UNIVERSIDAD TECNICA
FEDERICO SANTA MARIA

Departamento de Informática
Universidad Técnica Federico Santa María

NFRs y Priorización

Ingeniería de Software

Hernán Astudillo & Gastón Márquez
Departamento de Informática
Universidad Técnica Federico Santa María

Contexto [1]

- Comprar un automóvil
  - Se desea un medio de transporte que…
    - Sea barato
    - De una marca conocida
    - Grande/mediano
    - Pueda trasladar persona de un punto A a un punto B
  - Compremos el automóvil…

Contexto [2]

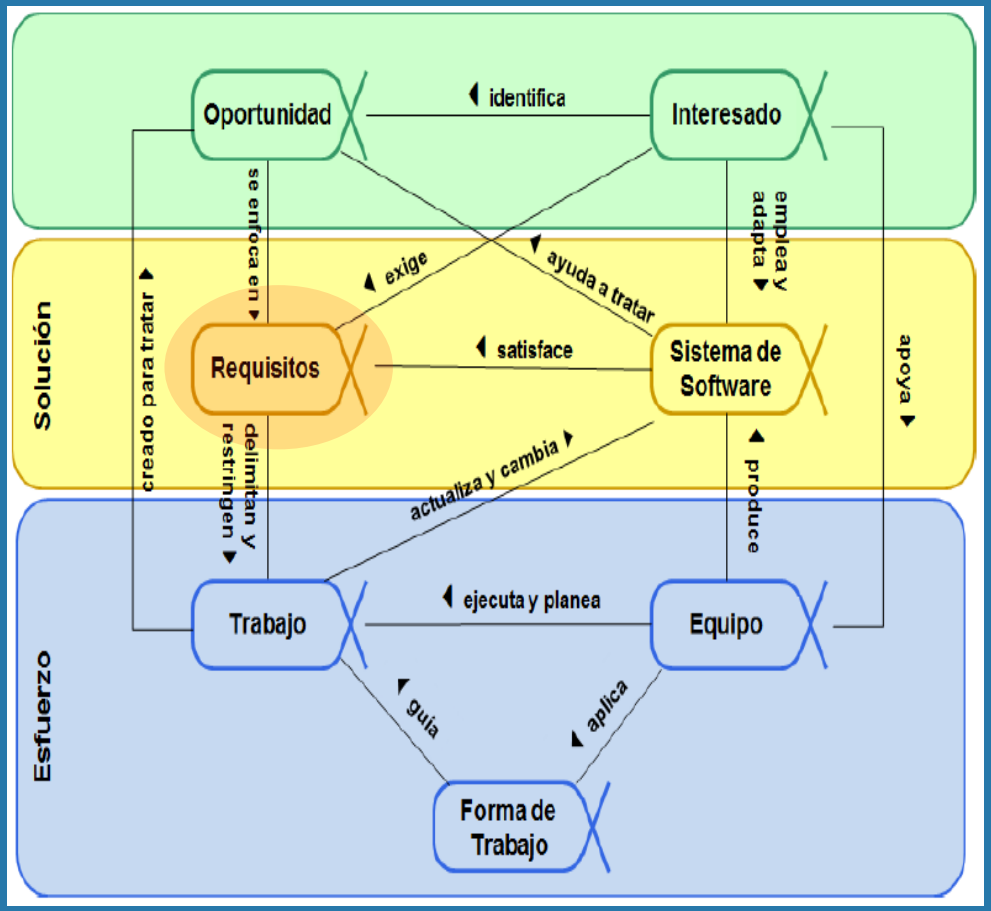
Contexto [3]

- ¿Qué ocurrió?
  - El automóvil
    - Sea barato → sí
    - De una marca conocida → sí
    - Grande/mediano → sí
    - Pueda trasladar persona de un punto A a un punto B → sí
  - A pesar de lo anterior, no nos sentimos satisfechos
  - Existen otras propiedades extras que también se desea que cumpla el automóvil
    - Rendimiento
    - Seguridad
    - Que sea usable
    - Que sea atractivo

Contexto [4]

- ¿Y qué ocurre en ISW?
  - Ocurre lo mismo que la paradoja del automóvil
  - En los proyectos de software, existen propiedades que no están consideradas en las funcionales del software
  - Reflexión
    - En su proyecto semestral de ISW
      - ¿Qué ocurre cuando un proceso se corrompe?
      - ¿Qué hará el sistema si se corta la conexión con el servidor?
      - ¿Qué ocurre si al usuario final no le gusta la interfaz?
      - ¿Qué ocurre si un componente falla? ¿Todo el sistema falla?

