

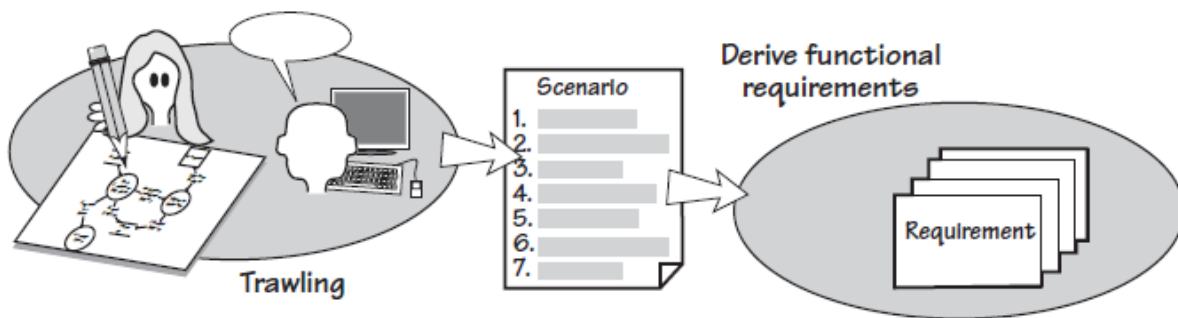
# Capitulo10\_Functional\_Requirements

Los req. funcionales especifican las cosas que el producto debe hacer (acciones que de cumplir para su existencia)

Los produce el *Analista* una vez que entiende la funcionalidad necesaria, y la utiliza para explicársela a los *Desarrolladores*.

Pasaje de PUC a  $\Rightarrow$  req. funcionales.

Los *Desarrolladores* los utilizaran para construir el *producto*.



Entender las diferencias entre **Req. y una solución** (la implementación tecnologica).

## Guía de formalidad

**Proyecto conejo:** es un metodo agil, basado en las iteraciones con los interesados y no tanto en la documentación formal, aunque se pueden incluir historias de usuarios.

**Proyecto caballo:** este método tiene al igual que el caballo iteraciones con los interesados, pero de forma más formal, se requiere documentar los Req. para despues transmitirlos de forma correcta. Es importante en la reuniones contar con el número adecuado de participantes.

Para comunicar req. se basa en Escenarios (casos de uso) y modelos de Clases de la información recolectada.

**Proyecto elefante:** tienen las especificaciones de los req. de forma correcta y completa.

## Req. Funcionales

Se deben expresar independientemente de la tecnología con la que se implementará.

La especificación del Req. permitirá desarrollar el producto correcto con la aprobación de la comprensión de los interesados.

## Descubriendo los Req. funcionales

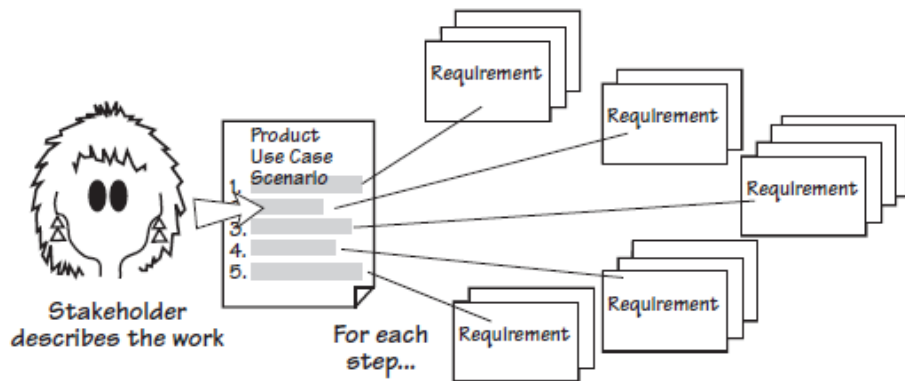
Partir dividiendo los **escenarios (casos de uso)** acorde los BE (eventos de negocio).

