REQUIREMENTS ANALYSIS: PART II

Material producido por la Universidad Peruana de Ciencias Aplicadas
UPC, 2021.

CONTENIDO

1. OBJETIVO DE APRENDIZAJE

Al finalizar la unidad de aprendizaje, el estudiante analiza los requisitos de software, a partir de la información recolectada y las necesidades identificadas empleando métodos, frameworks y artefactos ligados a la Ingeniería de Requisitos.

2. INTRODUCCIÓN

2.1. REQUIREMENTS ANALYSIS

- El **análisis de requisitos de software** es la práctica de ingeniería de software que traduce las necesidades y expectativas de los interesados en un conjunto viable de requisitos de software.
- Implica un conjunto de análisis y evaluaciones para examinar las necesidades de las partes interesadas y los requisitos de software para comprender las implicaciones de cada requisito en el alcance del esfuerzo de desarrollo.

Hasta el momento hemos aprendido sobre:

User personas — Empathy maps — As-is scenario maps

3. TO-BE SCENARIO MAP

Un Scenario (escenario) es un flujo de trabajo para una o más personas.

To-Be Scenario Maps pueden ser de ayuda para idear una mejor versión del futuro para sus usuarios.

Puede utilizarse para explorar posibles soluciones y ver cómo nuestras ideas encajarían en el mundo del usuario y ver cómo podría ayudar a dar atención a sus necesidades.

Puede servir para presentarlo ante stakeholders y usuarios para ayudar al equipo a enfocarse.

3.1. CARACTERÍSTICAS DE UN BUEN TO-BE SCENARIO MAP

- Se relaciona con un As-Is Scenario Map.
- Resuelve los pain points o satisface las necesidades del usuario.
- Trata aspectos de interés.
- La sesión de preparación no dura más de 60 minutos.

3.2. ¿CÓMO ELABORAR UN TO-BE SCENARIO MAP?

Instrucciones

1. Set up the activity

Dibuja cuatro filas y etiquétalas con phases, doing, thinking y feeling.

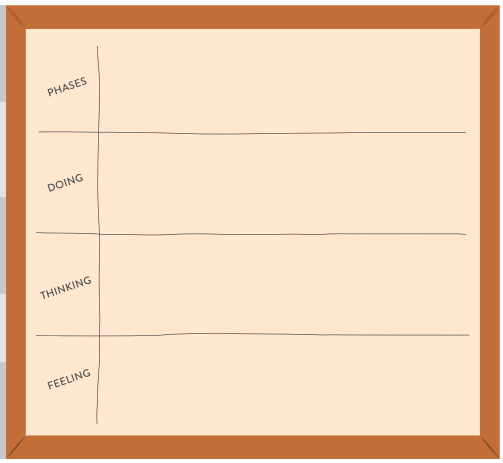
Phases. Los pasos que el user persona sigue para completar la tarea.

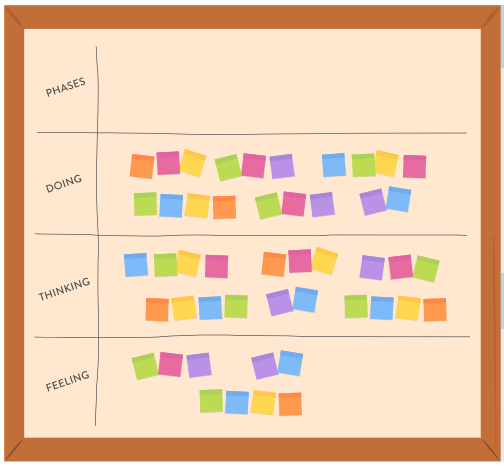
Doing. Las actividades sobre las que se avoca su user persona para completer el Step.

Thinking. Lo que la persona está pensando durante el Step.

Saying. Lo que la persona está diciendo durante el Step.

Feeling. Lo que la persona está sintiendo durante el Step.





2. Brainstorm, imagine your ideas exist in reality

Pregúntate: ¿Qué haría, pensaría y sentiría el usuario durante esta nueva experiencia que planteamos?
Fill in the corresponding rows, using one sticky per answer.
Revise su As-Is Scenario para encontrar oportunidades de mejora.