Contexto SEMAT



NFR: Non-Functional Requirements

FRs – Requisitos funcionales

- Los FRs describen las funcionalidades o servicios del sistema
- Dependen del tipo de software, las expectativa de los usuarios y el tipo de sistema donde el software será usado
  - Los requisitos funcionales del usuario pueden ser declaraciones de alto nivel de lo que el sistema debe hacer
  - Los requisitos funcionales del sistema describen los servicios del sistema en detalle
- En principio, los requisitos deben ser
  - Completos
  - Consistentes
- Un requisito ambiguo puede ser interpretado de muchas formas por desarrolladores y usuarios…
- …pero la práctica dice que es casi imposible producir requisitos completos y consistentes ☹

NFRs – Requisitos extra-funcionales

- Los NFRs definen las cualidades o atributos de un sistema de software
  - Tiempos máximos, disponibilidad mínima, tasa de falla máxima, etc.
- O pueden ser restricciones sobre…
  - La operación del sistema
  - El proceso de desarrollo
  - Las tecnologías a utilizar
  - etc
- Algunos aspectos de la gestión del proyecto como costos, tiempo, organización también son consideradores como NFRs

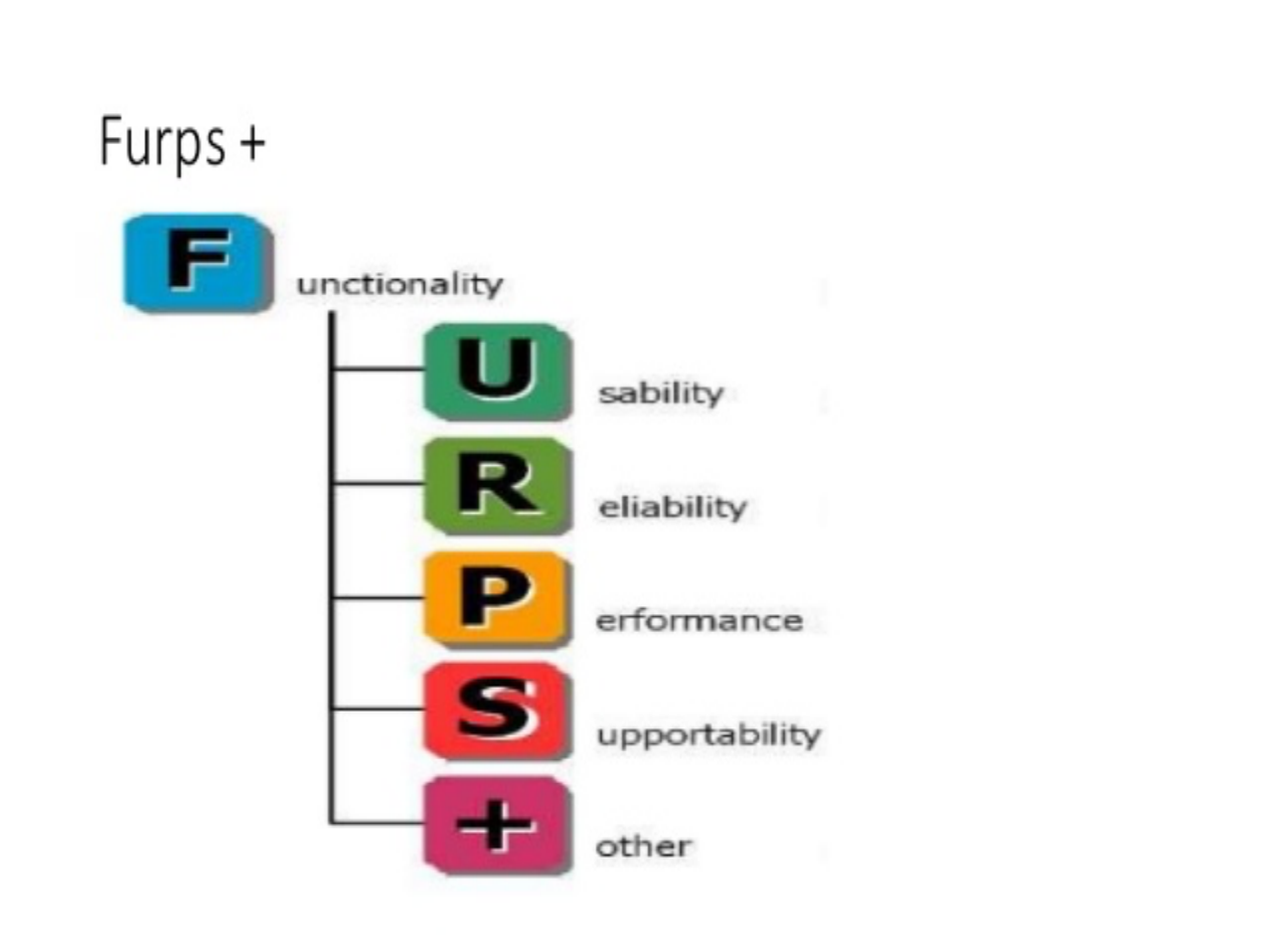
FR vs NFRs [1]