BE  $\Rightarrow$  producen  $\Rightarrow$  BUC  $\Rightarrow$  tienen 1 o más  $\Rightarrow$  PUC

PUCs brindan una vista a los involucrados de forma clara de lo necesario. (escribirlos en lenguaje entendible para los interesados)

**Figure 10.2**

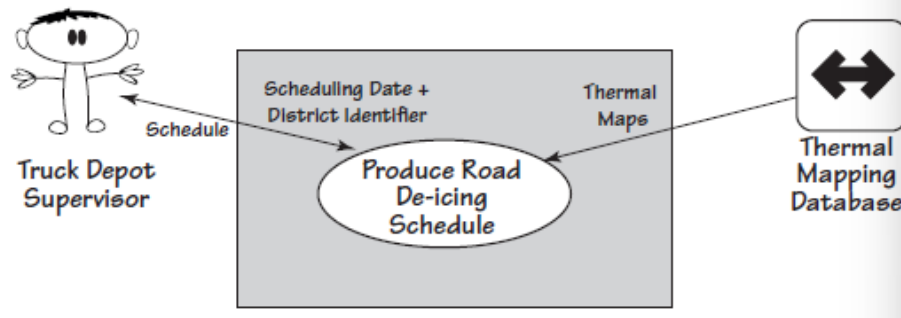
The scenario is a convenient way to work with stakeholders to determine the necessary functionality for a product use case. Each of the scenario's steps is decomposed into its functional requirements. The collection of functional requirements reveals what the product has to do to fulfill the product use case.



Ej. del Rompe Hielos

Product use case: Produce road de-icing schedule

1. Engineer provides a scheduling date and district identifier to the product.
2. Product retrieves the relevant thermal maps.
3. Product uses the thermal maps, district temperature readings, and weather forecasts to predict temperatures for each road section for the district.
4. Product predicts which roads will freeze and when they will freeze.
5. Product schedules available trucks from the relevant depots.
6. Product advises the engineer of the schedule.



Describir los pasos entre 3 y 10 puntos, asegura el detalle y comprensión de cada uno.

Después analizar cada paso para ver diferentes características que puedan tener (más detalles asociado al punto).

Una vez analizados y replanteados llevarlos con colegas para así validarlos y comprobar que sean entendibles.

## Nivel de detalle o Granularidad

Los Req. deben escribirse en una simple oración (evitando ambigüedad) y con un único verbo. Para hacerlo entendible por programadores y los interesados.

Utilizar componentes separados para los req., como forma de indicar las prioridades de los mismos.

## Descripción y justificación

**Descripciones**, cuando es simplemente algo que debe hacer.

**La justificación**, indica si el req. merece alguna atención especial.

Las justificaciones en las descripciones es para conocer porque están ahí, los desarrolladores y los testers conocen mejor de que se trata y podrán volcar más esfuerzo y dedicación en ello.

Las descripciones indican la solución al problema, pero con la justificación, se ve realmente la verdadera necesidad. (real req.)

## Información, tu arma secreta

Tan pronto como se comienza, se debe desarrollar un lenguaje con términos comunes a todos.

Usualmente se descubren cuando se pasa a identificar las entradas y salidas en el modelo de datos del negocio, esto provee un lenguaje común a tu equipo.

## Modelo de datos

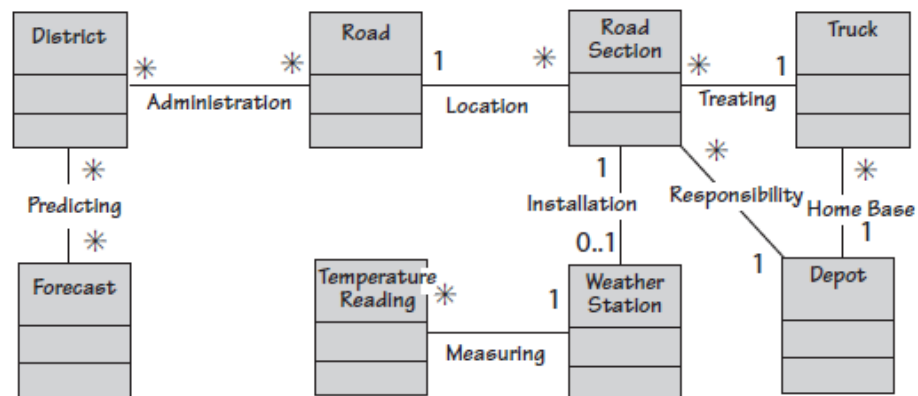
Usualmente llamados diagrama de clases o modelos entidad relación, son un rico indicador de las funcionalidades de un producto.