3. Cluster phases

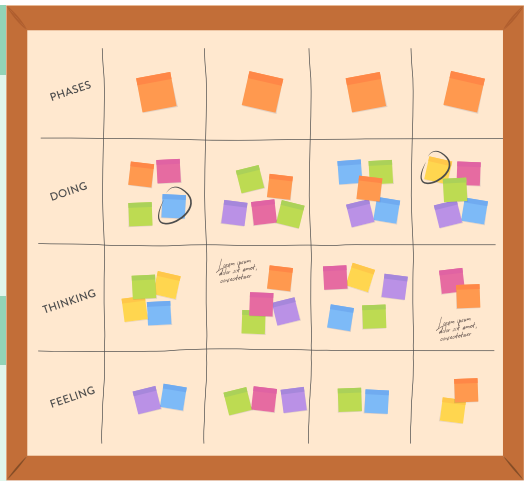
Agrupe notas similares, refine el orden y forme columnas que representen las fases de la futura experiencia.
Luego, nombre cada fase.

4. Identify highs and lows

Encierre en un círculo y etiquete las áreas particularmente positivas o negativas para su usuario.
Compare su Map con la experiencia actual del usuario.
Pregúntate: ¿Cómo este scenario puede mejorar la experiencia del usuario?

5. Playback

Seleccione una o dos personas para que presenten el to-be Scenario Map.



4. PRODUCT FEATURES

4.1. FEATURE (CARACTERÍSTICA)

Un Software Feature o característica de Software es uno de los términos fundamentales en la Ingeniería de Software. La IEEE (Institute of Electrical and Electronics Engineers) lo define como:

> "A distinguishing characteristic of software item (e.g. Performance, portability, or functionality)."
>
> IEEE

Desde el punto de vista Agile, Feature es una función con valor percibible por el cliente, de otro modo no justifica su incorporación en el producto. Se expresa en la forma <action><result><object>.

Feature Set = Conjunto de Features

4.2. FEATURE SET EXAMPLE

A continuación, un ejemplo de Feature Set típico de soluciones de E-commerce.

Administration /Backend Functionality

- Support unlimited products and categories
- Product-to-categories structure
- Add/Edit/Remove categories, products, manufacturers, costumers and reviews
- Categories-to-categories structure
- Support for physical (shippable) and virtual (downloadable) products
- Administration area secured with a username and password
- Contact costumers directly via email or newsletter
- Easily backup and restore the database
- Print invoices and packaging lists from the order screen
- Statistics for products and customers
- Select what to display, and in what order, in the product listing page
- Support for static and dynamic banners with full statistics

Search and Match

- All orders stored in the database for fast and efficient retrieval
- Customers can view their order history and order statuses
- Customers can maintain their accounts.
- Address book for multiple shipping and billing addresses
- Temporary shopping cart for guests and permanent shopping cart for customers
- Fast and friendly quick search and advanced search features
- Product reviews for an interactive shopping experience
- Foreseen checkout procedure
- Secure transactions with SSL
- Number of products in each category can be shown or hidden
- Global and per-category bestseller lists
- Display what other customers have ordered with the current product shown
- Breadcrumb trail for easy site navigation

Communication

- Dynamic product attributes relationship
- HTML based product descriptions
- Automated display of specials
- Control if out of stock products can still be shown and are available for purchase
- Customers can subscribe to products to receive related emails/newsletters

Site Administration

- Accept credit cards, paypal and offline payment methods
- Pay using purchase orders and admin purchase order management

5. BEHAVIOR & QUALITY ATTRIBUTES

5.1. BEHAVIOR (COMPORTAMIENTO)

- "The way in which something functions or operates".
- "Anything that an organism does involving action and response to stimulation".

