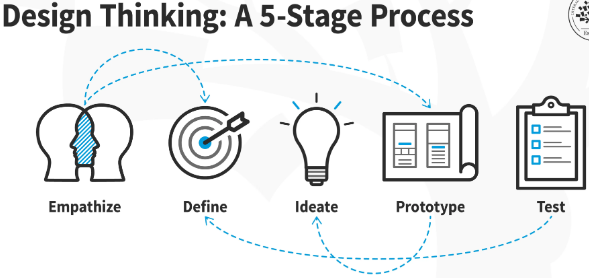
**Design Thinking**

El Design Thinking es un proceso iterativo no lineal que los equipos utilizan para entender a los usuarios, desafiar las suposiciones, redefinir problemas y crear soluciones innovadoras para prototipar y testear. Involucra 5 fases: **empatizar**, **definir**, **idealizar**, **prototipar** y **testear**. Es muy útil para abordar problemas mal definidos o desconocidos. 

¿Por qué es tan importante?

En UX (user experience) es crucial desarrollar las habilidades para entender y abordar los rápidos cambios en el entorno y comportamientos de los usuarios.

Los equipos de diseño utilizan el Design Thinking para abordar problemas mal definidos/desconocidos (conocidos como wicked problems) porque pueden replantearlos de manera centrada en el humano y centrándose en lo más importante para los usuarios. De todos los procesos de diseño, Design Thinking es el mejor para “pensar por fuera de la caja”. Con esto, los equipos pueden mejorar en cuanto a investigación de UX, prototipado y testing de usabilidad.

Con Design Thinking los equipos tienen la libertad para generar soluciones innovadoras. Al utilizarlo, son capaces de conseguir perspectivas difíciles de obtener y aplicar una variedad de métodos hands-on.

5 etapas de Design Thinking

Notar que estas etapas no son secuenciales, los equipos a veces las ejecutan en paralelo, fuera de orden y repitiendolas de forma iterativa.

Etapa 1: **Empatizar** (Investigar las necesidades de usuario)

Aquí se debe comprender de forma empática el problema que se está tratando de resolver. Generalmente a través de la investigación de los usuarios. La empatía es crucial para los procesos de diseño centrados en el humano, como Design Thinking, ya que permite apartar las suposiciones propias sobre el problema y obtener una mejor perspectiva sobre los usuarios y sus necesidades.

Etapa 2: **Definir** (Definir las necesidades de usuario y problemas)

Se trata de acumular la información recolectada durante la etapa 1. Se analizan y se sintetizan las observaciones para identificar los problemas principales. Estas definiciones son llamadas enunciados de problemas. Se pueden crear *personas* para ayudar a mantener el proceso centrado en el humano.

Etapa 3: **Idear** (Desafiar las suposiciones y crear ideas)

Estamos listos para crear ideas. La información generada en las anteriores fases permite “pensar por fuera de la caja”, buscar formas alternativas de observar el problema e identificar soluciones innovadoras para el enunciado del problema creado anteriormente.

El brainstorming es particularmente útil en esta etapa.

Etapa 4: **Prototipar** (Crear soluciones)

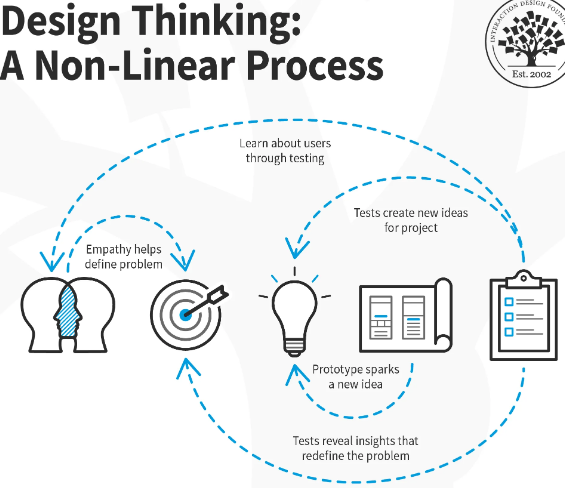
Esta es una fase para experimentar. El objetivo es identificar la mejor solución posible para cada problema. Se deben producir versiones del producto baratas, de baja escala para identificar las ideas generadas.

