Análisis y Especificación de Requisitos No Funcionales

Pablo Sánchez

Ingeniería del Software y Tiempo Real
Dpto. Ingeniería Informática y Electrónica
Universidad de Cantabria
Santander (Cantabria, España)
p.sanchez@unican.es

Advertencia

Todo el material contenido en este documento no constituye en modo alguno una obra de referencia o apuntes oficiales mediante el cual se puedan preparar las pruebas evaluables necesarias para superar la asignatura de Ingeniería de Requisitos.

Este documento contiene exclusivamente una serie de diapositivas cuyo objetivo es servir de complemento visual a las actividades realizadas en el aula para la transmisión del contenido sobre el cual versarán las mencionadas pruebas evaluables.

Dicho de forma más clara, estas transparencias no son apuntes y su objetivo no es en modo alguno servir para que el alumno pueda preparar la asignatura.

Objetivos del Tema

1. Entender la naturaleza y papel de los requisitos no funcionales en las especificaciones de requisitos.
2. Conocer y saber interpretar la norma ISO 25010.
3. Conocer y comprender el concepto de *sistema confiable* y su importancia dentro de los sistemas sociotécnicos.
4. Saber identificar, modelar y especificar requisitos de *seguridad* de un sistema software confiable.
5. Saber identificar, analizar y evaluar influencias entre requisitos no funcionales sobre otros.
6. Conocer y comprender el funcionamiento de las técnicas de negociación.

Bibliografía


Sommerville, I. (2010).
Software Engineering.
Addison Wesley, 9 edition.


International Standard Organization (ISO) / International Electrotechnical Commission (IEC) (2011).
Systems and software engineering Systems and software Quality Requirements and Evaluation (SQuaRE) System and software quality models. ISO/IEC Standard 25010


Haley, C., Laney, R., Moffett, J., and Nuseibeh, B. (2008).
Security Requirements Engineering: A Framework for Representation and Analysis.
IEEE Transactions on Software Engineering, 34(1):133–153.

Bibliografía


Chung, L., Nixon, B. A., Yu, E., and Mylopoulos, J. (1999).
Non-Functional Requirements in Software Engineering.
Kluwer Academic Puvlishers.

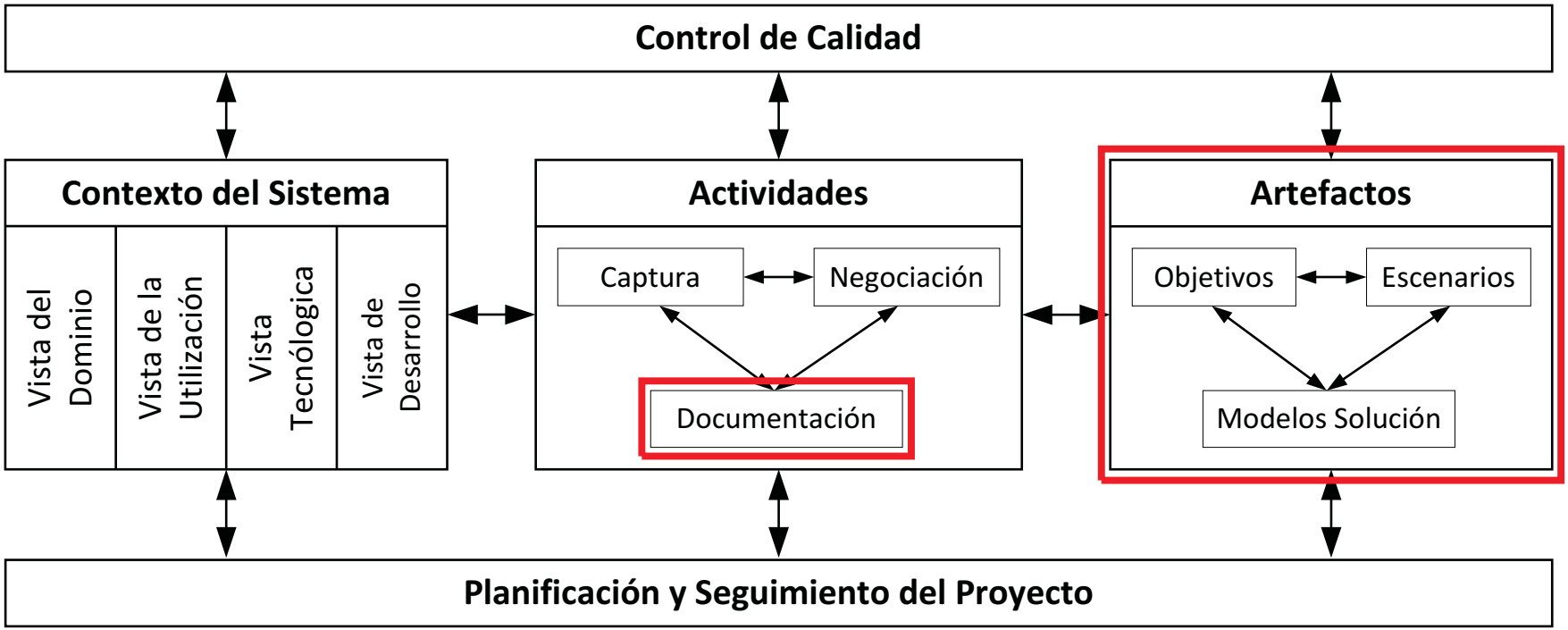

Pohl, K. (2010).
Requirements Engineering: Fundamentals, Principles and Techniques.
Springer.


Sindre, G. and Opdahl, A. L. (2004).
Eliciting security requirements with misuse cases.
Requirements Engineering, 10(1):34–44.

Índice

Índice

Proceso de Ingeniería de Requisitos



Índice

Requisitos No Funcionales

Requisito No Funcional [Chung et al., 1999]

Un Requisito No Funcional de un sistema sw es un requisito que no indica qué debe hacer el sistema, sino cómo debe hacerlo.