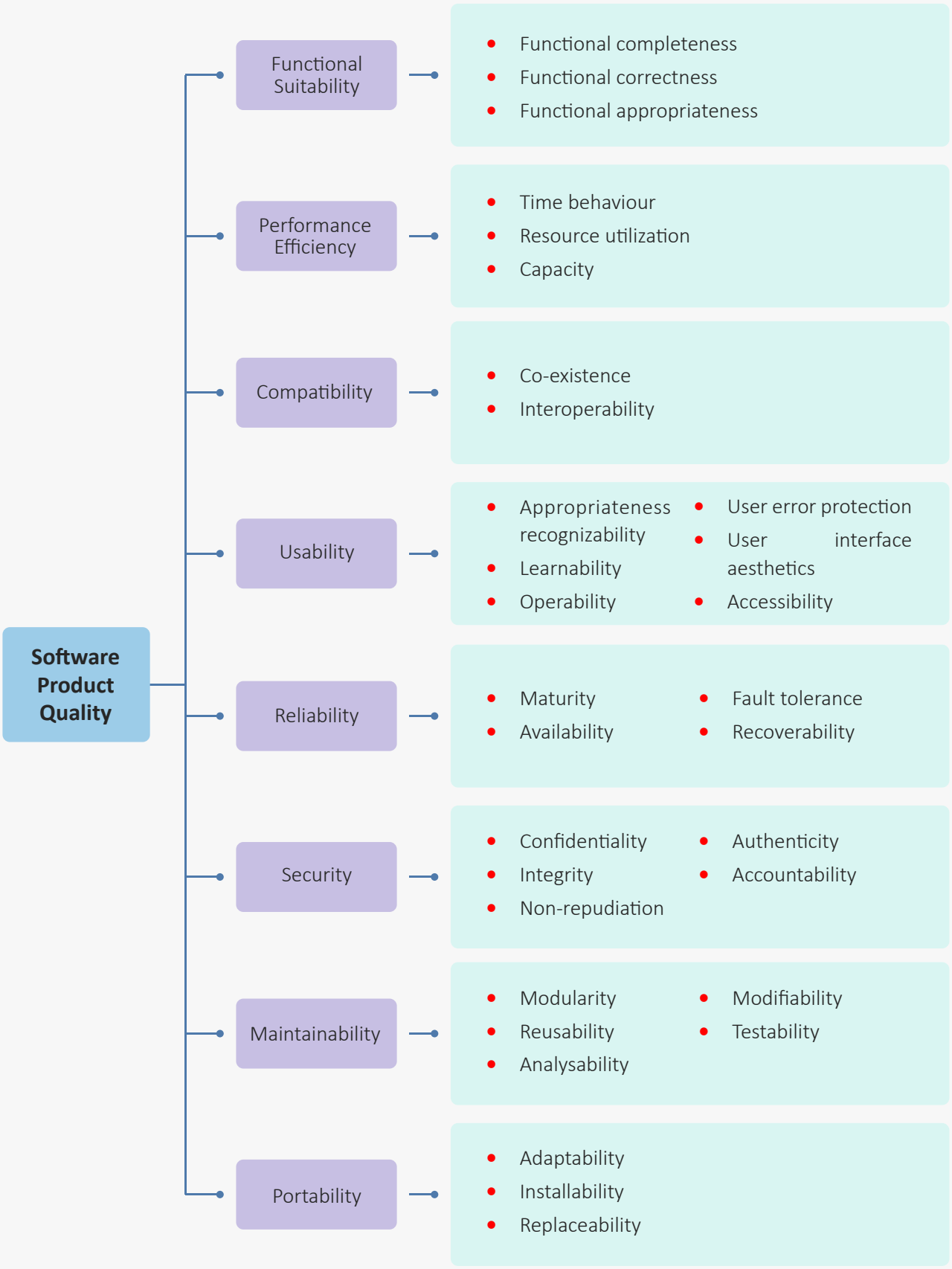
Merriam Webster

5.2. QUALITY ATTRIBUTES (ATRIBUTOS DE CALIDAD)

- Conforme vamos descubriendo qué debería hacer la solución, comienzan a aparecer las preguntas sobre cómo debería trabajar el producto, es decir, comienzan a descubrirse posibles Non-functional requirements.
- Para ello, es útil entender qué son Quality Attributes.