Se pueden utilizar prototipos de papel simple.

Etapa 5: **Test** (Prueba tus soluciones)

Se testea rigurosamente el prototipo. A pesar de ser la etapa final, Design Thinking es iterativo. Los equipos suelen usar los resultados para redefinir los problemas.

Estas etapas son diferentes modos de contribuir al proyecto de diseño en vez de pasos secuenciales. El objetivo es ganar un mayor entendimiento de los usuarios y cuál es su solución/producto ideal.



**BrainStorming**

Es la metodología más popular de generación de ideas. Su objetivo principal es encontrar el mayor número de ideas diferentes y relevantes posibles con las que dar respuesta a un reto (en equipo es la mejor forma, aunque también puede ser individual).

Ventajas adicionales:

* Potenciar una cultura ágil y de innovación en la empresa.
* Aumentar la creatividad en las personas que lo practican.

Además puede ser un punto de partida a la hora de empezar un proyecto. Practicarlo ayuda a:

* Desarrollar creatividad
* Generar más ideas
* Atreverse a pensar de forma diferente.

11 Claves para hacer un brainstorming:

* Hacerlo en grupo (crecen las ideas cuando se comparte, mejor si es entre 4 o 6)
* Buscar siempre cantidad (no juzgues las ideas que se te ocurran)
* Aprovechar las ideas que dicen los demás para generar otras nuevas.
* Anotar y enumerar (fijar una meta de ideas, 30 como mínimo)
* Tener un facilitador (un rol de guía de brainstorming, dice los tiempos y fomenta la generación de ideas)
* Una persona encargada de anotar ideas (y diga las suyas, puede ser el facilitador pero sin anotar sus ideas)
* Establecer tiempos
* Elegir un espacio cómodo.
* Ideas claras y autoexplicativas con sujeto y predicado.
* Incluir estímulos creativos (SCRAMPER, Hibridación o Brainwriting)
* Ideas locas e inesperadas.

Fases

**Fase 1:** Diseña la sesión de brainstorming

El diseño es importante, contar con un espacio inspirador para hacer brainstorming. Que el espacio cuente con luz, sillas cómodas, ambiente distendido, espacio para poder pintar, lugar suficientemente grande, estímulos visuales inspiradores.

Elegir el facilitador es fundamental, ayuda a preparar la sesión, dinamizarla y potenciar las posibilidades del grupo, lo habitual es que sea externo.

Labores del facilitador:

* Preparar el espacio (contar con flipchart o superficies ampliar, rotuladores)
* Explicar las reglas del juego antes.
* Dinamizar la sesión, abrir perspectivas y animar.
* Al final de la sesión, recoger conclusiones e ideas.

**Fase 2:** Desarrolla la sesión de brainstorming

Antes de empezar se debe calentar creativamente: escribir en un minuto el mayor número posible de usos de un elemento. Por ejemplo, una zapatilla, una silla. Y después hay que pedir que compartan uno de sus usos. Después de calentar, todo el mundo se sentirá más enchufado.

Inicio de la sesión: el facilitador dará los objetivos de la sesión y las pautas fundamentales del proceso (la única idea mala es la que no se dice). El tiempo recomendado es de 25 a 35 minutos, al menos 30 ideas.

Puede ocurrir estancamiento, el facilitador debe recurrir a estímulos como pedir que hablen pensando en otra persona o pidiendo que digan alguna idea absurda.

Luego de terminar el brainstorming hay que elegir las mejores ideas. Lo recomendable es 4 votos para cada participante. Tras la primera votación el facilitador subraya las ideas más votadas, luego los participantes tendrán un único voto, las que pasarán a ser las ideas seleccionadas.

