

INTRODUCCIÓN

En el ámbito del desarrollo de software, la ingeniería de requisitos juega un papel crucial para asegurar que los proyectos cumplan con las expectativas y necesidades tanto de los clientes como de los desarrolladores. Este documento recoge una **entrevista realizada a Francisco Trujillo**, ingeniero de software en **TUI**, una empresa multinacional líder en el sector turístico, sobre las prácticas, retos y metodologías empleadas en su ciclo de desarrollo. A lo largo de esta entrevista, se exploran aspectos clave como la captura, análisis y gestión de requisitos, además de las particularidades de trabajar en un entorno que integra servicios tradicionales y digitales en múltiples mercados a nivel mundial.

El formato a seguir en la sección de preguntas y respuestas comprenderá la pregunta, que en la entrevista en bruto siempre ha sido realizada por el entrevistador, y la respuesta dada, que es una síntesis de la respuesta literal que dio el entrevistado.

Además de presentar las respuestas del entrevistado, se llevará a cabo un análisis crítico de las estrategias y soluciones que Fran plantea en relación con la ingeniería de requisitos, evaluando su efectividad dentro del contexto empresarial y sus posibles áreas de mejora.

PREGUNTAS Y RESPUESTAS

INTRODUCCIÓN: romper el hielo.

- 1. Presentación** — ¿Podrías presentarte y describir tu rol dentro de TUI? ¿Qué responsabilidades tienes dentro de los proyectos de desarrollo de software?

Mi nombre es Fran, llevo unos siete años trabajando como desarrollador de software, y en TUI aproximadamente cinco años. Mi principal responsabilidad es mantener varias aplicaciones en funcionamiento, lo que implica desarrollar nuevas funcionalidades y solucionar cualquier problema que surja, en colaboración con otros equipos cuando es necesario. Básicamente, mantener la aplicación viva. En concreto, mi rol se centra en backend el 90% del tiempo, aunque también toco algo de sistemas y frontend.

- 2. Importancia** — ¿Qué **importancia** tiene la ingeniería de requisitos en vuestra metodología de desarrollo? ¿Cómo encaja dentro del ciclo de vida de desarrollo de software en la empresa?

La ingeniería de requisitos es clave. Si los requisitos no están claros a la hora de desarrollar una nueva funcionalidad o un nuevo producto, al final eso va a causar malentendidos entre ambas partes: stakeholders y los desarrolladores. En consecuencia, durante el desarrollo, pueden surgir nuevos requisitos, lo que puede impactar significativamente en el proyecto y aumentar costos y tiempos.

- 3. Metodología de desarrollo** — ¿Cómo gestiona TUI el **ciclo de vida** del desarrollo de software? ¿Utilizan **metodologías ágiles, iterativas** o algún otro enfoque? ¿Qué es lo **mejor** de esa metodología con respecto a las demás?

En nuestra empresa se utilizan diferentes metodologías como Scrum o Kanban. Pero dentro de nuestro equipo utilizamos Kanban, ya que es el que mejor se ajusta a nuestras necesidades. Llevamos muchas aplicaciones a la vez, esto provoca cambios de prioridades prácticamente diarios que, en caso de usar Scrum, nos dificultaría aún más nuestro trabajo. Kanban es mucho más flexible ya que se trabaja con tickets, es decir, con funcionalidades. Esto es muy útil, ya que nos permite movernos entre los diferentes tickets sin necesidad de quedarnos trabajando en uno un tiempo determinado.

En resumidas cuentas, tratamos de refinar esos tickets lo máximo posible para evitar problemas durante el desarrollo. El Product Owner, recopila los requisitos en función del negocio, nos reunimos con él y vemos cual sería la mejor aproximación al problema. Si tenemos más dudas cuando estamos refinando el ticket, lo hablamos directamente con los stakeholders.

- 4. Visión general del proceso** — ¿Cómo ha **evolucionado** el proceso de ingeniería de requisitos en la empresa en los últimos años? ¿Qué **mejoras y problemas** principales habéis experimentado?

En los cinco años que llevo trabajando en la empresa siempre hemos seguido la misma metodología, no hemos cambiado nada.

En cuanto a los problemas, uno de los más recurrentes es que a veces los stakeholders no proporcionan la funcionalidad de los tickets de forma clara y completa. Esto implica que a menudo tengamos que devolver el ticket a los stakeholders, reunirnos con ellos y refinar el ticket antes de empezar el desarrollo.