Modelo:

- clases o entidades, en rectángulos
- las asociaciones, con líneas entre 2 entidades.
- la cardinalidad, con asteriscos

**Figure 10.4**

A class diagram showing the stored data that the business needs to predict the formation of ice on roads. A truck is dispatched to treat a road when the readings from the weather station and the forecast indicate the road section is about to freeze.



## Diccionario de datos

Define los significados de las clases, atributos y asociaciones del modelo de datos.

Ej. Road = Road Name + Road Number

Para darle mayor interes, se le agrega una descripción.

La Road, esta junto a una autopista...

No contiene completamente las funcionalidades, pero debería especificar parte de los Req. Funcionales.

## Excepciones y alternativas

Las **excepciones** no son buscadas, pero si inevitables, por errores de procesamiento o acciones incorrectas. Muestran como el sistema se recupera de una falla.

Los req. también abarcan el que hacer cuando ocurre esta situación, como paso a seguir. Se debe tener claro cuando ocurre y una sola realidad para cada excepción.

Las **alternativas** son variaciones de los casos normales, usualmente provistos por los negocios de los interesados.

## Req. condicionales

A veces se necesita una condición para un req. Esto sucede cuando el req. solo se cumple si ocurre cierto proceso previo.

Este tipo raramente aparece en la especificación de req. Se deben discutir excepciones y alternativas de forma temprana para lograr pasárselos a los desarrolladores para que lo incorporen.

## Evadir la ambigüedad

Por más que se lleven escritos los req. se debe tener gran importancia a la potencial ambigüedad o desentendimiento que la ambigüedad puede causar.

Ej.

*The product shall show the weather for the next 24 hours.*

*The product shall communicate all roads predicted to freeze.*

## Req. tecnológicos

Surgen a partir de la solución tecnológica empleada, como parte de la necesidad de la tecnología utilizada.

Ej. si necesita recabar datos de la internet, es necesario que pueda conectarse y de manera segura.

## Agrupando req.

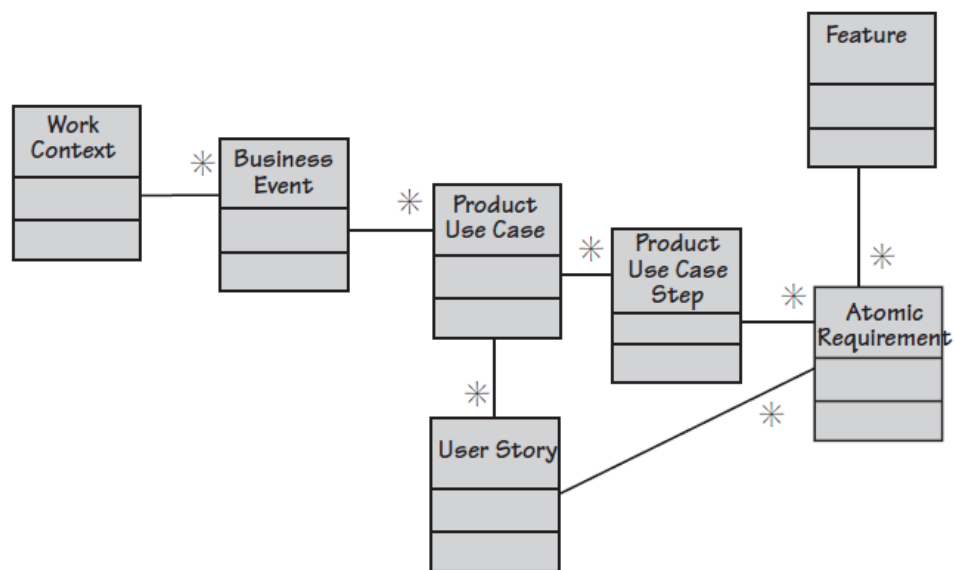
Agrupar los req. funcionales segun los casos de uso lleva a entenderlos mejor, asi como testearlos por completo.