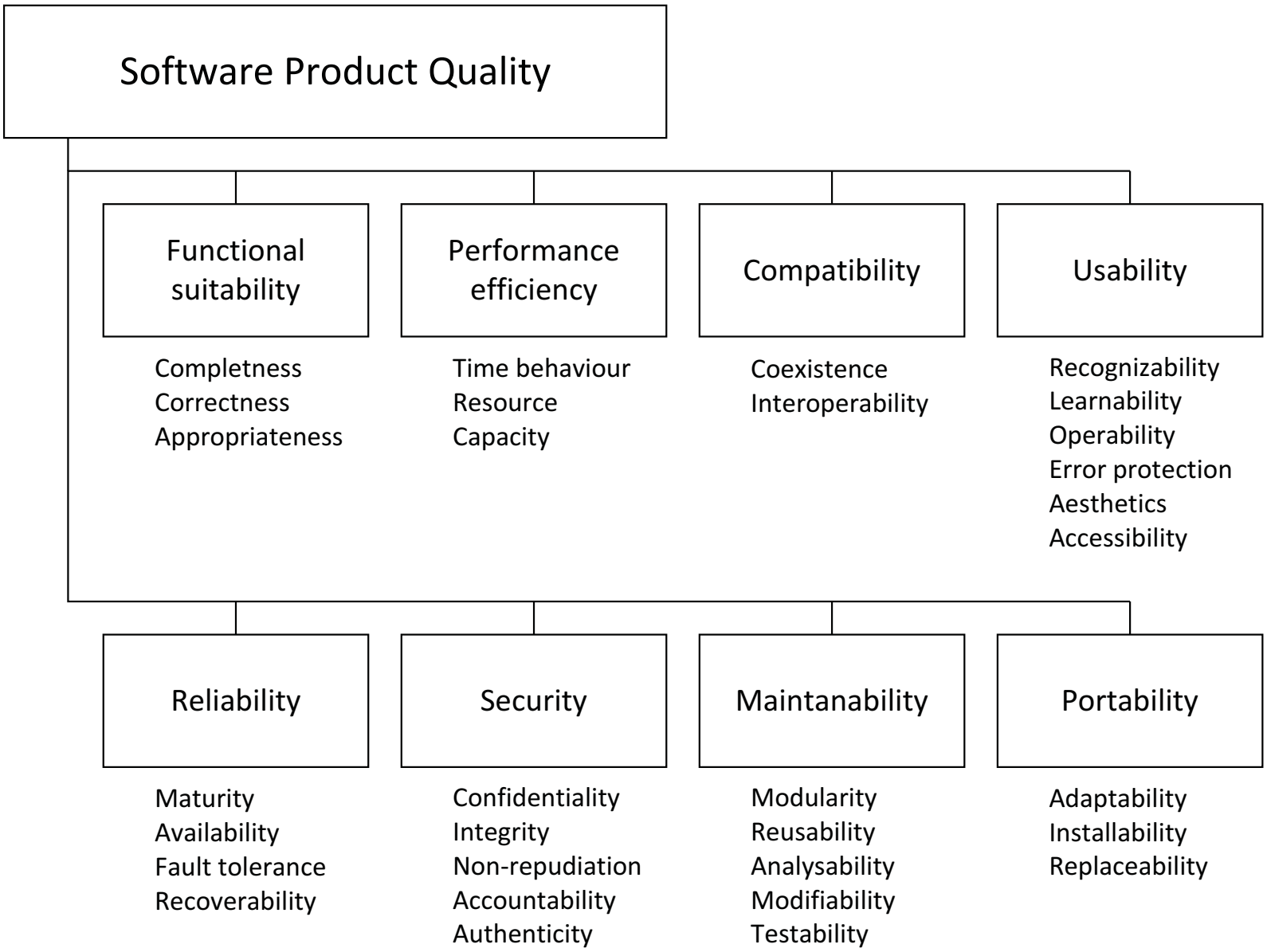
Índice

Características Requisitos No Funcionales

1. Suelen aplicarse al sistema como un todo, tienen influencia global.
2. Su no satisfacción puede significar el fracaso del proyecto.
3. Aparecen recurrentemente a través de aplicaciones de diferente índole y/o dominio.
4. Se materializan en requisitos funcionales frecuentemente.
5. Complejos y costosos de verificar y medir en ocasiones.
6. Suelen presentar conflictos entre ellos.
7. A pesar de su coste, podrían no utilizarse nunca.

Índice

ISO 25010 - Calidad de un Producto Sw



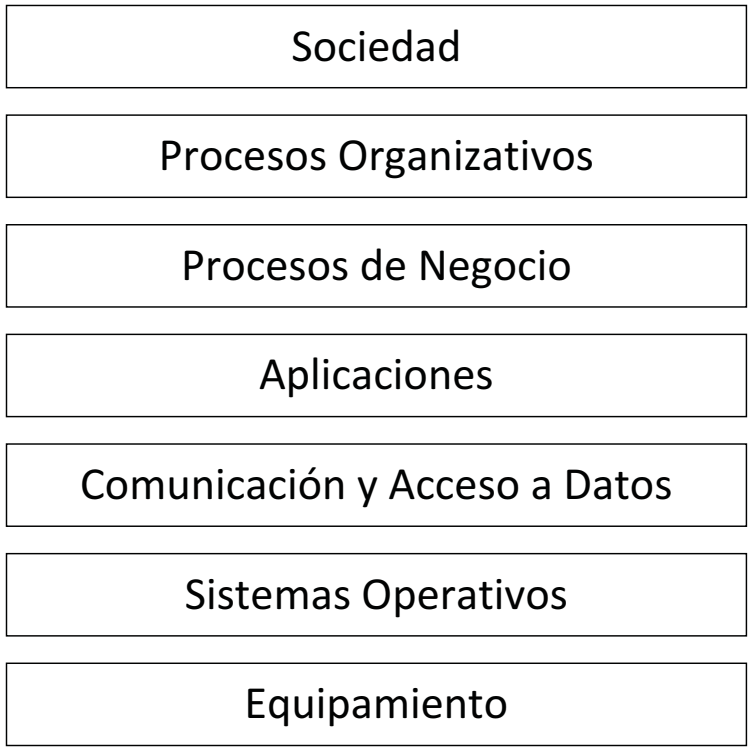
Índice

Índice

Sistemas Sociotécnicos

Sistemas Sociotécnicos

Un *sistema sociotécnico* es un sistema empresarial diseñado para ayudar a conseguir un objetivo estratégico.