ELICITACIÓN: recopilar requisitos.

- 5. Captura de requisitos** — ¿Cómo se **capturan los requisitos** en los proyectos de software en vuestra empresa? ¿Utilizáis **entrevistas, reuniones, prototipos, cuestionarios** o alguna combinación de estas? ¿Cuáles técnicas piensas que es **mejor**?

Lo que utilizamos principalmente son entrevistas. Dependiendo del proyecto, el Product Owner se involucra directamente con los stakeholders, o bien los desarrolladores nos reunimos con ellos. Creo que la reunión es la mejor técnica de todas ellas. La comunicación es directa y puedes hacer las preguntas directamente a los stakeholders para solucionarlas al momento.

- 6. Requisitos escalables y mercados** — Con más de 28 millones de clientes a nivel mundial, ¿cómo abordáis la necesidad de que los requisitos sean **escalables y aplicables en diferentes mercados y geografías**? ¿Tenéis algún enfoque específico para **gestionar estas variaciones**?

Cada país tiene su propio equipo y su infraestructura específica, por lo que no necesitamos desarrollar una única aplicación global para todos los países. Cada equipo se encarga de asegurarse de que su aplicación cumpla con las normativas y requerimientos locales de su país. Esto nos permite enfocarnos en las necesidades particulares de cada mercado sin sobrecargar el sistema global.

- 7. Integración de sistemas tradicionales** — TUI gestiona una vasta red de hoteles, cruceros, agencias de viajes y aerolíneas. ¿Cómo manejáis los requisitos cuando se trata de **integrar sistemas digitales con estos servicios tradicionales**? ¿Qué **particularidades** encontráis **al trabajar en el sector turístico**?

Particularidades en sí... No veo que haya diferencia con respecto a otros sectores. Trabajé en una empresa de telecomunicaciones, y no había diferencia en la forma de trabajar.

Por otro lado, aunque el turismo se desarrolla prácticamente al 100% de forma digital, curiosamente, en TUI trabajamos con agencias de viajes tradicionales. En nuestro caso, nos encargamos de desarrollar las herramientas internas que utilizan las aplicaciones de las agencias de viajes en Bélgica. Estamos sincronizados con los servicios tradicionales por así decirlo.

- 8. Normativas locales** — Dado que operáis en múltiples países, ¿cómo aseguráis que los requisitos del software **se ajusten a las diferentes normativas y regulaciones locales** de cada región (por ejemplo, protección de datos o normativas de transporte)?

Cuando hay empresas grandes, normalmente hay un equipo legal que son los que se encargan de asegurarse de que se cumplen las normativas y las regulaciones del país, y si hay algún cambio en la regulación que provoque cambios dentro del software, crean un nuevo ticket donde explican cuáles son esos cambios y nosotros lo aplicamos en el software. Y si hablamos de temas de seguridad, nosotros, por ejemplo, tenemos auditorías cada seis meses de código donde se encargan de confirmar que las aplicaciones son seguras y no son vulnerables a ataques.

- 9. Implicación de partes interesadas** — ¿Cómo se **involucra a los stakeholders** (partes interesadas) a la hora de recopilar requisitos para el proyecto para asegurar que los requisitos se ajusten a sus expectativas? ¿Hay diferencias entre la implicación de los clientes internos y externos?

Ya hemos hablado anteriormente de cómo se involucran los stakeholders a través de las entrevistas. Además de la toma de requisitos, posteriormente al desarrollo, los stakeholders suelen hacer un testing antes de desplegar el software al cliente.

En cuanto a la diferencia entre clientes internos y externos, nosotros no trabajamos con aplicaciones externas, son todas aplicaciones internas dentro de la empresa. Es decir, no tenemos una aplicación que vaya a usar el cliente directamente. Son herramientas que utilizan agencias de viaje o conexiones con la web. Si alguien hace un booking o compra un vuelo desde la web directamente, se conecta con nosotros.

ANÁLISIS: consideraciones posteriores a la captura, antes de abordar el desarrollo.

10. Conflictos de requisitos — ¿Qué métodos utilizáis para gestionar **conflictos entre diferentes partes interesadas** cuando existen diferencias sobre lo que consideran un requisito esencial?