Las “**Características**” pueden ayudar al proceso de prueba y validación de req. funcionales, ya que muchas veces es mejor agruparlas por las características de los req.

Las características pueden ser variadas e incluso pueden variar dentro del negocio, dependiendo del tiempo y del interesado.

Figure 10.6

A hierarchy of requirements. The work context is the highest-level statement of requirements; it is decomposed into the next level, the business events. The level below the business events comprises the product use cases, each of which is decomposed into a number of product use case steps. The lowest level includes the atomic requirements, each of which can be traced back up the hierarchy. Activity diagrams and user stories are also used to group atomic requirements. Features are a grouping often used by stakeholders from marketing or product version planning.



Pensar atómicamente en los req. ayuda a ver las especificaciones de req. al nivel más bajo.

Agrupar los req. según su jerarquía por 3 razones:

- Ser capaz de involucrar a las partes interesadas con diferentes profundidades, amplitudes y enfoques de interés.
- Para ayudarte a descubrir los req. atómicos en primer lugar
- Para ser capaz de ajustar el volumen y complejidad

Como sugerencia para saber el nivel de req. es necesario el evaluarlos de forma no subjetiva, para encontrar así la naturaleza de los mismos.

## Alternativas para Req. Funcionales

Manten en mente lo que haces, deberías entender de primera la funcionalidad correcta. Las alternativas presentadas son otra forma de verlas y comunicar el entendimiento de estas.

### Escenarios (casos de uso)

Los PUC y BUC son la forma de entender los req. de manera sencilla y asegurar el entendimiento de los mismos, por parte de los desarrolladores y los testers.

Lograr PUCs y BUCs de forma corta y entendible, genera un nivel de confianza en los mismos y evade los mal entendidos que se puedan ocasionar.

### Historias de Usuarios

Utilizan métodos ágiles para describir una necesidad funcional.

Se escriben de la siguiente forma:

*As a [role] I want [functionality] so that [reason for or use of the functionality].*

Usualmente son escritas en tarjetas por el propietario del producto, para clarificar el punto de vista del negocio. Muy frecuentemente hablan de características que deberían agregarse al producto.

Ej.

*As a moviegoer I want to have my tickets sent to my mobile phone so that I can avoid the box-office queue at the cinema.*

## Modelos del proceso de negocio

Realizar diagramas de actividad, junto con la descripción del mismo, pueden servir como req. funcional.

Figure 10.7

An activity diagram showing the product use case "Truck Depot reports problem with truck."