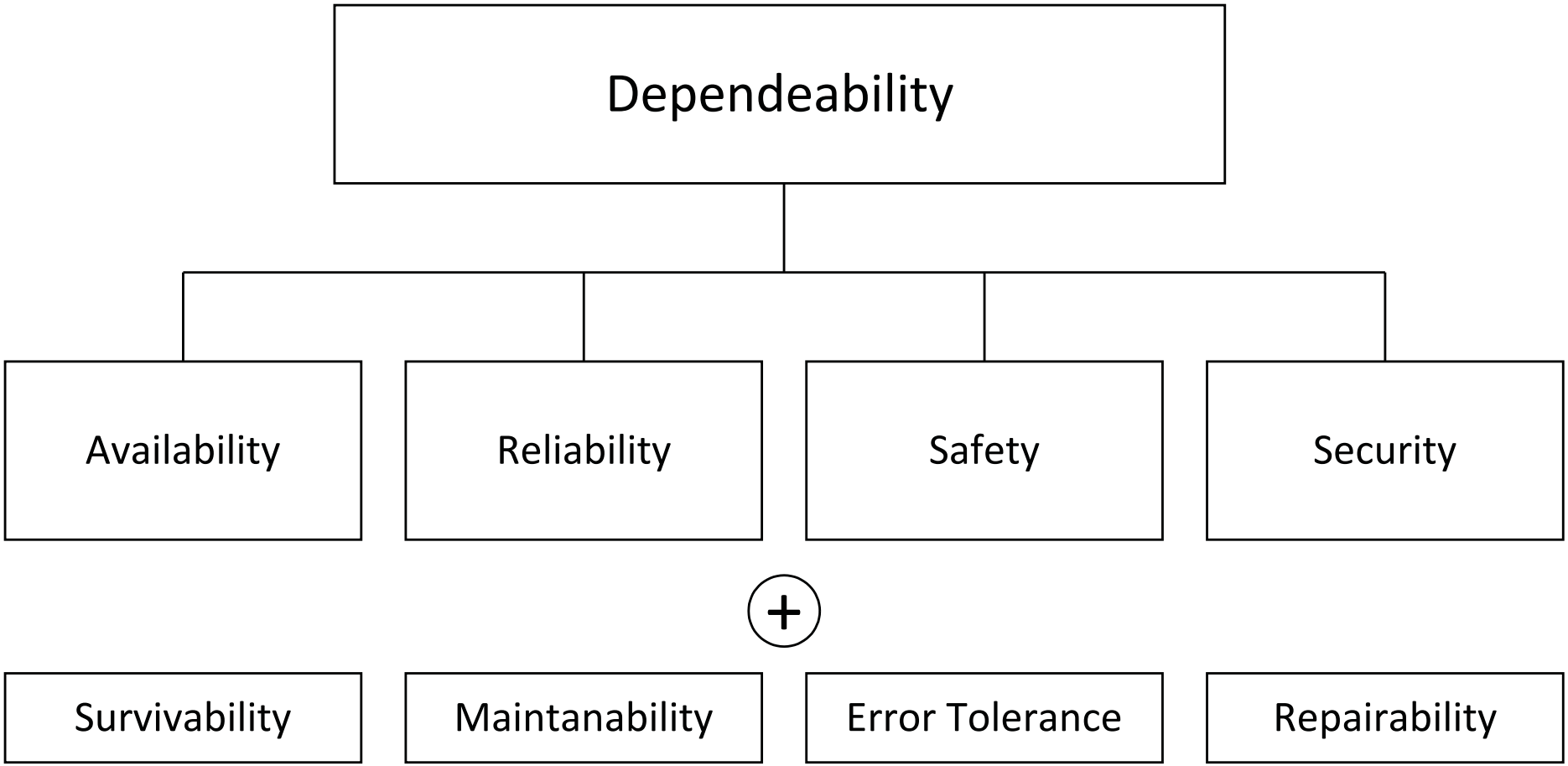
Índice

Sistemas Confiables

Sistema Confiable (*Dependeable System*)

Un sistema se dice confiable cuando podemos tener cierta seguridad de que operará correctamente cuando lo necesitemos, no produciéndonos ni daños ni perjuicios. Un sistema confiable destaca por satisfacer cuatro requisitos no funcionales claves: (1) *availability*; (2) *reliability*; (3) *safety*; y (4) *security*.

Sistemas Confiables



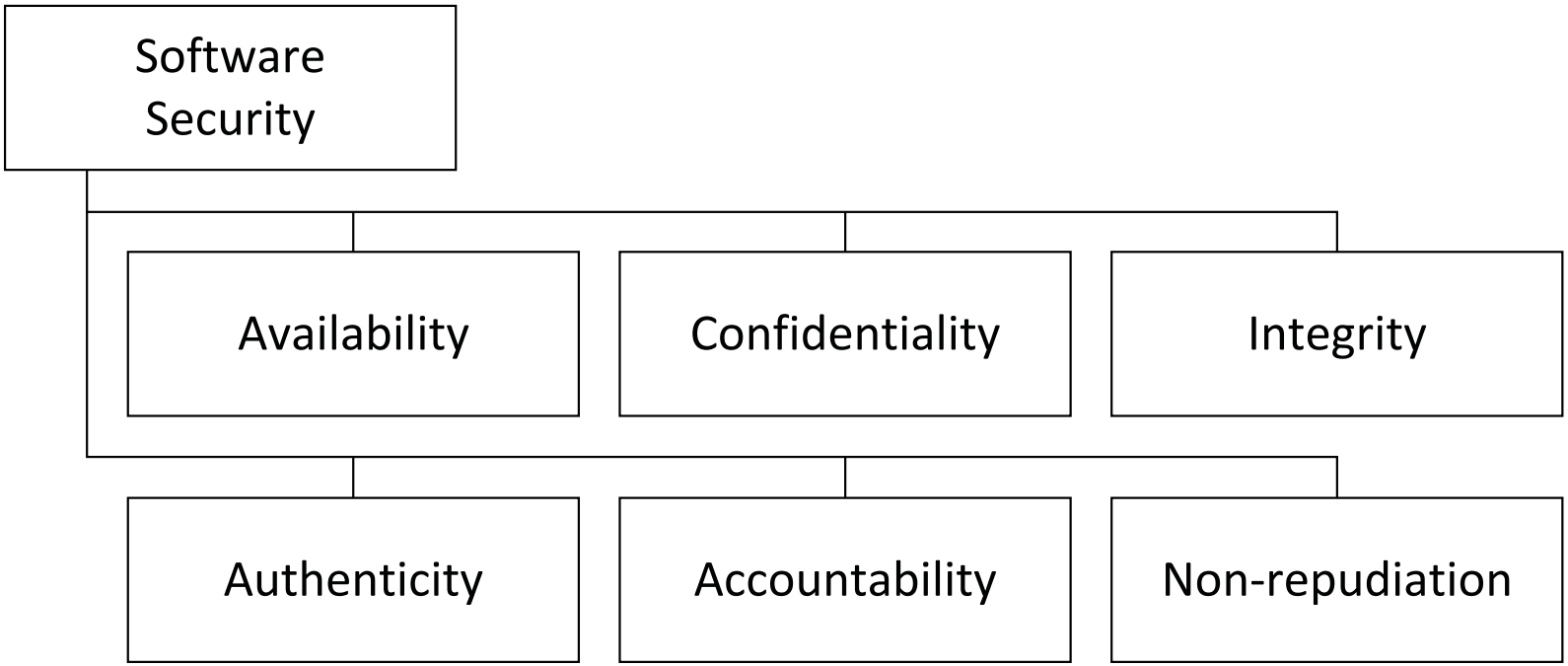
Índice

Índice

Seguridad de un Sistema Software [Sommerville, 2010]

Seguridad de un Sistema Software

Capacidad de un sistema software de protegerse de ataques externos, los cuales pueden ser accidentales o deliberados, así como de resistirlos en caso de que se produzcan.