Nos reunimos con ellos e intentamos llegar a un punto medio. En ocasiones, alguna de esas partes te pide una funcionalidad que requiere demasiado tiempo o no es viable. En ese caso, tratamos de aportar otra solución que nos permita llegar a un punto intermedio. Por así decirlo, “negociamos” con ellos.

11. Análisis de impacto y gestión de riesgos — ¿Realizáis algún tipo de **análisis de impacto antes de implementar cambios en los requisitos**? ¿Cómo se evalúan los **riesgos** asociados a los requisitos críticos o técnicamente desafiantes?

Este análisis se produce durante la fase de refinamiento. Antes de la entrevista, nos reunimos los desarrolladores para identificar los posibles riesgos y soluciones que pueden conllevar las funcionalidades del software. Si hay riesgos que no se pueden solucionar, tratamos de dar posibles soluciones alternativas durante la entrevista e intentamos negociar con los clientes un punto intermedio que no suponga un cambio a largo plazo en la aplicación.

ESPECIFICACIÓN: documentar requisitos de manera clara y precisa.

12. Requisitos ambiguos y comprensibilidad — ¿Cómo gestionáis los **requisitos ambiguos** o que puedan interpretarse de múltiples formas? ¿Qué métodos utilizáis para **asegurar que los requisitos son comprensibles tanto para desarrolladores como clientes**?

Si detectamos ambigüedades en los requisitos, nos reunimos directamente con los stakeholders para aclarar cualquier duda antes de comenzar el desarrollo. Al final la comunicación directa con los stakeholders creo que es clave: es la manera más sencilla de solucionar los problemas.

13. Control de calidad y herramientas — ¿Cómo se asegura la **calidad y completitud** de los requisitos antes de empezar el desarrollo? ¿Qué proceso o herramientas usáis para revisar, refinar y terminar de definir un requisito antes de la fase de desarrollo?

El Product Owner actúa como el primer filtro para asegurarse de que los requisitos son claros y completos. Básicamente, hace que el lenguaje de negocio sea claro. Si aún así persisten dudas, nos reunimos con los stakeholders para refinarlos antes de iniciar el desarrollo. Casi todas las respuestas son iguales: entrevista o reunión, este proceso es el que al final garantiza que los requisitos estén alineados con las expectativas.

VALIDACIÓN Y VERIFICACIÓN: comprobar si los requisitos son los que los stakeholders quieren.

14. Validación y prototipado — ¿Qué técnicas utilizáis para **validar y verificar los requisitos** con las partes interesadas, especialmente en **proyectos que incluyen requisitos complejos o innovadores**? ¿Utilizáis técnicas de **prototipado** u otro tipo de técnicas más específicas?

Ya respondimos antes, para validar y verificar los requisitos: reunión.

Tenemos la lista de las funcionalidades mínimas que necesita el cliente y vamos trabajando con iteraciones. Primero desarrollamos lo mínimo necesario y se lo enseñamos al cliente. Si da el visto bueno, seguimos trabajando en una nueva iteración buscando de nuevo el producto mínimo viable. De esta forma, aseguramos que el proyecto avanza en la dirección correcta.

15. Determinación del éxito — ¿Cómo determináis que un proyecto ha sido **"exitoso"**? (cómo miden que el software tenga calidad y se cumplan los requisitos)

Consideramos que un proyecto ha sido exitoso cuando se despliega en producción, los clientes lo están utilizando y no hay problemas. Además, los stakeholders prueban la funcionalidad tanto en el entorno de testing como en el de producción para asegurarse de que cumple con las expectativas.

GESTIÓN (de cambios): adecuar los requisitos a los cambios del proyecto.

16. Cumplimiento de requisitos no funcionales — ¿Cómo se asegura el **cumplimiento de requisitos no funcionales** a lo largo del desarrollo?

Como trabajamos con cargas de datos muy grandes, normalmente hacemos test de carga antes de desplegarlo, para asegurarnos que el software es capaz de soportar cierto volumen de solicitudes. En cuanto a la escalabilidad, no tenemos en sí porque trabajamos con servidores físicos, entonces no tenemos la posibilidad de escalarlo, no es como trabajar en la nube. Para asegurar la calidad del código, tenemos las Code Review para asegurar que el código que se está desplegando tiene buena calidad. Dependiendo de la aplicación que sea, tenemos tests unitarios o tests funcionales.