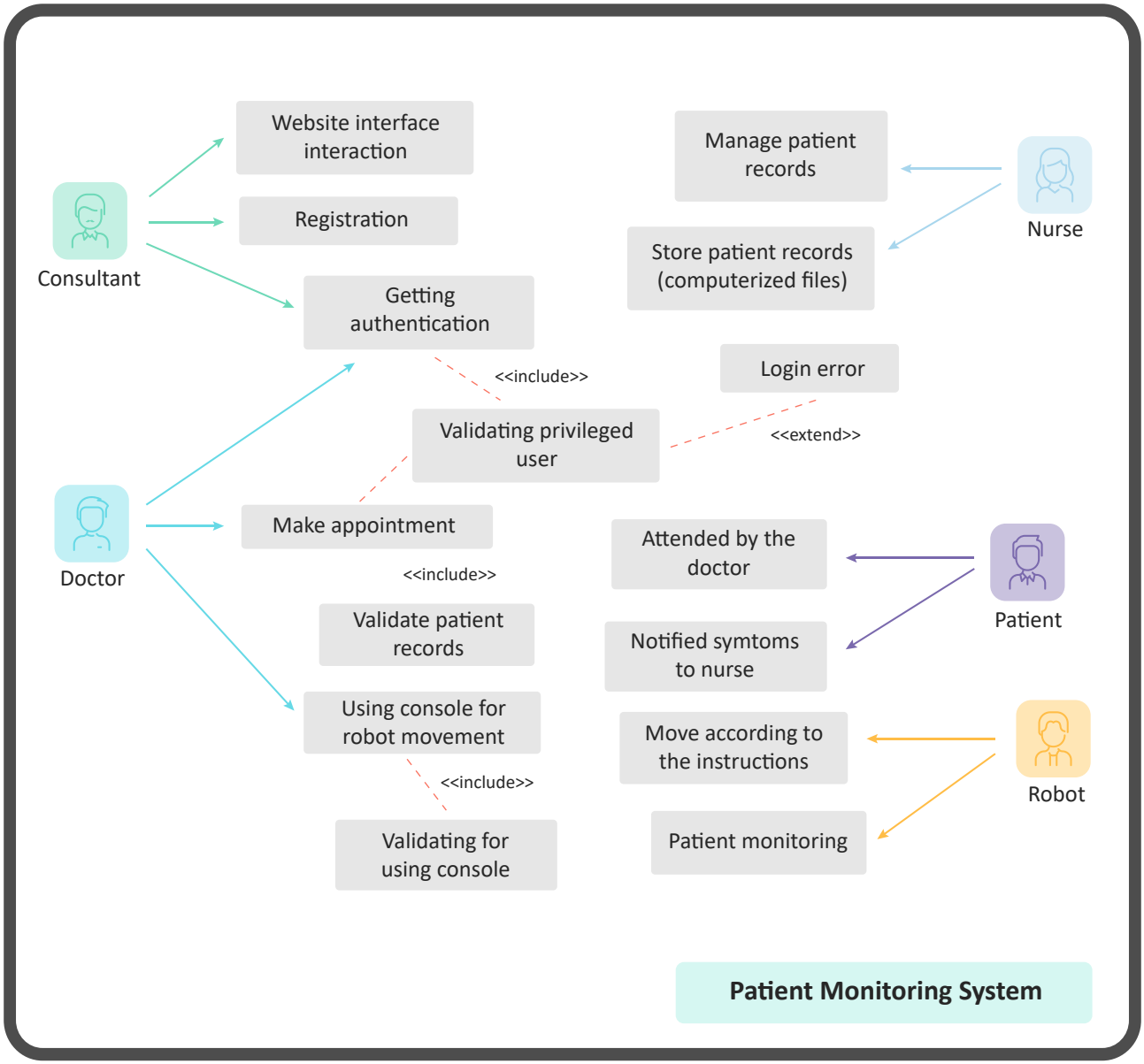
- Podemos decir que son non-functional requirements identificados los que se utilizan para evaluar el producto. Suelen ser relevantes para la Arquitectura del Software.

- La imagen que se presenta ilustra los principales atributos de calidad de software que considera la familia de normas ISO/IEC 25000 para lo que se conoce como SQuaRE – System and Software Quality Requirements and Evaluation (Requisitos y Evaluación de Calidad de Productos de Software y Sistemas).