Índice

Terminología sobre Seguridad Software

Activo Elemento (físico o lógico) de un sistema sw que posee cierto valor y por tanto debe protegerse de posibles ataques.

Amenazas Circunstancia que, de materializarse, causaría un daño o perjuicio al sistema.

Exposición Perjuicio o daño que sufriría un sistema sw cuando se materializa una amenaza.

Vulnerabilidad Debilidad de un sistema sw que puede explotarse para materializar una amenaza.

Ataque Utilización de una vulnerabilidad para materializar una amenaza.

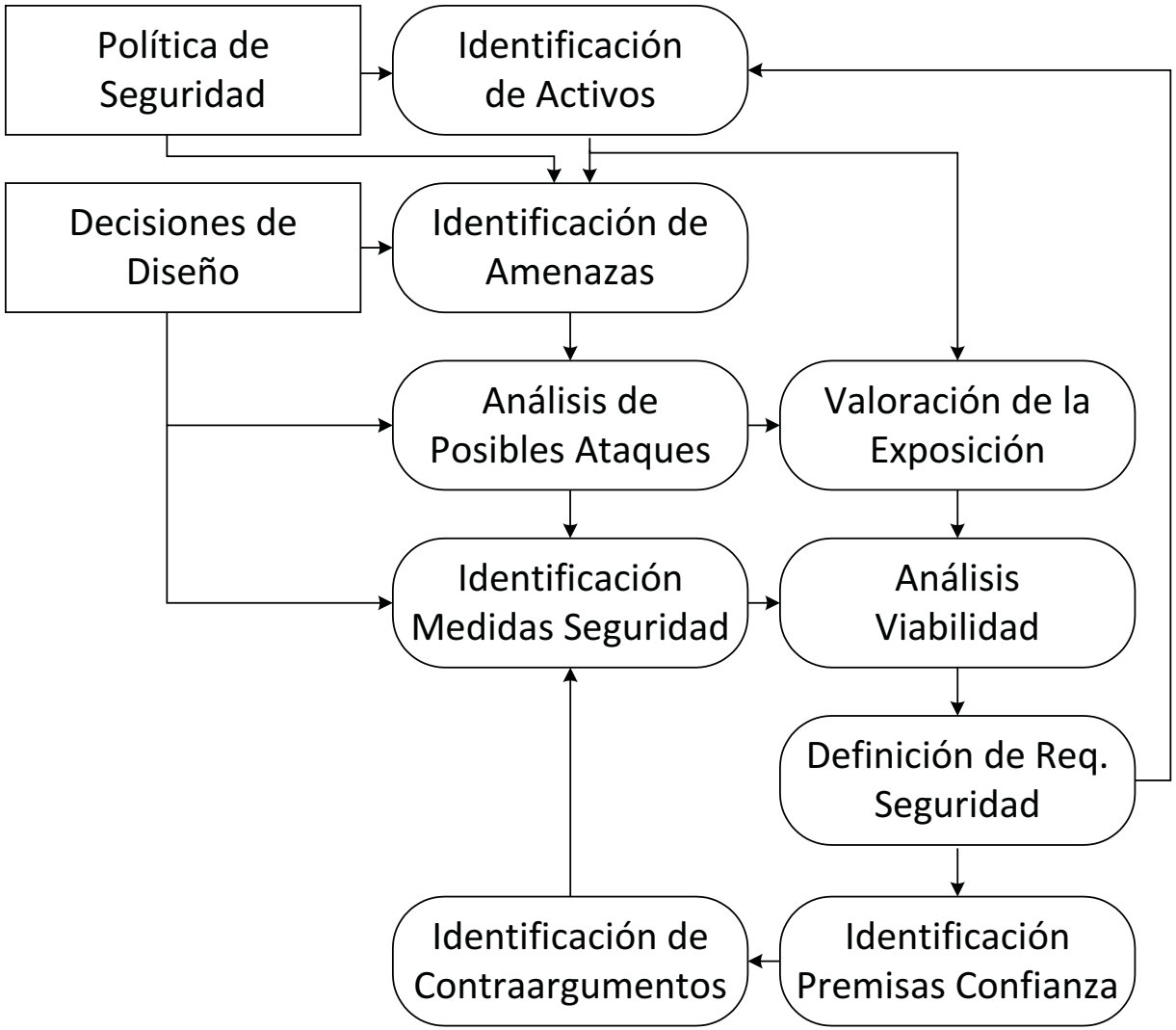
Medida de Control Decisión adoptada para reducir la vulnerabilidad de un sistema o mitigar un ataque.

Premisa de Confianza Hipótesis que ha de ser verdad para que la medida de control sea efectiva.

Contraargumento Argumento que invalidaría la premisa de confianza.

Índice

Proceso de Ingeniería de Requisitos de Seguridad



Índice

Modelo STRIPE de Amenazas

Spoofing Suplantación fraudulenta de un tercero.

Tampering Modificación no autorizada de datos.

Repudiation Realización de acciones que impidan rastrear otras operaciones.

Information Disclosure Acceso a información no autorizada.

Denial of Service Impedir el acceso o utilización a usuarios legítimos.

Elevation of Privilege Adquirir un rol que permita ejecutar acciones para los que no se posee permiso en inicio.

Índice

Técnicas Generales de Control de la Seguridad

1. Evitar o reducir el riesgo.
2. Detección y neutralización de ataques.
3. Limitación de la exposición y capacidad de recuperación.