17. Priorización de requisitos — Cuando tenéis que **priorizar requisitos**, ¿cómo afectan las **diferencias entre los distintos servicios turísticos** en esas decisiones.

La priorización de los requisitos es responsabilidad del Product Owner. Ellos determinan qué funcionalidades son más importantes para el negocio en cada momento. Como desarrolladores, seguimos sus indicaciones.

18. Cambios en los requisitos y áreas de negocio — ¿Cómo gestionáis los **cambios en los requisitos**, especialmente en **proyectos que requieren la coordinación de múltiples áreas** de negocio como hoteles, cruceros y aerolíneas?

Si hay un cambio de requisitos durante el desarrollo, analizamos qué es lo que necesitamos cambiar dentro de la aplicación y aplicamos los cambios. No hay mucha magia... Además, no es necesaria la coordinación entre las diferentes áreas ya que como he mencionado antes, no hay una aplicación global como tal.

19. Herramientas de gestión de requisitos — ¿Qué **herramientas** usáis para la gestión de requisitos (con esto me refiero a organizarlos por prioridad, adaptarlos, etc)? ¿Tenéis algo especial que creas que os ha podido hacer **sobresalir** sobre otras empresas?

Especial que haga sobresalir sobre otras empresas, no. Para la toma de requisitos utilizamos Jira, que es el estándar del sector. Es muy similar a Trello pero con más funcionalidades, permite organizar los proyectos con un board, donde las tareas se mueven a través de las diferentes columnas como “Por hacer”, “En desarrollo”, “En revisión”...

20. Trazabilidad de requisitos — ¿Cómo gestionáis la **trazabilidad** de los requisitos a lo largo del ciclo de vida del proyecto?

Utilizamos herramientas como Jira para gestionar la trazabilidad, lo que nos permite seguir cada requisito desde su definición hasta el despliegue en producción. Los requisitos se refinan y ajustan a medida que avanzamos, y si hay cambios, se aplican en la siguiente iteración o fase, dependiendo de su prioridad. Cambios estéticos, por ejemplo, se posponen. La trazabilidad y la adaptabilidad están directamente relacionadas con las reuniones, donde discutimos el impacto de los cambios para asegurar que todo esté alineado.

21. Evolución de requisitos y equipo de desarrollo — A medida que los proyectos avanzan, ¿cómo se **adapta el equipo de desarrollo** a la evolución de los requisitos? ¿Tenéis algún mecanismo formal para **ajustar el alcance del proyecto sin comprometer las fechas de entrega**?

Nosotros realmente no tenemos casi fechas límite estrictas como tal, gozamos de una alta flexibilidad. En las raras ocasiones en las que las tenemos, si se añade algún requisito nuevo o cambio durante el desarrollo, priorizamos los requisitos necesarios y eliminamos funcionalidades estéticas o que no sean tan importantes, para ajustarnos a la fecha de entrega. Obviamente, una vez más, antes de eliminar alguna funcionalidad, lo hablamos con los stakeholders y juntos decidimos.

CONSIDERACIONES ADICIONALES: preguntas relacionadas con las fases de IR, pero que no caen en ninguna en específico.

22. Herramientas usadas en el día a día — ¿Qué **herramientas** utilizas en tu **día a día** como desarrollador de TUI (principalmente enfocado a la ingeniería de requisitos, aunque también interesa en general)?

En mi día a día utilizo varias herramientas. Para el desarrollo de software, trabajo con PHPStorm, que es un IDE especializado para PHP, y también uso IntelliJ y GoLand. Para la base de datos DataGrip. Como puedes observar, nos gusta bastante JetBrains.

Para la gestión de requisitos utilizamos Jira como ya he comentado anteriormente. Y por último usamos GitLab para el control de versiones de software. Es muy similar a GitHub pero interno de la empresa. Con lo relacionado al pipeline de despliegue y similares utilizamos Jenkins y GitLab Pipelines, que viene a ser una funcionalidad de GitLab.

23. Seguridad de datos — Teniendo en vuestras manos los **datos** de tantos usuarios (al ser un grupo turístico), ¿cómo **aseguráis y gestionáis la seguridad** en el desarrollo del software? ¿Cómo **protegéis** los datos de los clientes?