Software Product Quality
Functional Suitability
Functional completeness
Functional correctness
Functional appropriateness
Performance Efficiency
Time behaviour
Resource utilization
Capacity
Compatibility
Co-existence
Interoperability
Usability
Appropriateness recognizability
Learnability
Operability
User error protection
User interface aesthetics
Accessibility
Reliability
Maturity
Availability
Fault tolerance
Recoverability
Security
Confidentiality
Integrity
Non-repudiation
Authenticity
Accountability
Maintainability
Modularity
Reusability
Analysability
Modifiability
Testability
Portability
Adaptability
Installability
Replaceability

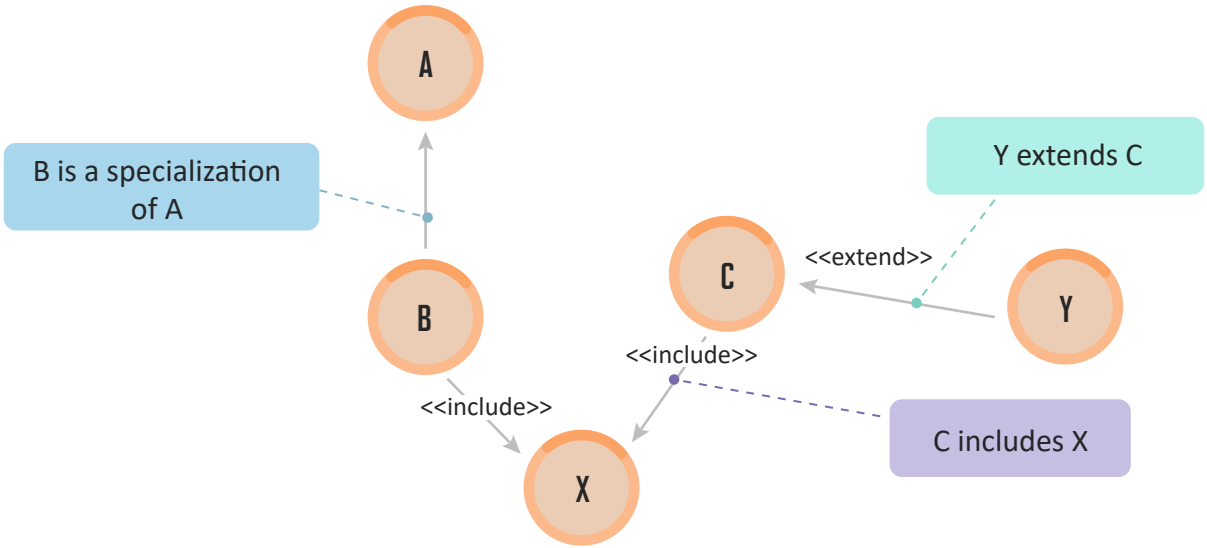
6. USE CASE MODEL

- Permite representar una propuesta de solución como un conjunto de Actors que interactúan con Use Cases.
- Compuesto por Use Case Diagram y Use Cases Specifications.
- Cada Use Case es un conjunto de escenarios que buscan lograr un propósito concreto de valor para el Actor.



El Use Case Diagram pertenece a la notación UML y es utilizado por marcos para el proceso de software como RUP (Rational Unified Process).

6.1. RELACIONES EN LOS USE CASE DIAGRAM



Include (Inclusión)

Esta relación se utiliza cuando un caso de uso base incorpora de manera explícita el comportamiento de otro en algún lugar de su secuencia. La relación de inclusión sirve para enriquecer un caso de uso con otro y compartir una funcionalidad común entre varios casos de uso. Es posible utilizarla para estructurar un caso de uso describiendo sus subfunciones. El caso de uso incluido existe únicamente con ese propósito, ya que no responde a un objetivo de un actor.

