

Actividad práctica de Lean Development

Introducción

Sois los integrantes de un equipo de trabajo que tiene que crear una aplicación para un supermercado siguiendo el modelo de desarrollo de software conocido como Lean Development. Como ya sabréis, esta metodología se basa en aplicar 7 principios fundamentales:

- 1.- Eliminar los desperdicios
- 2.- Amplificar el aprendizaje
- 3.- Decidir lo más tarde posible
- 4.- Entregar lo más rápido posible
- 5.- Capacitar al equipo
- 6.- Construir integridad intrínseca
- 7.- Ver todo el conjunto

Se os plantean una serie de decisiones que deberéis tomar para desarrollar la aplicación de la empresa. En vuestro grupo de trabajo tenéis un experto en cada uno de los 7 principios del desarrollo de software Lean y **cada una de las decisiones que toméis deberá estar justificada en base uno o más de los 7 principios**. Si dos principios entran en conflicto, explicadlo y proponed una solución alternativa. Por cada respuesta correcta, el equipo sumará un punto. Si la respuesta correcta viene justificada por medio del principio adecuado se añadirán otros dos puntos más al equipo. Al final del recuento, el equipo que tenga más puntuación recibirá 5 puntos extra. Vuestro objetivo como equipo es demostrar que sois los que más sabéis de Lean Development obteniendo el máximo número de puntos posibles para ganar aquello por lo que todos estamos aquí, el título del Doble Grado de Ingeniería Informática y Matemáticas, y los 6 créditos de descuento.

Elaboración del proyecto

El supermercado quiere una aplicación para dispositivos móviles que permita realizar una compra online en uno de los supermercados de la franquicia. También se pretende que la aplicación ofrezca un servicio de compra a domicilio desde la propia app. La empresa divaga sobre la posibilidad de implementar otras funcionalidades como poder reservar plaza en el parking para días de mucha clientela y está indecisa sobre ampliar la aplicación apta a otros soportes.

Vuestra primera responsabilidad es elaborar una **lista de requisitos** para la aplicación (no olvidéis la **justificación de vuestras decisiones**):

Una vez establecidas los requisitos tenéis una reunión con vuestros jefes y os plantean distintos **sistemas de trabajo** de entre los cuales debéis **elegir uno**:

- a) Establecer una jerarquización en la que ellos organizarán las tareas y marcarán el trabajo a seguir por vuestro equipo.
- b) Libertad absoluta de vuestro equipo para tomar cualquier tipo de decisión sobre el proyecto sin necesidad de ser consultada.
- c) Establecer un sistema de comunicación entre miembros del equipo que tenga que tener como intermediario el inmediato superior para garantizar la coordinación entre los miembros.
- d) Regirse de manera autónoma mientras los administradores se limitarán explicar el qué se debe hacer y no el cómo.

En la reunión también debéis acordar como se van a dividir las tareas. Hay dos posturas enfrentadas. Por un lado, hay quien pretende que los trabajadores se especialicen únicamente en uno de los requisitos de la aplicación y que ignoren el resto. Otros, sin embargo, apuestas por un sistema en el que la interacción entre los distintos equipos dentro del proyecto sea fundamental. Escoged la opción más acorde con la filosofía Lean.




Antes de comenzar a trabajar, la empresa se pone en contacto con vuestro equipo para entregaros un dossier con el trabajo de otro equipo que fracasó a la hora de satisfacer sus necesidades. Entre la montaña de documentación que os entregan, encontráis un calendario de reuniones con la empresa de supermercados, que muestra que estas se realizaban con poca frecuencia, y un organigrama de distribución de tareas, que refleja que toda la comunicación entre el cliente y los desarrolladores se realizaba a través del manager del equipo. También habéis podido encontrar un libro de Tom y Mary Poppendieck y una lista con funcionalidades añadidas que la empresa de supermercados no había planteado.

Vuestra misión es **aprender de los errores** del equipo que lo había intentado anteriormente y también de sus **aciertos**. ¿Cómo interpretáis la información que habéis obtenido?





Una vez leída toda la documentación y analizada, se os plantea la duda de si vosotros debéis **incluir documentación** en vuestro proyecto.

Ya habéis comenzado a trabajar y debéis decidir la estructura del menú para vuestra aplicación. Habéis llegado a 2 diseños:

a)

| | |
|---|--|
| Hacer pedido  | Envío de compra a domicilio  |
| Ayuda  | "Insertar funcionalidad" |

b)

| | |
|-------------------------|---|
| • Hacer pedido |  |
| • Pago en app |  |
| • Ofertas y descuentos | € |
| • Datos de contacto |  |
| • Mapa de supermercados |  |

¿Cuál decidís escoger y por qué?

Tras escoger el diseño, programáis el menú. Si habéis escogido la opción b), ¿la presentaríais al cliente? Si habéis escogido la opción a), ¿organizaríais una reunión con la empresa de supermercados o esperaríais a que os sugiriera una nueva funcionalidad?

Una vez más estáis ante una disyuntiva. El proceso lean software development es un modelo ágil e iterativo, como scrum, pero estas **iteraciones**, ¿han de ser **cortas o largas**?

Continuáis avanzando en el proyecto manteniendo reuniones sucesivas con el cliente, pero vuestro experto en la eliminación de desperdicios cree que hay muchas cosas en el proyecto que no añaden valor al cliente. Propone realizar un “**mapa de flujo de valor**”, ¿en qué consiste y que beneficios puede aportar?

Habéis conseguido entregar el proyecto y la empresa de supermercados está muy contenta, ¿pero habréis seguido al pie de la letra el desarrollo de software lean?