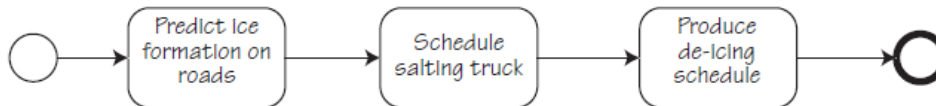
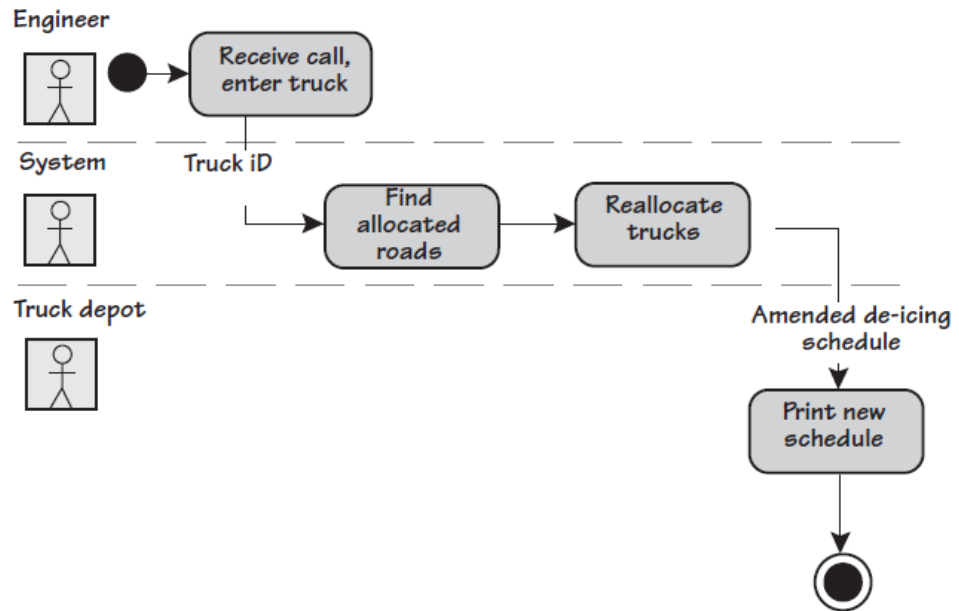


Figure 10.8

A BPMN process model for the BUC that produces the de-icing schedule. Consider how much extra specification your developers need to build the correct product.

Es preciso adecuar el estilo de diagrama acorde a la necesidad de entendimiento y la del equipo.

## Requerimientos para COTS (commercial off-the-shelf)

No importa que se refiera a productos instalables de carácter comercial o a los que no lo son, en ambos casos se refiere a COTS.

Son una alternativa a la construcción de software desde cero. Son BETAS que contienen necesidades pero requieren cierta modificación después de instalados. Usualmente son parametrizados y los desarrolladores pueden agregar código extra que sea necesario.



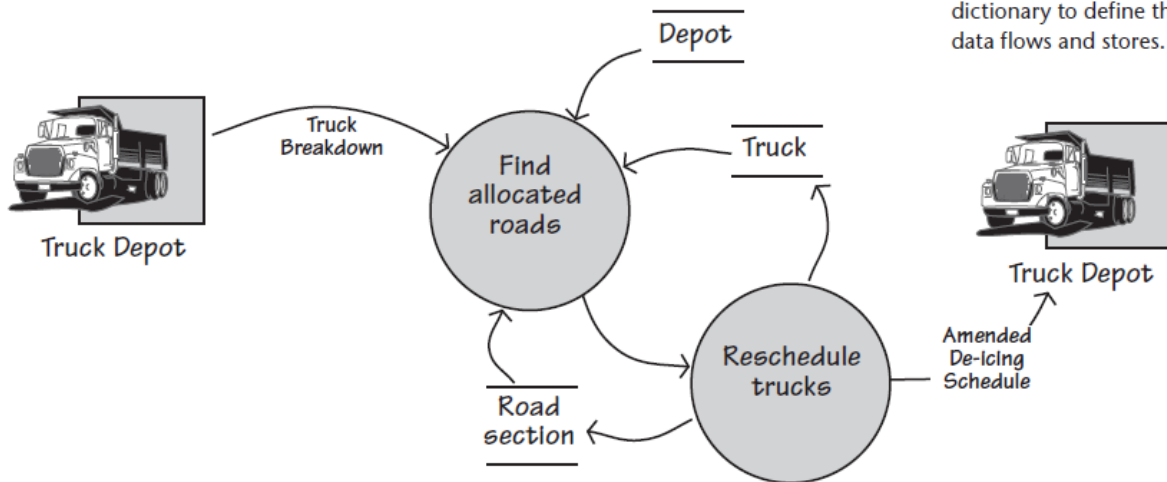


Figure 10.9

A data flow model of the product use case "Truck Depot reports problem with truck." This diagram is supported by process specifications for each of the processes shown, and a data dictionary to define the data flows and stores.

Son preferidos por muchas organizaciones ya que permite descubrir las verdaderas necesidades y es trabajo del analista de req. el descubrir las deficiencias.