

Chris era el encargado de la versión 1 del sistema de inventario de productos químicos del MIT, el MITCIS. Con la creciente necesidad de seguridad, el MIT se dio cuenta de que su sistema informal de realización de pedidos, envíos y entregas de productos químicos necesitaba evolucionar. Chris tenía una idea general de las capacidades necesarias, ya que conocía el sistema y había asistido a la primera reunión del comité directivo del MIT para el proyecto. Dave, el presidente del comité, le planteó la siguiente pregunta:

"Chris, ¿cuánto tardará el MITCIS en conseguirlo?"

"Creo que se tardará unos 9 meses, pero ten en cuenta que es un cálculo aproximado", contestó Chris.

"No puede ser", argumentó Dave. "Esperaba que me dijese 3 ó 4 meses como mucho. Es imprescindible que podamos disponer del MITCIS en un periodo de 6 meses, **no podemos permitirnos riesgos de seguridad más de 6 meses, en el peor de los casos. Eres consciente de la gravedad del problema, ¿verdad? ¿Podrías hacerlo en 6 meses?**"

*Si quieres hacer un proyecto más rápido pregunta qué medios son necesarios para lograrlo y proporcióñalos. La presión no hace que la gente trabaje más rápido.*

"No estoy seguro", contestó Chris con sinceridad. "Tendría que analizar el proyecto con más detenimiento, pero haré todo lo posible por terminarlo en 6 meses".

"Perfecto, entonces marcamos 6 meses como objetivo", dijo Dave. "De todas formas, no podemos permitirnos más tiempo". El resto del comité estuvo de acuerdo.

*Estimación sin evaluación de requisitos ??*

En la semana 5, la **carga de trabajo adicional de los requisitos** del sistema sirvió para que Chris se diese cuenta de que los 9 meses que él había calculado era un periodo mucho más realista que los 6 meses, pero confió en que, con un poco de suerte, podría terminarlo en 7 u 8 meses. No quería que el resto del comité pensase que **no quería colaborar o que no era competente** y quería a toda costa que las cosas fuesen bien, ya que el resultado de este encargo sería decisivo para su promoción en la empresa y para los futuros trabajos.

El equipo de Chris realizó grandes progresos pero **los requisitos eran mucho más amplios de lo que habrían esperado. Cada laboratorio y departamento tenía una forma distinta de hacer las cosas** y llegar a acuerdos y a un enfoque común no fue sencillo. Habían pasado ya 3 de los 6 meses y le dijo a Dave lo siguiente: "Es imposible que podamos terminar lo que queda en 3 meses". Añadió que necesitaban una ampliación de 2 meses y reorganizarlo todo para 8. Siendo realista, creía que podría terminarlo en 9 ó 10 meses, pero estaba seguro de que no podría plantárselo al comité.

*Realizar los requisitos por lotes priorizados y no intentar hacer todos a la vez*

Semanas más tarde, Chris se dio cuenta de que **los diseños de la base de datos, de la página web y del programa iban mucho más despacio de lo programado**. "Implementad las partes más sencillas", instruyó al equipo. "Ya nos encargaremos de las partes más complicadas cuando lleguen".

Chris se reunió con el comité directivo del MIT y anunció lo siguiente: "Llevamos 7 de los 8 meses del proyecto. El **diseño detallado** casi está terminado y estamos progresando, pero

*Hacer algo físico, algo de código antes de detallar tanto el diseño. Desde una perspectiva en la que todavía no has hecho nada físico puede que haya cosas que se te estén pasando.*

*No intentar asumir responsabilidades por encima de tus posibilidades. Es mejor ser realista y sincero.*

*Este tipo de situaciones se pueden mitigar con trabajo en paralelo, puede mitigarse mejor los retrasos en una línea de progreso.*

no podremos terminarlo en 8 meses". Le solicitó a Dave una segunda ampliación a 10 meses.

Al llegar el noveno mes, el equipo había terminado el diseño detallado, pero el **desarrollo del programa aún no había comenzado** en algunas partes del sistema. Era evidente que el equipo tampoco podría cumplir la programación de 10 meses. Chris anunció una tercera ampliación a 12 meses. La cara de Dave cambió de color al escucharlo y las presiones del comité del MIT eran cada vez mayores. Chris empezó a preocuparse por su puesto de trabajo.

La creación del código iba relativamente bien, pero era preciso **reescribir el código y el diseño de algunas áreas**. El equipo no había coordinado bien los detalles de diseño de los subsistemas y algunas de las implementaciones generaban conflictos. En la reunión del comité en el mes 11, Chris pidió una cuarta ampliación del plazo, a lo que Dave, enfurecido, respondió: "¿Pero sabes lo que estás haciendo? **¡Resulta evidente que no tienes ni idea de cuándo podrás terminar el proyecto! ¡O lo terminas para el mes 12 o irás a la calle!** Tú y tu equipo trabajaréis 80 horas por semana hasta que lo hayáis terminado". Chris sintió cómo le subía la tensión, sobre todo ante el hecho de que Dave le había presionado desde el principio con una fecha de entrega inviable. Pero sabía que tras cuatro solicitudes de ampliación de plazo, su credibilidad era nula.

Chris informó a su equipo de la reunión. Trabajaron muy duro y consiguieron **entregar el software poco después del mes 13**.

*Se hacen entregas preliminares con funcionalidad reducida que permitan evaluar el producto con antelación permitiendo un margen de maniobra en vez de una sola entrega final*

**Proceso software. Enumere los problemas y los pasos clave que debería haber utilizado el equipo de desarrollo para evitarlos.**

### Problemas

- Solicitar tiempo para redilizar una previsión de tiempo realista y argumentada. Expone por qué se da el plazo de tiempo predicho. Justificar la previsión.  
Estimar utilizando técnicas.
- No comenzar a trabajar por la funcionalidad principal.
- Establecer de forma clara los límites del sistema, minimizar los requisitos a la funcionalidad básica, especialmente cuando hay una fecha de entrega.
- Falta de comunicación y coordinación, no se utiliza ninguna herramienta CASE ni ninguna de agrupación o integración del material producido como los repositorios.  
Es necesaria la integración continua.
- Falta de metodología y herramientas.
- El ciclo de vida en cascada como el aplicado no es un ciclo de vida adecuado para un proyecto como este en el que los requisitos no estaban claros.  
Un ciclo de vida en espiral hubiera sido más adecuado, se podría haber empezado a trabajar con los requisitos más claros de modo que al final la primera espiral se tuviera otro lote de requisitos claros con los que empezar a trabajar. Esto hubiera mitigado los retrasos también, existe una reducción de carga de trabajo por lotes.  
Se pueden hacer pruebas y control al final de cada espiral.  
— La decisión de cómo trabajar influye en el resultado.

— Lo importante no es hacer diagramas si no lo que se cuenta en ellos.

*En realizando desarrollo de forma simultánea a las entregas para obtener feedback del usuario*

*???  
Falta planificación y gestión*

*Excesiva presión en el ambiente de trabajo.*