- Las propiedades de un sistema con NFRs a alto nivel pero deben ser convertidas a FRs al bajar nivel
  - P.ej. “El sistema deberá estar disponible 99,9%” -> “El respaldo deberá detectar que el principal está caído y ocupar su lugar”
- La expresión de los requisitos depende de:
  - El nivel de detalle del documento de requisitos
  - La comprensión del dominio de la aplicación del sistema deseado
  - La experiencia de los desarrolladores

Enfoques de NFRs

- ¿Producto versus Proceso?
  - Enfoques orientados al producto
    - Se enfocan en las cualidades del sistema
    - El objetivo es tener una forma de medir el producto una vez que se construye (métricas)
  - Enfoques orientados al proceso
    - Se enfoque en cómo los NFRs pueden ser usado en el proceso de diseño
    - El objetivo es tener una manera apropiada de realizar decisiones de diseño
- ¿Cualitativo o Cuantitativo?
  - Enfoque cuantitativo
    - Encontrar escalas medibles para los atributos de calidad
    - Calcula el grado de cómo el diseño satisface los objetivos de calidad
  - Enfoque cualitativo
    - Estudia varias relaciones entre objetivos de calidad
    - Trade-offs

Clasificación de NFRs: FURPS + [1]

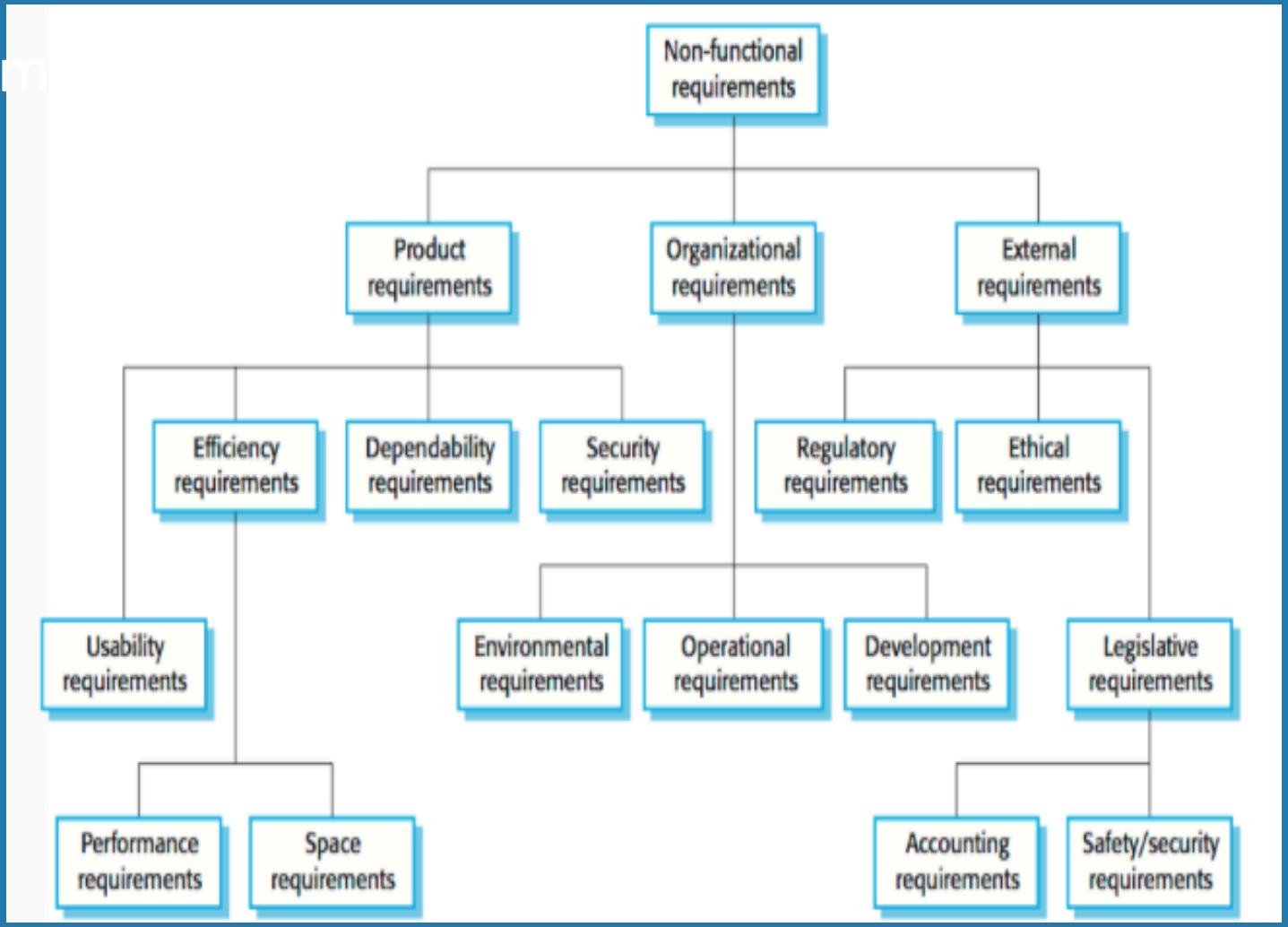


Clasificación de NFRs: FURPS + [2]

- Jacobson [1999] propuso el modelo FURPS+ (originalmente en RUP)
  - F= requisitos Funcionales
  - U= Usabilidad
  - R= confiabilidad (“Reliability”)
  - P= rendimiento (“Performance”)
  - S= Supportability
  - += otros
    - requisitos de Implantación
    - requisitos de Interfaz
    - requisitos de Operación
    - requisitos Legales

Clasificación de NFRs: Sommerville [1]