Redundancia y Diversidad

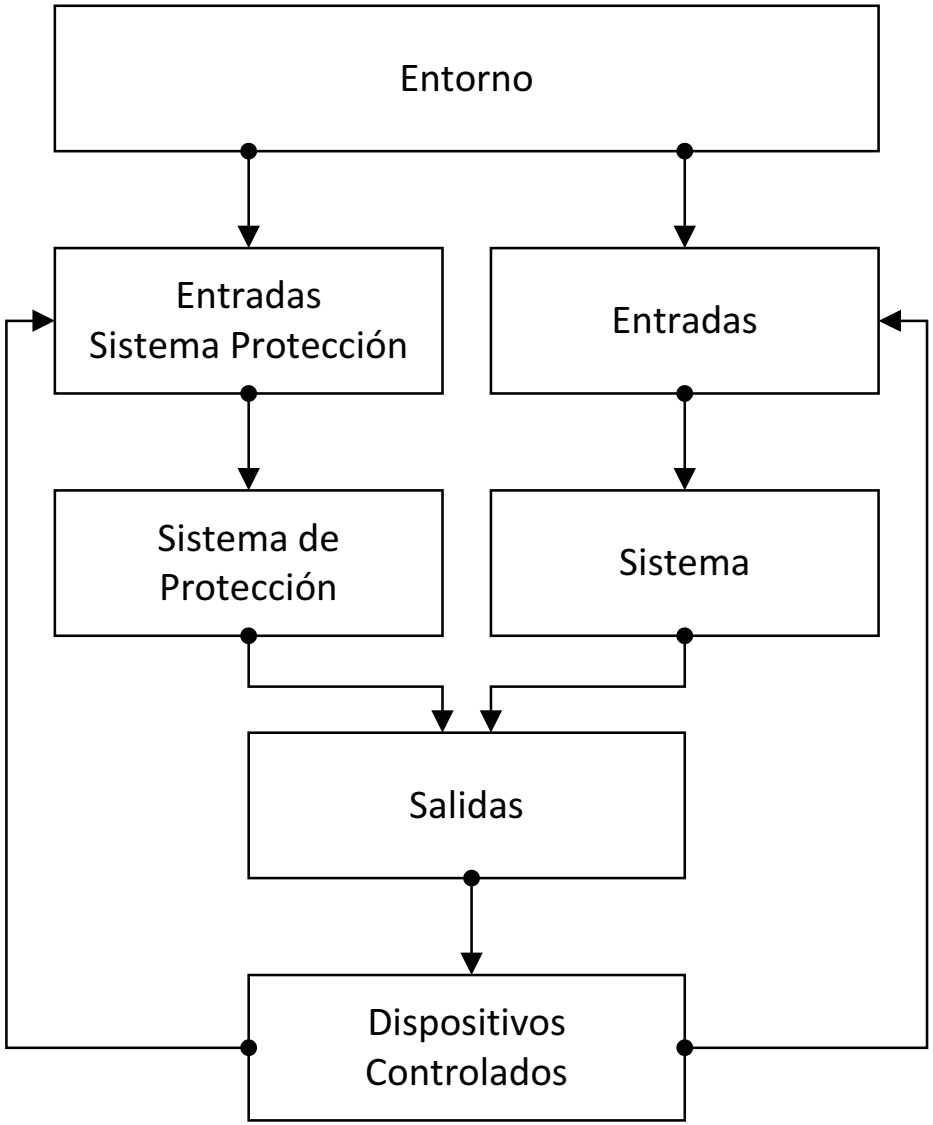
Redundancia

La *redundancia* consiste en añadir recursos adicionales a un sistema de forma que puedan ser utilizados en caso de que fallen sus elementos principales.

Diversidad

La *diversidad* consiste en hacer que los recursos adicionales sean diferentes a los principales, de manera que se evite que un mismo fallo afecte a todos.

Sistemas de Protección



Índice

Casos de Mal Uso - Elementos

Actor de mal uso (*Misuser*) [Sindre and Opdahl, 2005]

Actor(persona o sistema) que, de forma deliberada o no deliberada, inicia un caso de mal uso.

Caso de Mal Uso (*Misuse Case*) [Sindre and Opdahl, 2005]

Escenario, incluyendo variaciones y extensiones, ejecutado por un actor de mal uso, que, si se ejecuta con éxito, constituye una amenaza para la seguridad del sistema.

Casos de Mal Uso - Relaciones

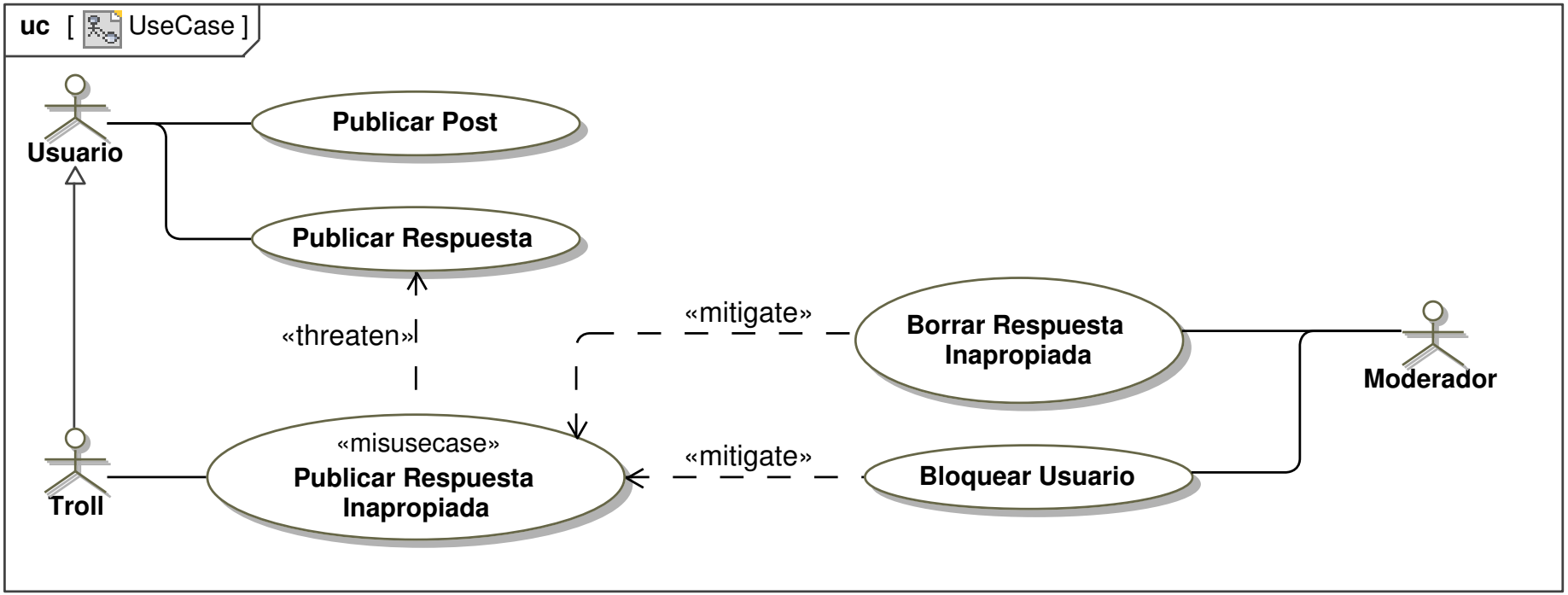
Amenaza (*theaten*) [Sindre and Opdahl, 2005]

Relación entre un caso de mal uso y un caso de uso que indica que el el caso de mal uso utiliza o se basa en el caso de uso para ejecutar un ataque.