**Fase 3:** Conclusiones y siguientes pasos

Es importante que existan momentos de reflexión: espacios de oportunidad en los que revisar el proceso vivido y aprender para hacerlo mejor la próxima vez.

En caso de que el brainstorming se haya llevado en un proceso de Design Thinking, el siguiente paso es prototipar (tomar las/la idea elegida y con ellas aterrizar la solución del reto). Por último se documenta el proceso. Sacar fotos a los paneles con todas las ideas, recoger reflexiones.

El Brainstorming en Design Thinking forma parte de la Ideación (tercera fase).

**Product Use Case Scenarios**

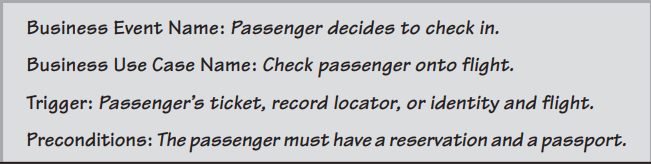
Los escenarios de product use case (PUC) son usados para comunicar la intención del producto automatizado a los stakeholders. Habitualmente no es el único documento disponible, pero aunque lo fuera, a lo que muestra las funcionalidades del producto, es un documento sencillo con el cual transmitir las intenciones a los stakeholders. Es recomendable presentar el escenario de PUC en una reunión adecuada, para obtener un feedback.

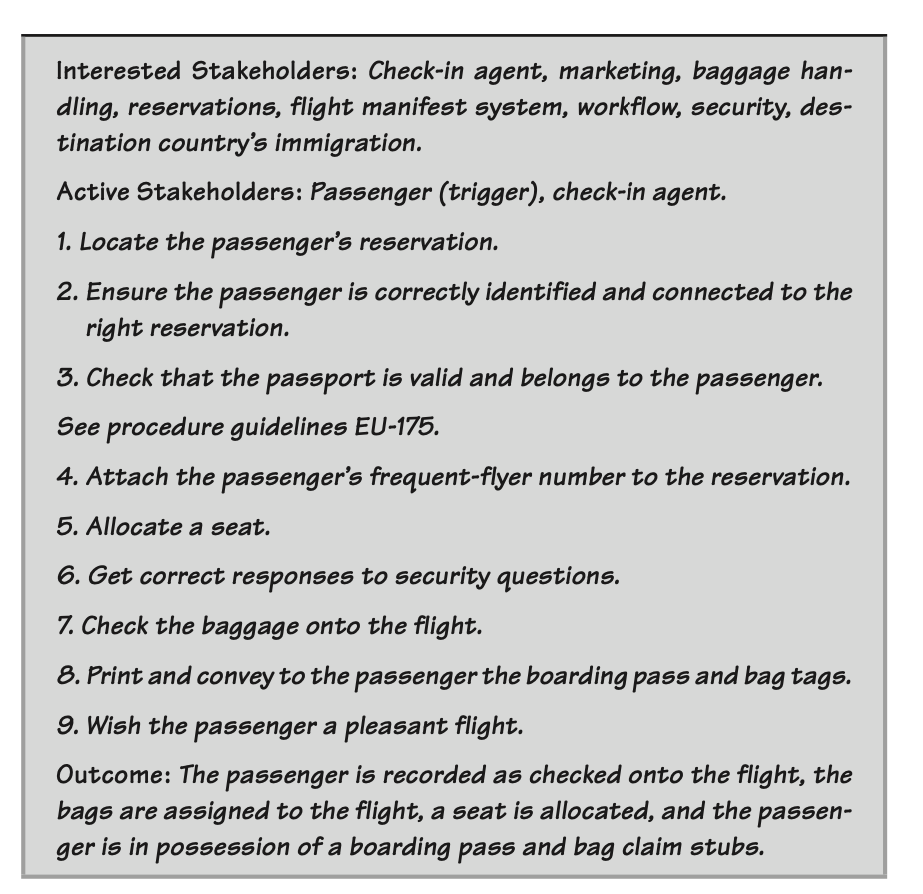
Estos son una técnica para resolver el mayor problema de la especificación de requisitos, ya que los últimos son difíciles o imposibles de leer y entender.