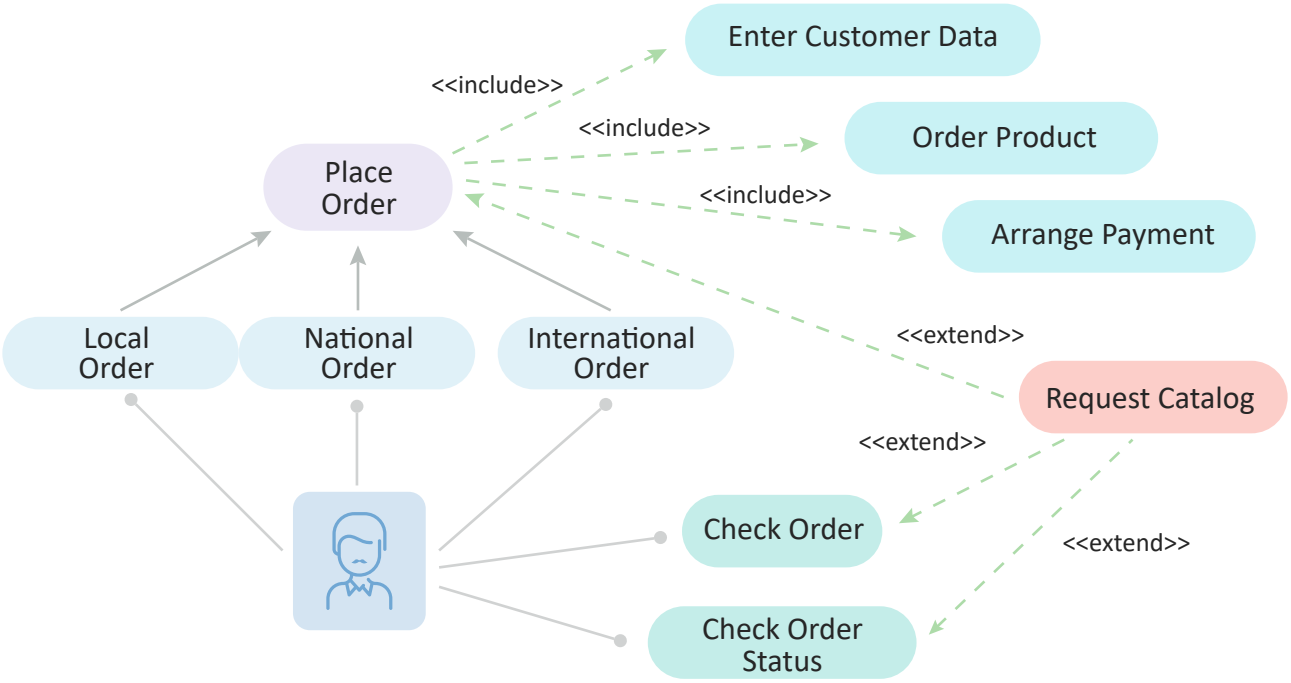
Extend (Extensión)

Se presenta cuando un caso de uso base incorpora implícitamente el comportamiento de otro caso de uso en el lugar especificado indirectamente por este otro caso de uso. En el caso de uso base, la extensión se hace en una serie de puntos concretos y previstos en el momento del diseño. También son llamados **puntos de extensión**, los cuales no son parte del flujo principal. La relación de extensión sirve para modelar: la parte opcional del sistema, un subflujo que solo se ejecuta bajo ciertas condiciones o varios flujos que se pueden insertar en un punto determinado.

Specialization (Especialización)

Esta relación representa que un caso de uso (subcaso) hereda el comportamiento y significado de otro, es decir, las relaciones de comunicación, inclusión y extensión del super-caso de uso. Se podría dar que este super-caso de uso es abstracto y corresponde a un comportamiento parcial completado en el subcaso de uso. O dicho de otra manera, los casos de uso “hijo” son una especialización del caso de uso “padre”.