Mitiga (*Mitigate*) [Sindre and Opdahl, 2005]

Relación entre un caso de uso y un caso de mal uso que indica que el caso de uso se utiliza para evitar o mitigar el daño o perjuicio causado por un caso de mal uso.

Casos de Mal Uso - Notación



Índice

Directrices para Garantizar la Seguridad Sw

1. Basar el proceso de gestión de la seguridad en la política de seguridad de la organización.
2. Evitar los *talones de Aquiles*.
3. Fallar de forma segura.
4. Balancear seguridad y usabilidad.
5. Registrar las acciones de los usuarios.
6. Utilizar redundancia y diversidad para reducir riesgos.
7. Validar todas las entradas.
8. Parcelar los activos.
9. Gestionar y monitorizar el despliegue.
10. Diseñar mecanismos de recuperación.

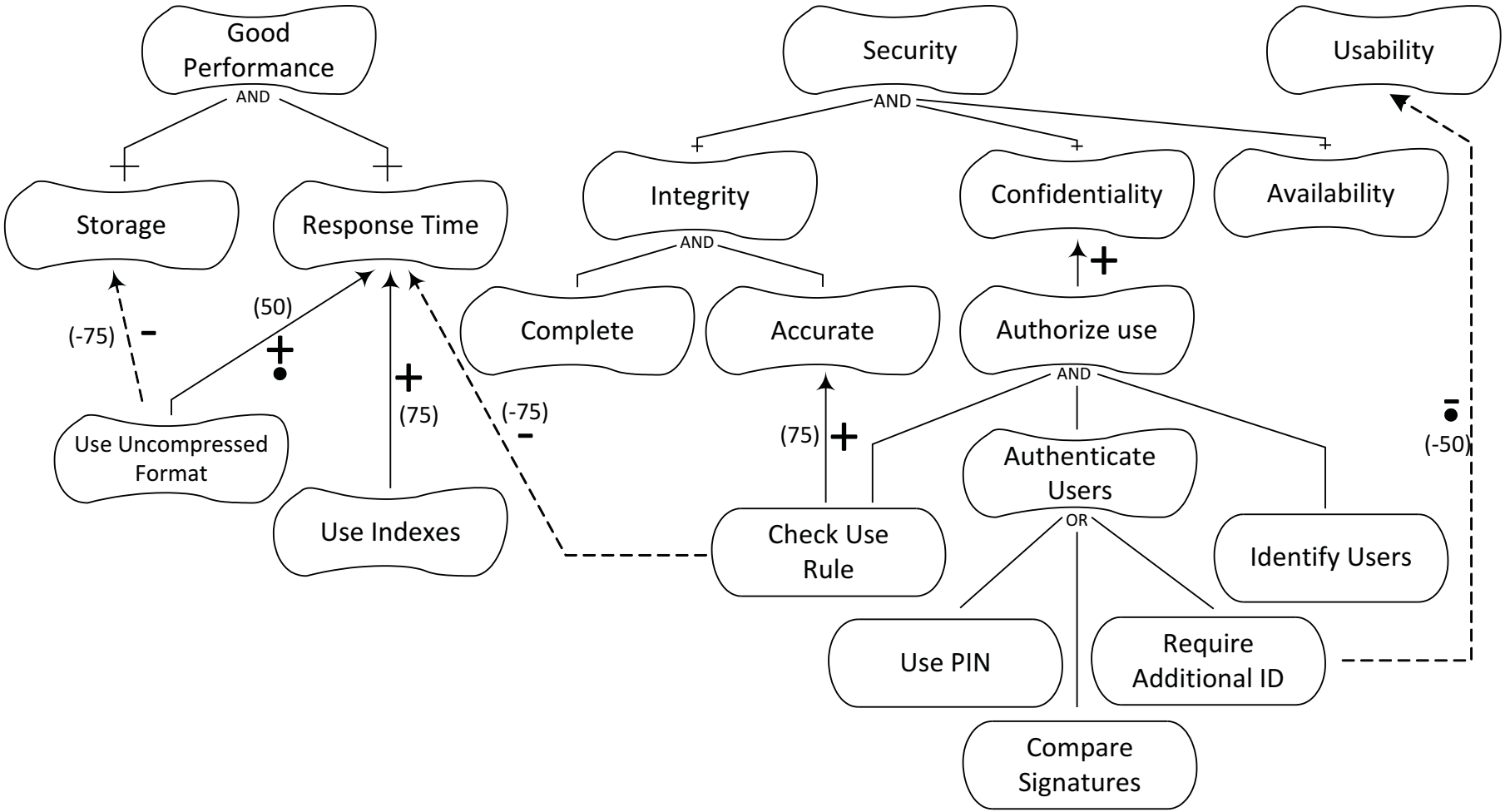
Gestión y Monitorización del Despliegue

1. Incluir soporte para visualizar y analizar configuraciones.
2. Centralizar la gestión de la configuración de la aplicación.
3. Reducir los privilegios por defecto.
4. Proporcionar mecanismos simples para solucionar problemas de seguridad.