Un equipo de seguridad dentro de la empresa se encarga internamente, aparte de la auditoría externa, de comprobar las brechas de seguridad de las aplicaciones. Hay otro equipo de IT que se encarga de que los servidores sean seguros, y además nosotros contamos con ciertas medidas de anonimización de datos en las bases de datos.

24. Sostenibilidad — La **sostenibilidad** es clave en la cultura de TUI. ¿Cómo se traducen estos **valores en requisitos no funcionales** dentro del desarrollo de software? ¿Consideráis aspectos como **eficiencia energética** o **reducción de la huella de carbono** en el desarrollo de vuestros sistemas?

Es algo que, sinceramente, no abordamos en nuestro día a día como desarrolladores. Pero ciertamente, es un valor importante para la empresa, y es probable que los aspectos de sostenibilidad estén integrados en decisiones estratégicas sobre la infraestructura y diseño de los sistemas.

25. IAs generativas — Dada la creciente popularidad de las **IAs generativas** y su aumento en capacidades, ¿habéis implementado u os planteáis implementar el uso de herramientas como ChatGPT en alguna fase de la ingeniería de requisitos?

Sí, la empresa nos provee cuenta de Copilot de empresa, el cual es privado, es decir, Microsoft supuestamente no puede utilizar los datos para entrenar la IA. La utilizamos para el desarrollo del software y es muy cómoda. Por ejemplo, la semana pasada, lo estuve utilizando para mapear un XML en código y me ahorró a lo mejor tres o cuatro horas de trabajo en cinco minutos. Pero para la ingeniería de requisitos no la utilizamos, ChatGPT no puede hacer un meeting, todavía...

26. Integración de sistemas de terceros — ¿Cómo gestionáis la **integración de sistemas de terceros**, como sistemas de reservas aéreas o plataformas de pagos, dentro de vuestro ecosistema digital?

La comunicación con sistemas de terceros normalmente se realiza mediante APIs, pero en nuestro equipo, son todo comunicaciones internas por lo que no trabajamos con terceros, solamente con la aerolínea Lufthansa, y utilizamos API. En cuanto a plataformas de pago, es un sistema legacy que utiliza servidores FTP, pero es algo que se está intentando modernizar.

27. Gestión del feedback y mantenimiento — ¿Cómo **gestionáis** el *feedback* de los clientes después de entregar el producto? ¿Realizáis un **mantenimiento continuo** de los mismos?

Mantenimiento siempre. Llegan nuevas funcionalidades que tenemos que implementar y las llevamos a cabo. Una vez que la aplicación está en despliegue, nosotros somos dueños de la misma, por tanto, es nuestra labor asegurar que la aplicación sigue funcionando correctamente. Si hay feedback positivo, todo correcto, en el caso de que haya feedback negativo, pues se crean nuevos tickets o funcionalidades para solucionar los problemas que ha tenido el cliente.

CIERRE: cierre de la entrevista.

28. Duda adicional final — Si una reunión no se puede llevar a cabo a corto plazo ya que el stakeholder no encuentra disponibilidad para llevarla a cabo, ¿tenéis que dejar el trabajo parado o podéis hacer algo?

Si el problema bloquea el resto del desarrollo de la aplicación, esa funcionalidad se bloquea hasta que el stakeholder solucione el problema o duda. Si el problema es bloqueante, se bloquea el ticket hasta que ellos lo solucionan. Esto me ha ocurrido muchas veces, y lo que hacemos es informar del bloqueo, bloquear el ticket, ponerle un flag, coger el siguiente ticket de la lista de prioridades y empezar a trabajar en él hasta que se solucione lo que te está bloqueando. Siempre tenemos un nuevo ticket que coger, tenemos muchísimo trabajo.

29. Cierre — Teniendo en cuenta que ya hemos terminado, ¿hay algún, **aspecto o práctica** que no hayamos abordado que creas importante y debamos mencionar?

Yo creo que está claro todo, hemos cubierto todo lo esencial en cuanto al proceso de ingeniería de requisitos.

ANÁLISIS POSTERIOR

La entrevista a Fran, ingeniero de software en TUI, revela varias prácticas sólidas en cuanto a la ingeniería de requisitos dentro de la empresa, pero también identifica algunas **áreas que podrían beneficiarse** de mejoras.