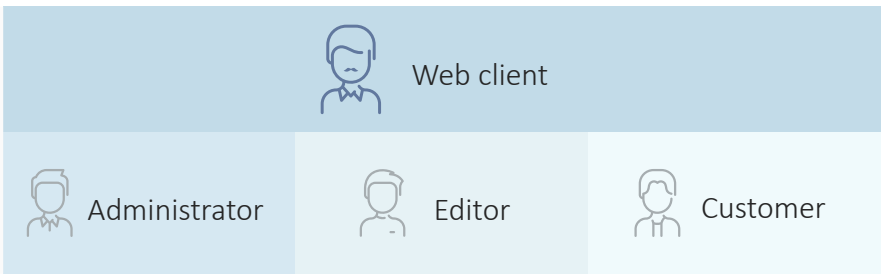
Aquí un ejemplo de un Use Case Diagram que ilustra el uso de las relaciones entre Use Cases.

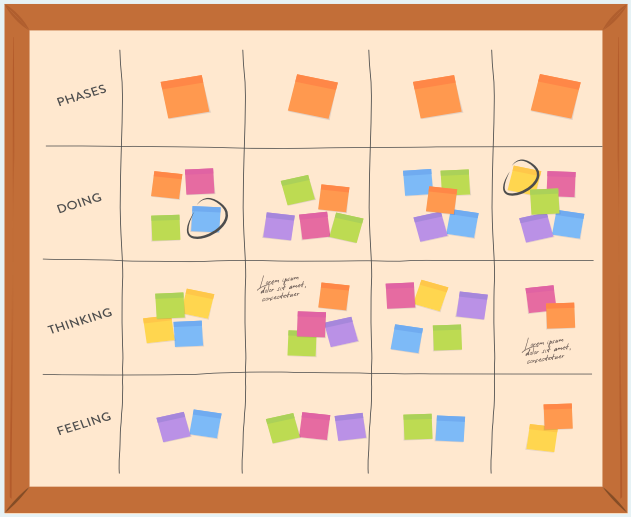


6.2. RELACIONES ENTRE ACTORES

Specialization

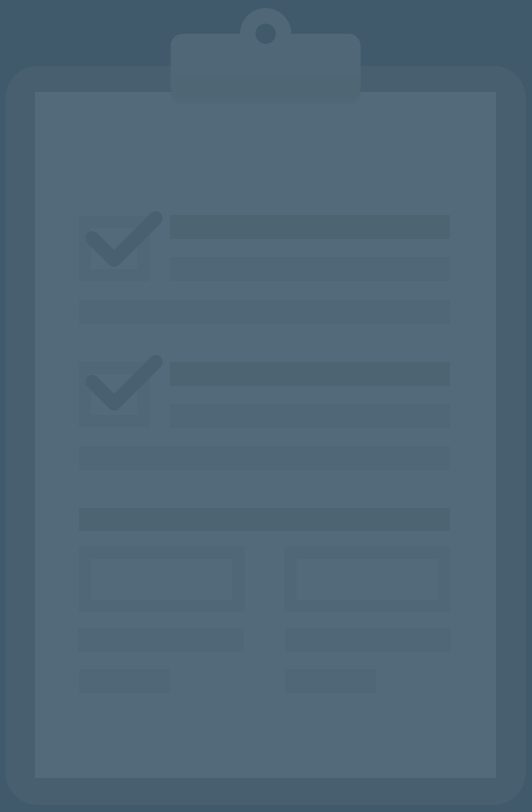
Puede existir relaciones y herencia entre Actors (actores) de un Use Case.





7. CONCLUSIONES

- Requirements Analysis es un proceso importante para identificar e investigar las necesidades y expectativas de los Stakeholders para el desarrollo del software.
- El enfoque To-Be Scenario Map permite explorar situaciones y soluciones a las necesidades de los usuarios
- El Feature es el elemento de valor que es percibible por el cliente
- El Behavior y Quality Attributes son las características del funcionamiento del software en la evaluación producto
- Con el Use Case Model se establecen las interacciones entre distintos usuarios y los tipos de relaciones.



REFERENCIAS

Para profundizar:

- https://www.ibm.com/design/thinking/page/toolkit/activity/to-be-scenario-map
- https://syndicode.com/2018/05/03/12-software-architecture-quality-attributes/
- http://agilemodeling.com/artifacts/feature.htm
- https://www.uml-diagrams.org/use-case-diagrams-examples.html

Autor: Ángel Augusto Vasquez Nuñez