Un escenario de PUC es similar a un escenario de BUC, con la diferencia de que el BUC contiene toda la funcionalidad que corresponde al BE, mientras que el PUC correspondiente contiene únicamente la funcionalidad a ser implementada en el producto.

El primer paso es identificar los BE, luego se debe seleccionar uno para recabar información hasta descubrir la funcionalidad que responde a ese evento (BUC). Luego se escribe un escenario de BUC para este evento. Cuando los stakeholders están felices con el escenario, determina qué tanto del BUC puede ser implementado en el producto. El resultado de esto termina siendo el PUC, se sugiere describir un escenario para este PUC.

Ejemplo de este proceso:

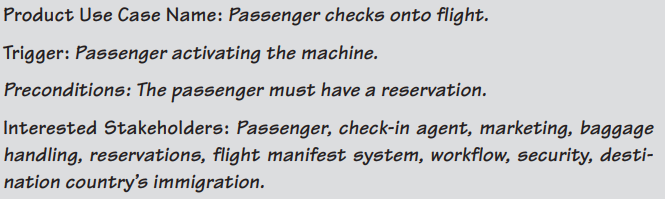


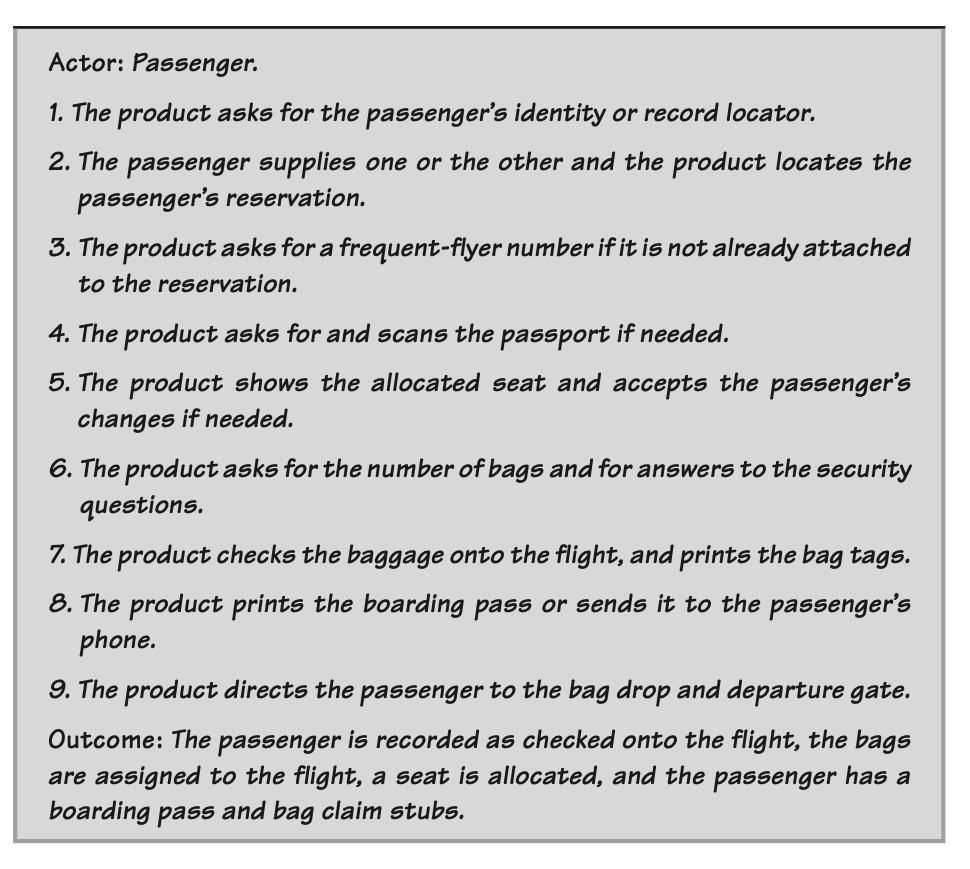


Para tomar decisiones sobre los límites del producto este BUC se definen las restricciones.