Puntos fuertes

Una de las principales fortalezas que emerge es la clara **comprensión de la importancia de la ingeniería de requisitos** en el desarrollo de software. Fran destaca que, si los requisitos no se definen correctamente desde el principio, es probable que surjan problemas que afecten tanto el costo como el tiempo del proyecto. Esta conciencia de la necesidad de claridad y alineación entre stakeholders y desarrolladores es crucial para el éxito de cualquier proyecto de software, ya que ayuda a prevenir retrasos y malentendidos costosos.

Otra fortaleza significativa es el uso de **metodologías ágiles**, específicamente **Kanban**, para gestionar el ciclo de vida del desarrollo. TUI ha optado por Kanban en lugar de Scrum debido a la **flexibilidad** que ofrece, lo que resulta particularmente útil en un entorno en el que las prioridades pueden cambiar de forma rápida y constante. La flexibilidad inherente a Kanban permite que el equipo de desarrollo ajuste sus tareas diarias de acuerdo con las necesidades más inmediatas de la empresa, lo cual es una **ventaja clave para la productividad** y la capacidad de respuesta. Este enfoque permite a TUI ser más ágil en su respuesta a las demandas del mercado y de los clientes, mejorando la eficiencia general del equipo.

Fran hace un especial énfasis en la **importancia de la comunicación con el cliente** y destaca lo crucial de que está sea directa y sin rodeos, a través de reuniones y entrevistas. Esto es clave tanto para la **solución de problemas y riesgos** antes del desarrollo, durante del desarrollo y después de este. Antes para asegurar que los requisitos están claros y completamente refinados para asegurar un desarrollo correcto; durante, para comprobar que el producto sigue el camino esperado y solucionar las posibles dudas y cambios que surjan y después, para confirmar que el producto funciona correctamente y el testeado es fructífero.

Puntos débiles

Uno de los aspectos en los que TUI podría mejorar es la **formalización del proceso de captura de requisitos**. Aunque Fran menciona que las reuniones con los stakeholders y el Product Owner son claves para asegurar la claridad de los requisitos, la falta de un proceso formal puede llevar a **ambigüedades o malentendidos**. En organizaciones grandes y distribuidas como TUI, donde los requisitos pueden cambiar frecuentemente, no tener una estructura rigurosa y documentada puede retrasar el desarrollo y causar confusiones. Un enfoque más formal ayudaría a reducir la necesidad de volver a aclarar los requisitos constantemente.

Además, la **falta de escalabilidad** en los sistemas de TUI es otro punto a considerar. Fran señala que trabajan con servidores físicos y que no tienen un sistema fácilmente escalable. En un entorno global y en un sector tan competitivo como el turismo, esta limitación puede suponer una desventaja. Migrar hacia soluciones más flexibles, como el uso de infraestructura en la nube, podría ofrecer la **agilidad y capacidad de respuesta** necesarias para manejar grandes volúmenes de datos y picos de demanda.

Finalmente, otra área de mejora es la **gestión de conflictos entre partes interesadas**. Aunque se menciona que se negocia con los stakeholders cuando surgen conflictos, un enfoque más sistemático para resolver estas discrepancias podría agilizar el proceso. Contar con una metodología estructurada para gestionar conflictos y tomar decisiones podría facilitar la alineación entre todas las partes implicadas, evitando retrasos en la toma de decisiones clave.

Cómo mejorar

Para mejorar sus prácticas de ingeniería de requisitos, TUI podría comenzar por **formalizar el proceso de captura de requisitos**. Establecer un sistema más estructurado ayudaría a documentar de manera clara y consistente, lo que a su vez reduciría ambigüedades y facilitaría el seguimiento de los requisitos a lo largo del desarrollo.

La migración a soluciones basadas en la **nube** también podría ser un paso positivo. Esto no solo mejoraría la escalabilidad de los sistemas, sino que también permitiría a la empresa adaptarse con más agilidad a los picos de demanda, algo esencial en el competitivo sector del turismo.

En cuanto a la **gestión de conflictos** entre stakeholders, un enfoque más sistemático podría agilizar la toma de decisiones. Implementar técnicas de priorización, como MoSCoW, permitiría al equipo centrarse en lo que realmente importa para el negocio, asegurando que todos estén alineados.

Por último, **automatizar el análisis de impacto** con herramientas que rastreen las dependencias entre requisitos y funcionalidades podría ser de gran ayuda. Esto facilitaría la adaptación a los cambios y garantizaría que el equipo mantenga la claridad en las expectativas de los stakeholders.