa y que además le permita obtener un beneficio.

c) Hoy en día, sabemos que Cascada o Waterfall tiene una alta probabilidad de llevar a proyectos fallidos o “challenged”. ¿Por qué entonces esta metodología fue tan popular en los 90 e incluso el día de hoy se usa en algunas partes? (1 punto)

El modelo de cascada es:

- muy simple de entender
- funciona en otras áreas de la ingeniería
- permite dibujar cartas gantt para el proyecto entero
- permite separar tareas y equipos asociados
- le da al jefe de proyecto un sentido de avance

**Nota de pauta:** Los alumnos deben enunciar al menos 2 razones sobre el por qué algunos siguen usando cascada.

d) Explique los conceptos de “Lead Time” y “Cycle Time” en Kanban. (1 punto)

Lead Time: Tiempo que transcurre entre que una tarea entra al backlog y esta se completa.

Cycle Time: Tiempo que transcurre entre que se empieza a trabajar en una tarea y se termina de codificar y testear.

e) ¿Qué es el WIP en Kanban y cuál es la razón de poner un límite máximo? (1 punto)

El WIP (work in progress) es el número máximo de tareas que puede haber en una determinada columna del tablero (por ejemplo, en desarrollo, en testing). La razón para poner este límite es maximizar el flujo y por lo general requiere ajustes durante el proceso.

f) Explique la diferencia entre output y outcome y por qué los procesos ágiles enfatizan el segundo. (1 punto)

Output se refiere simplemente al producto del proyecto de software propiamente tal y alude al número de funcionalidades que ofrece un determinado programa. Outcome se refiere al impacto que genera este programa en sus eventuales usuarios y cuán útil resulta para ellos. Enfatizar el outcome permite maximizar la satisfacción del cliente enfocándose en lo esencial. Un gran output no necesariamente implica un proyecto exitoso.