Índice

Índice

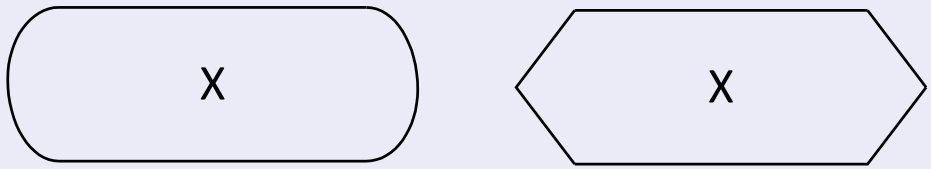
Influencias entre NFRs



Índice

Evaluación de Objetivos/Tareas Hoja

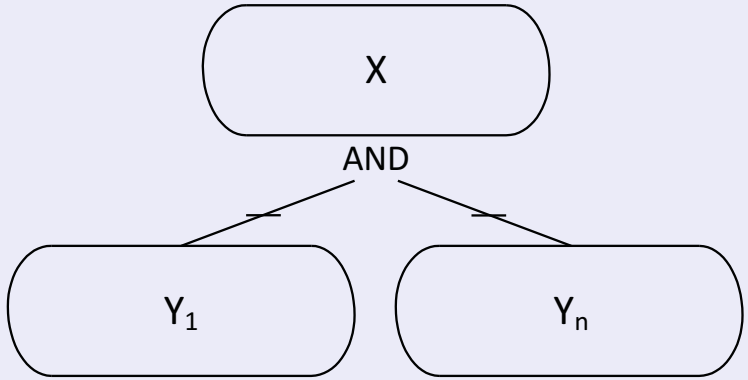
Regla de Evaluación de Objetivos/Tareas Hoja (Hao)



$$V(X) = \begin{cases} 100, \text{si } X \text{ está seleccionado} \\ 0, \text{otro caso} \end{cases}$$

Evaluación de Relaciones AND

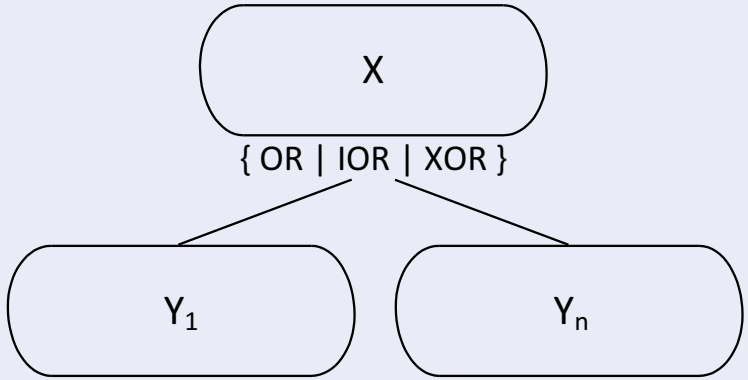
Regla de Evaluación de Relación AND (Hao)



$$V(X) = min_{i=1}^{n} V(Y_i)$$

Evaluación de Relaciones OR

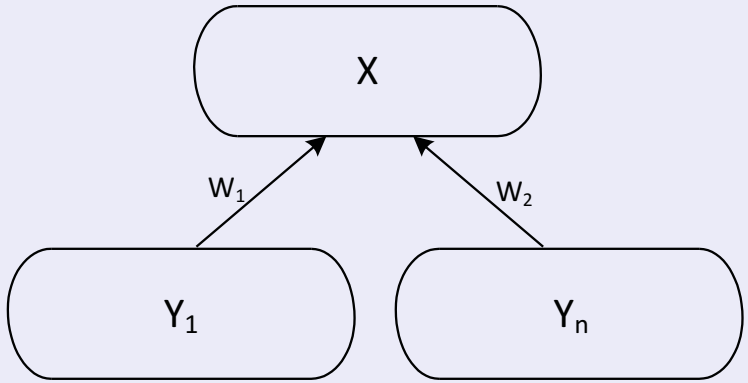
Regla de Evaluación de Relación OR (Hao)



$$V(X) = max_{i=1}^{n} V(Y_i)$$

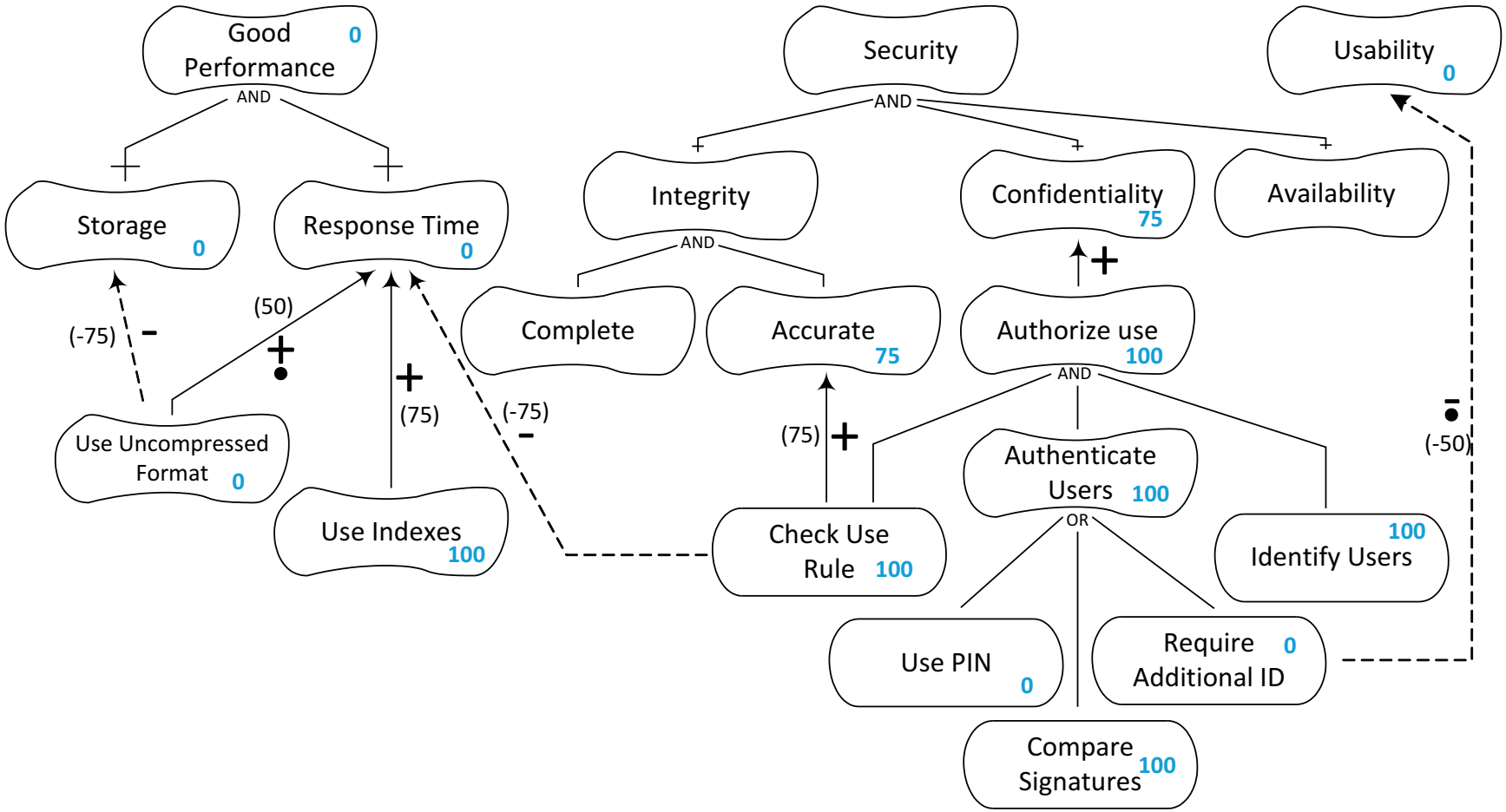
Evaluación de Contribuciones/Correlaciones