- [Som



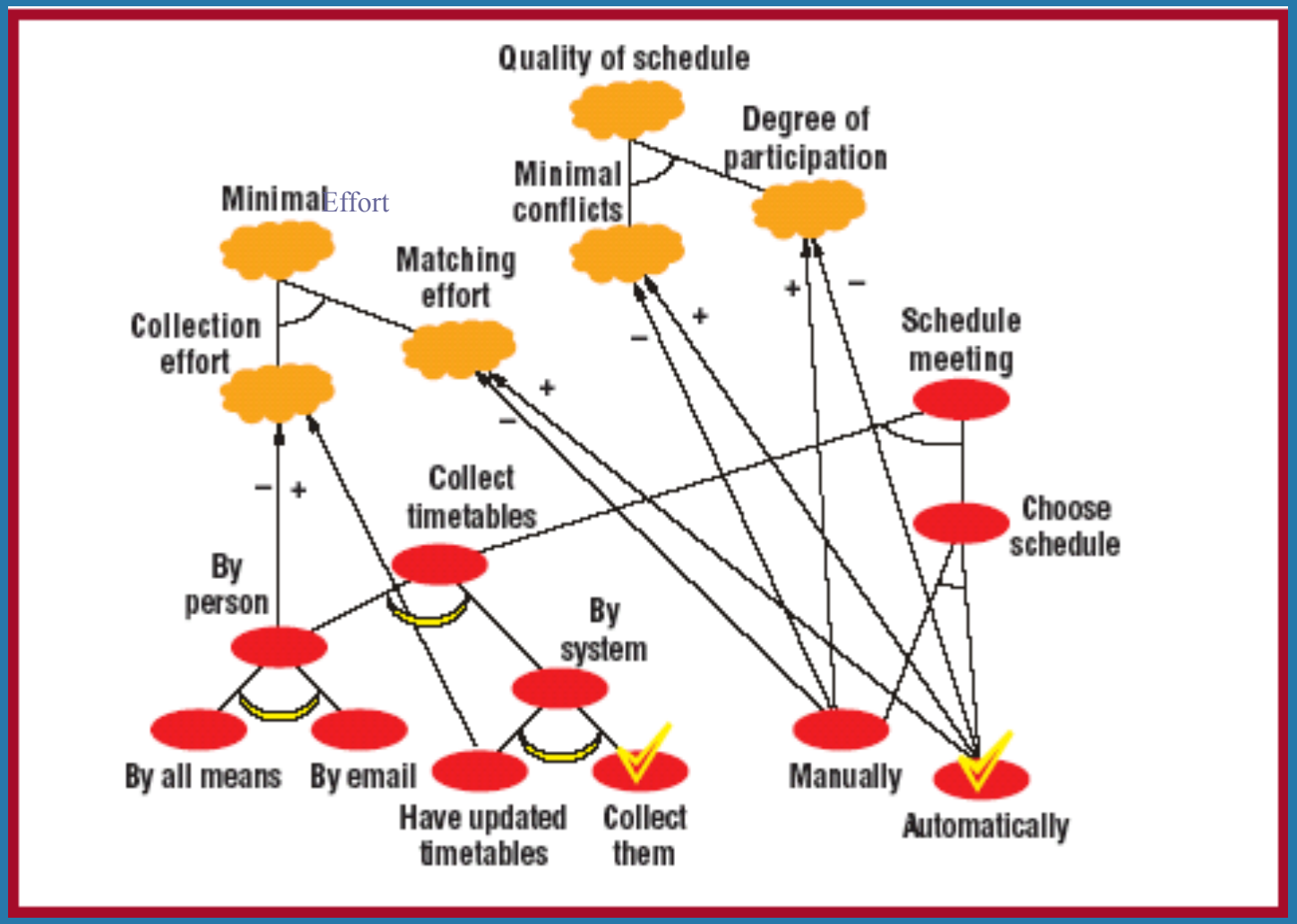
Clasificación de NFRs: Sommerville [2]

- Sommerville [2004] identifica tres tipos de NFRs
  - Requisitos del Producto
    - Especifican cómo debe comportarse el producto entregado
    - P.ej. tiempo de ejecución, confiable, etc
  - Requisitos Organizacionales
    - Reflejan políticas organizacionales y procedimientos
    - P.ej. estándares usados, requisitos de implementación, etc
  - Requisitos Externos
    - Nacen desde factores externos al sistema y al desarrollo
    - P.ej. interoperabilidad, leyes, etc

Softgoals

Identificando requisitos extra-funcionales, perspectiva de objetivos

Motivación


Quality of schedule
Degree of participation
Minimal conflicts
Minimal Effort
Matching effort
Collection effort
Schedule meeting
Collect timetables
Choose schedule
By person
By system
By all means
By email
Have updated timetables
Collect them
Manually
Automatically

NFR Framework [1]

- NFRs: Non-Functional Requirements
  - "NFR Framework": Chung, Nixon, Yu, Mylopoulos (1999)
  - Catálogo de 161 tipos de NFRs en categorías ortogonales
    - Orientación
      - consumidor (e.g. eficiencia, correctitud)
      - técnica (e.g. alcance, completitud)
    - Evaluación
      - producto (¿software posee alguna calidad?)
      - proceso (decisiones de diseño tomadas satisfacen un NFR)
    - Criterio
      - cuantitativo (e.g. performance)
      - cualitativo (e.g. requerimientos de look-and-feel)