También se necesita un input de los stakeholders, quienes entienden las implicaciones técnicas y de negocio, y las posibilidades del límite del producto junto con los objetivos del negocio para este proyecto.

El escenario del PUC muestra lo que se pretende hacer:





Al ser presentado a los stakeholders probablemente se deban hacer revisiones del escenario, pero una vez presentado debería de ser una representación precisa del producto a crear.

**Evitar Ambigüedad**

Los requisitos pueden tener mucha ambigüedad y malentendidos, más cuando no se entiende muy bien el modelo de negocio. Para esto se recomienda que en grupo se escriba un PUC a la vez, para poder discutir al respecto. Si bien cualquier cosa puede ser ambigua, el escenario PUC puede reducir la ambigüedad.

Se recomienda eliminar todos los pronombres del requisito y reemplazar por un sujeto u objeto.

Cuando se escribe un requisito, decirlo en voz alta, si es posible, tener un compañero que lea en voz alta. Confirmar que el stakeholder alcance el mismo significado.

**Requisitos tecnológicos**

Los requisitos tecnológicos son necesarios para especificar la tecnología elegida. Es necesario que se escriban diferenciados a los demás requisitos. Es importante que aparezcan después de entender los business requirements.

**Grupos de requisitos**

Recomendamos que se agrupen los requisitos por BUC. La ventaja es que es fácil describir grupos de requisitos relacionados y probar su integridad.

Agrupar los requisitos por características es más fácil para manipularlas. También tiene sentido agrupar por PUC.

Hay jerarquía por estas razones:

* Para poder involucrar a las partes interesadas con diferentes enfoques.
* Para ayudar a descubrir los requerimientos atómicos.
* Para ser capaz de negociar entre volumen y complejidad.

**Alternativas a los requerimientos funcionales**

**Escenarios**

Se sugiere escribir escenarios de PUC, se puede simplemente agregar detalles de especificación al escenario y usarlo como especificación. Si el escenario se vuelve demasiado largo o elaborado se debe escribir los requisitos funcionales de manera normal.

