



Trabajo Final de Unidad 1
Análisis y Diseño de Aplicaciones I
2024
Equipo 4

Grupo TI3B - Ing. José Abadie e Ing. Sebastián Feirrés

Parte 1

En nuestro equipo decidimos utilizar la experiencia de los Retos de Informática de la UCU. Principalmente se distinguen dos tipos de reto, el primero es cuando se crea una aplicación para un cliente externo y el segundo es cuando se crea para la Universidad.

En el caso de crear una aplicación para el cliente, ocurre que muchas veces los requisitos iniciales son confusos, en cuanto a que el cliente no sabe exactamente qué cosas se pueden pedir del software y por lo tanto, dan una idea general acerca de las funcionalidades pero nada específico. Es por ello que se comienza definiendo el alcance de la aplicación haciendo una priorización de requerimientos, ya que al tratarse de proyectos con una duración de 3 semanas, es esencial ser capaz de incluir las funcionalidades fundamentales para no perder el valor que le aportará al negocio.

Esto lleva a un proceso de creación de una aplicación básica que cumple con lo pedido, sin la elaboración previa de un prototipo debido a la limitante de tiempo, que luego se le muestra al cliente y este indica cambios para hacer o pide funcionalidades nuevas, con mayor precisión ya que tiene una idea más clara de cómo es el programa y cómo podría llegar a utilizarlo. En términos de requisitos no funcionales, la forma de crear el software siempre es priorizando la optimización (tanto en memoria como rendimiento), por lo cual no ha habido necesidad de indicar algún atributo de calidad respecto al funcionamiento.

En el caso de crear una aplicación para la Universidad, como los docentes conocen detalladamente el problema a resolver, son capaces de explicar mejor las funcionalidades que debe tener la aplicación. Esto lleva a que el primer acercamiento sea más certero, planificando mejor cómo va a ser el programa y causando que el equipo de desarrollo esté más enfocado en los requisitos funcionales. Adicionalmente, si surgen dudas acerca del funcionamiento, los docentes son capaces de responder con precisión las dudas. Al igual que en el otro caso, tampoco se piden requisitos no funcionales específicos más allá de priorizar la optimización de la aplicación.

En ambos casos, siempre hay una instancia inicial de reunión en la cual se conoce mejor el proyecto y qué se quiere lograr con él. Aquí el cliente cuenta cuál es el objetivo que busca y lo que espera obtener, los alumnos preguntan las dudas que surjan, y también confirman haber entendido al cliente. Para llevar la cuenta de los requisitos, se utilizan distintas herramientas como documentos compartidos con toda la información, repositorios en donde se colocan los requisitos como issues, aplicaciones de intercambio de mensajes donde se escribe lo hablado, o tableros kanban con la capacidad de crear tareas y asignarlas a participantes. Una vez que se comienza el proyecto, las dudas que surgen tanto del funcionamiento como de requisitos, son documentadas para preguntarlas al cliente vía correo electrónico o reunión presencial, en donde se repite el proceso de obtener

requisitos y documentarlos. Este proceso tiene un problema y es que muchas veces la comunicación es lenta, llevando a que haya tiempo perdido entre que se pregunta y se obtiene una respuesta. Una posible solución sería tener una instancia periódica de reunión con el cliente, en donde se clarifiquen las dudas existentes y donde se pueda agregar o cambiar requisitos.

A la hora de realizar la construcción del programa, el equipo se divide en sub equipos, donde cada uno se hace cargo de una parte del proyecto, es aquí donde se toman decisiones específicas de cada área (frontend, backend, testing, etc). A pesar de discutir parte de las restricciones en la reunión inicial, es en estos subequipos que se definen decisiones de diseño, cronograma y entregables.

Parte 2

En un principio, independiente que el problema sea de carácter externo o interno a la universidad, siempre se da una instancia de Blastoff. Esto es lo que mencionamos como reunión inicial, donde se puede ir descubriendo a los distintos stakeholders y con ellos las necesidades más altas del proyecto, para luego poder definir mejor un alcance inicial, y de ahí los siguientes posibles hitos.

Durante la instancia de Blastoff, generalmente se da de manera natural una entrevista, de los estudiantes a la persona/equipo que presenta el problema. Se discutirá y buscará entender el problema, comúnmente en su totalidad, pero a la vez buscando diferenciar los principales puntos a resolver, con la idea de que el reto tiene una duración corta, y los proyectos deben ser bien delimitados para esto.

A pesar de que se considera al análisis de riesgos fundamental en esta instancia, es algo que en los retos no se evalúa. Esto se debe a un tema de tiempos, por lo que en estos proyectos, van surgiendo inconvenientes a lo largo del desarrollo y se solucionan dentro de cada subequipo, lo cual claramente puede llevar a un desenlace grave, como fallos en el programa final, por no hacer este análisis de riesgos.

En el proceso y la falta de tiempo, podríamos decir que utilizamos una cierta forma de *high fidelity prototypes*, siendo que aunque no es un prototipo desechable, si seguimos el concepto de poder ir descubriendo nuevos requisitos a medida que se va construyendo el programa, y en nuestro caso, mostrando los avances al cliente.

El momento de mostrar avances a los clientes, discutir las funcionalidades y si son acordes a lo que se está buscando, así como pensar en posibles necesidades futuras a partir de lo que hizo hasta ahora, sería una forma de BUC workshops, donde se buscaría identificar nuevos BUCs y si los actuales están acertados.

Otra técnica utilizada en retos, fue la técnica de Mind Map. Recordamos tomar un pizarrón con personas del cuerpo docente y los desarrolladores involucrados en el proyecto,

y realizar un mind map para descubrir los requisitos del mismo. Este Mind Map consistió en arrojar ideas, tareas que deberían ser prioritarias y fechas tentativas para los finales de los sprint y features que deberían ser cumplidas en la fecha. Esta técnica logró plasmar mejor en nuestra mente los tiempos del proyecto y las ideas principales a concretar por el mismo.

Por otro lado, cabe destacar que en el corto período de tiempo que dura el reto, muchos de los requisitos se van descubriendo/profundizando a medida que se va desarrollando, por ende, genera una mayor carga en el equipo y presenta problemas en el proceso que no se previeron antes. Una forma que se nos ocurre de tratar de recabar aún más los requisitos antes de iniciar el proyecto es teniendo una entrevista previamente meditada con el cliente, dándonos la posibilidad de plantear nuestras propias preguntas y ser más específicos en ciertos temas, habiendo preparado el material necesario, teniendo más conocimientos del tema, y así poder plantearnos mejor entender qué necesita. También podría ser de gran utilidad tener una reunión con el cliente para hacer un brainstorming, dado que combinando las ideas de todos los stakeholders, se podría llegar a un producto más realizable y claro para ambas partes, siendo más fácil así visualizar los BUCs actuales y cuáles serían sus BEs para poder acatarlos.

Otra problemática que se le podría criticar al reto, sería que dado el reducido período de tiempo, no da para hacer un prototipo low fidelity, dado que lo que vendría a ser el prototipo, termina siendo el producto final. Quizás el tener un poco más de tiempo y crear un prototipo más básico podría ayudar a clarificar más los requisitos. Cuando decimos esto, nos referimos que a medida que vamos desarrollando, vamos encontrando nuevos requisitos, los cuales se suman a lo que ya se venía desarrollando. Esto hace que el producto no sea tan “prolijo”, con nuevas partes que no se habían contemplado y se sumaron luego de ya empezado su desarrollo.