Regla de Evaluación de Contribuciones



$$V(X) = max(-100, min(100, \frac{\sum_{i=1}^{n} V(Y_i) * W_i}{100}))$$

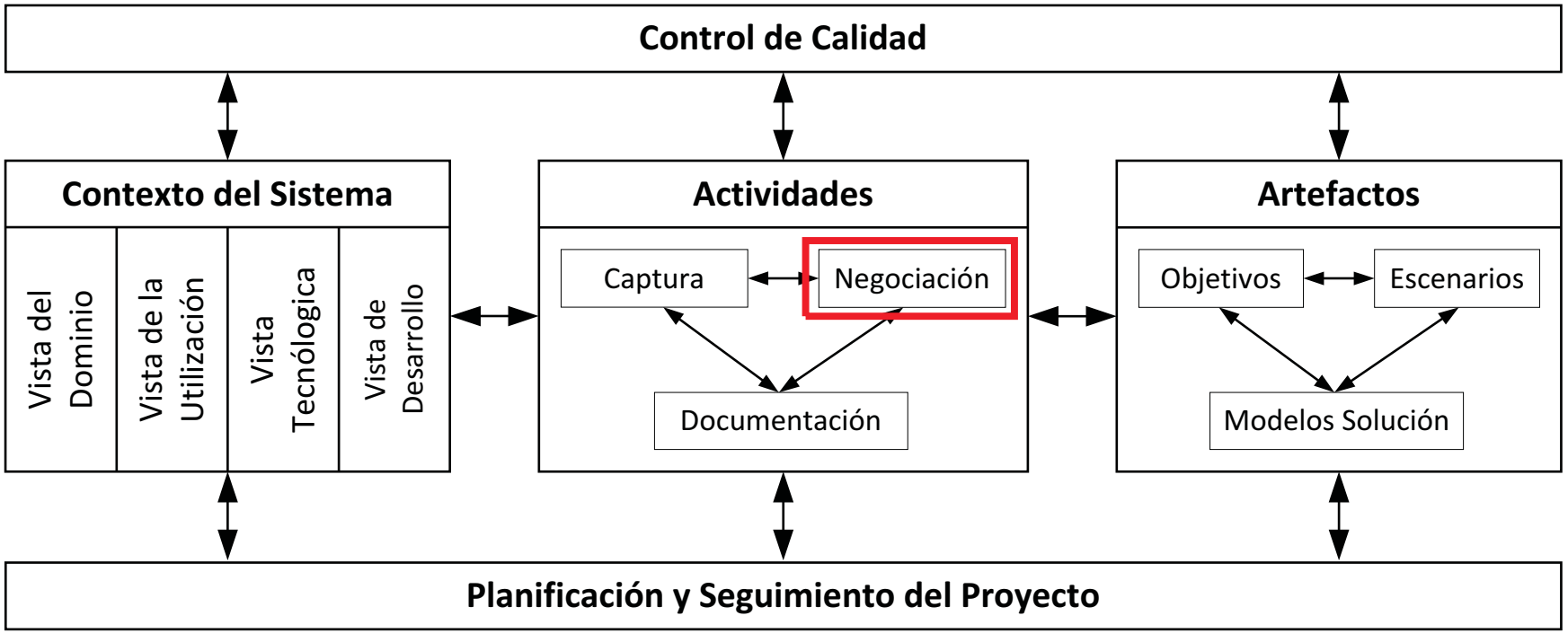
Ejemplo 1: Índices y Verificación de Firmas



Índice

Índice

Proceso de Ingeniería de Requisitos



Índice

Definición de Conflictos

Conflicto en Ingeniería de Requisitos

Un conflicto en Ingeniería de Requisitos aparece cuando las necesidades y deseos de diferentes (grupos de) *stakeholders* con respecto al sistema se contradicen, o si algunas necesidades y deseos no pueden ser tenidos en consideración.

Tipos de Conflictos

1. Conflictos entre los datos recogidos.
2. Conflictos de intereses.
3. Conflictos de valorización.
4. Conflictos de relaciones interpersonales.
5. Conflictos estructurales.

Índice

Técnicas de Resolución de Conflictos

1. Negociación (*Win-Win*).
2. Adoptar nuevas soluciones (*Win-Win*).
3. Decisión Externa.

Índice

Estrategias de Negociación

1. Consider all facts.
2. Plus Minus Interesting

Índice

Índice

¿Qué Tengo que Saber de Todo Esto?

1. Conocer y entender los rasgos diferenciadores de los NFR.
2. Conocer y entender la terminología de la norma ISO 25010.
3. Conocer y entender el papel de los sistemas software en los sistemas sociotécnicos.
4. Comprender la importancia de los NFR en los sistemas sociotécnicos.
5. Conocer y entender la terminología asociada a la especificación de la seguridad de un sistema sw.
6. Saber analizar y especificar requisitos de seguridad de un sistema sw.
7. Saber especificar casos de mal uso y casos de uso de seguridad.
8. Saber especificar, modelar y analizar las influencias entre NFRs.
9. Conocer y entender los principales tipos de conflictos que pueden surgir entre requisitos.
10. Saber aplicar técnicas de negociación para la resolución de conflictos.

Índice

Referencias

Chung, L., Nixon, B. A., Yu, E., and Mylopoulos, J. (1999).
Non-Functional Requirements in Software Engineering.
Kluwer Academic Puvlishers.

Sindre, G. and Opdahl, A. L. (2005).
Eliciting security requirements with misuse cases.
Requirements Engineering, 10(1):34–44.

Sommerville, I. (2010).
Software Engineering.
Addison Wesley, 9 edition.