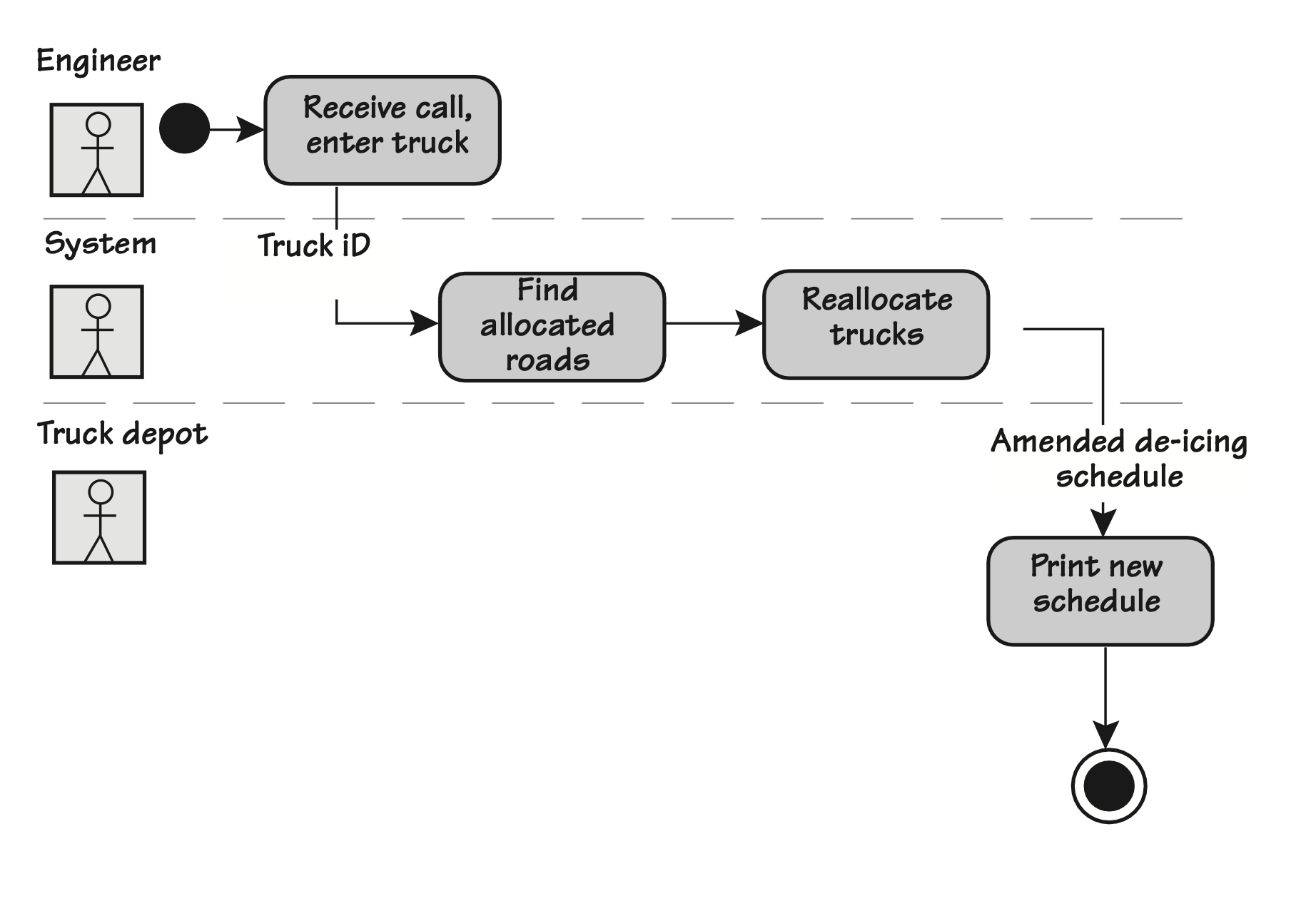
**User Stories**

Son otra forma de describir las funcionalidades del producto. Se debe considerar como una alternativa a los requerimientos funcionales. Son escritos así: Yo como [rol]. Quiero [funcionalidad] porque [razón].

Las user stories son escritas usualmente por el representante del cliente en story cards. Las story cards son usadas como punto de referencia para armar los requerimientos. Estos se desarrollan con conversaciones entre los desarrolladores y los stakeholders.

**Modelo de proceso de negocio**

Sí como práctica habitual construyes diagramas, considera si estos junto a las descripciones de procesos pueden servir como requisitos funcionales. Muchas empresas prefieren usar modelado de negocios como forma de llegar a la compresión de la funcionalidad.



**Requerimientos COTS**

COTS => commercial off the self (comercial fuera de la estantería) son productos o soluciones que están disponibles en el mercado para ser comprados. No necesitan desarrollo adicional ni personalización. En el mejor de los casos, en el peor necesita un desarrollo y una personalización después de instalado.

**Requerimientos no funcionales**

Aspecto y sensación, usabilidad y humanidad (experiencia de usuario), performance (que tan rápido, que tan seguro), operacional (entorno de operación del producto), mantenimiento y soporte, seguridad, cultura y política, legal.

Un requisito a veces se puede categorizar de varias formas. No importa mucho eso. El método cuando un requerimiento es ambiguo es atacar directamente la intención del stakeholder.