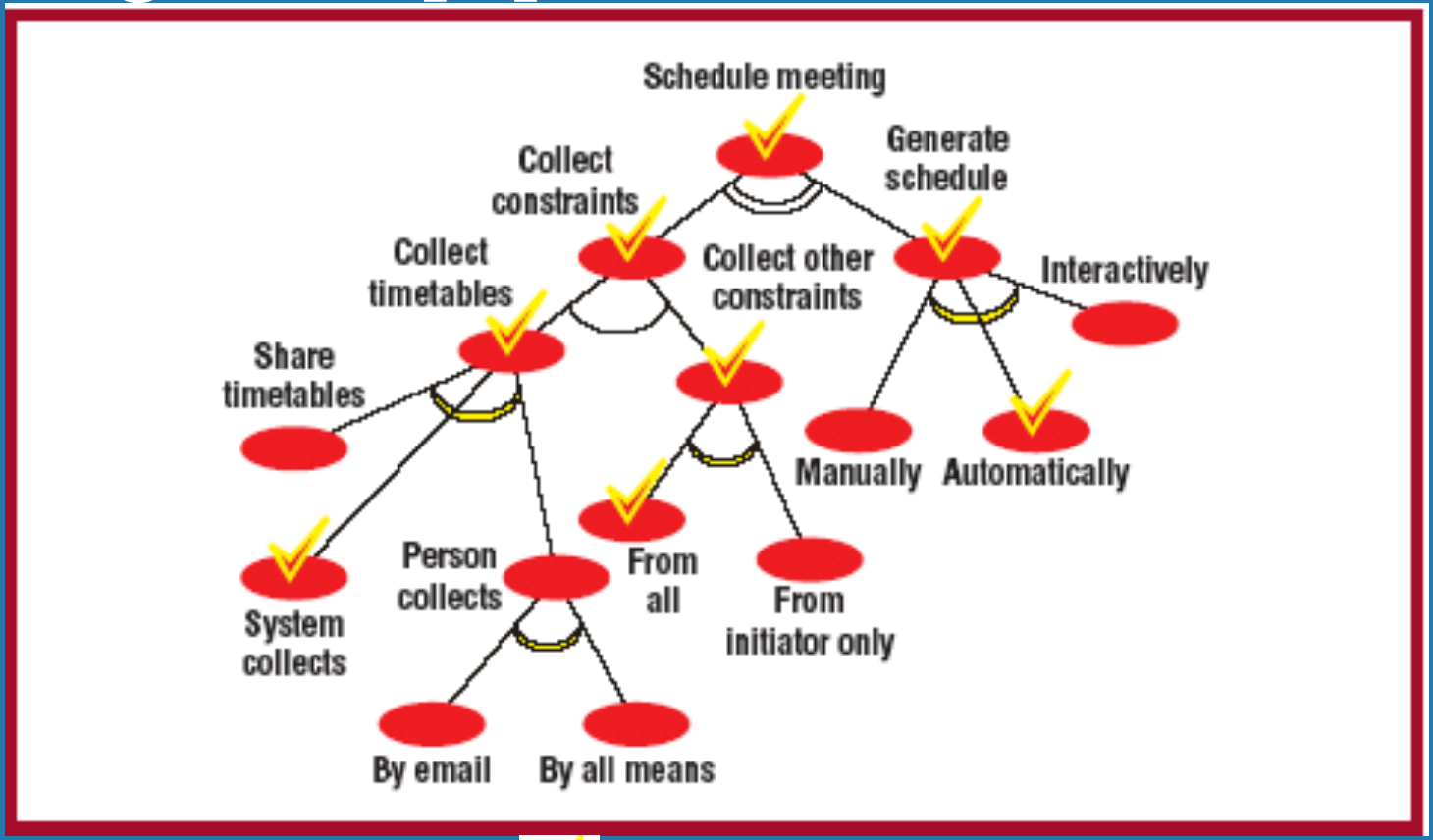
NFR Framework [2]

- Herramienta conceptual: “softgoals”
  - “Goal” == “meta”; “Softgoal” ~= “meta blanda”
  - Objetivos sin un criterio de satisfacción bien definido
- NFR FW es un método para descomponer softgoals en sub-goals más concretos
  - Desarrolladores tienen mejores oportunidades de satisfacer sub-goals concretos

Softgoals [1]

- Objetivo: explorar alternativas, y evaluar factibilidad & conveniencia
  - Y entender y modelar funciones, datos e interfaces
  - Explorar alternativas permite refinar significado de términos y definir las funciones básicas del sistema
- Pasos (input == conjunto de metas funcionales)
  - Analizar Metas alternativas (descomponer cada meta funcional usando AND/OR)
  - Analizar Softgoals – análisis de NFRs/QAs (descomponer cada atributo de calidad en una jerarquía de softgoals)
  - Analizar correlación de Softgoals
  - Analizar correlación de entre Metas y Softgoals
  - Evaluar alternativas y seleccionar metas y softgoals

Softgoals [2]



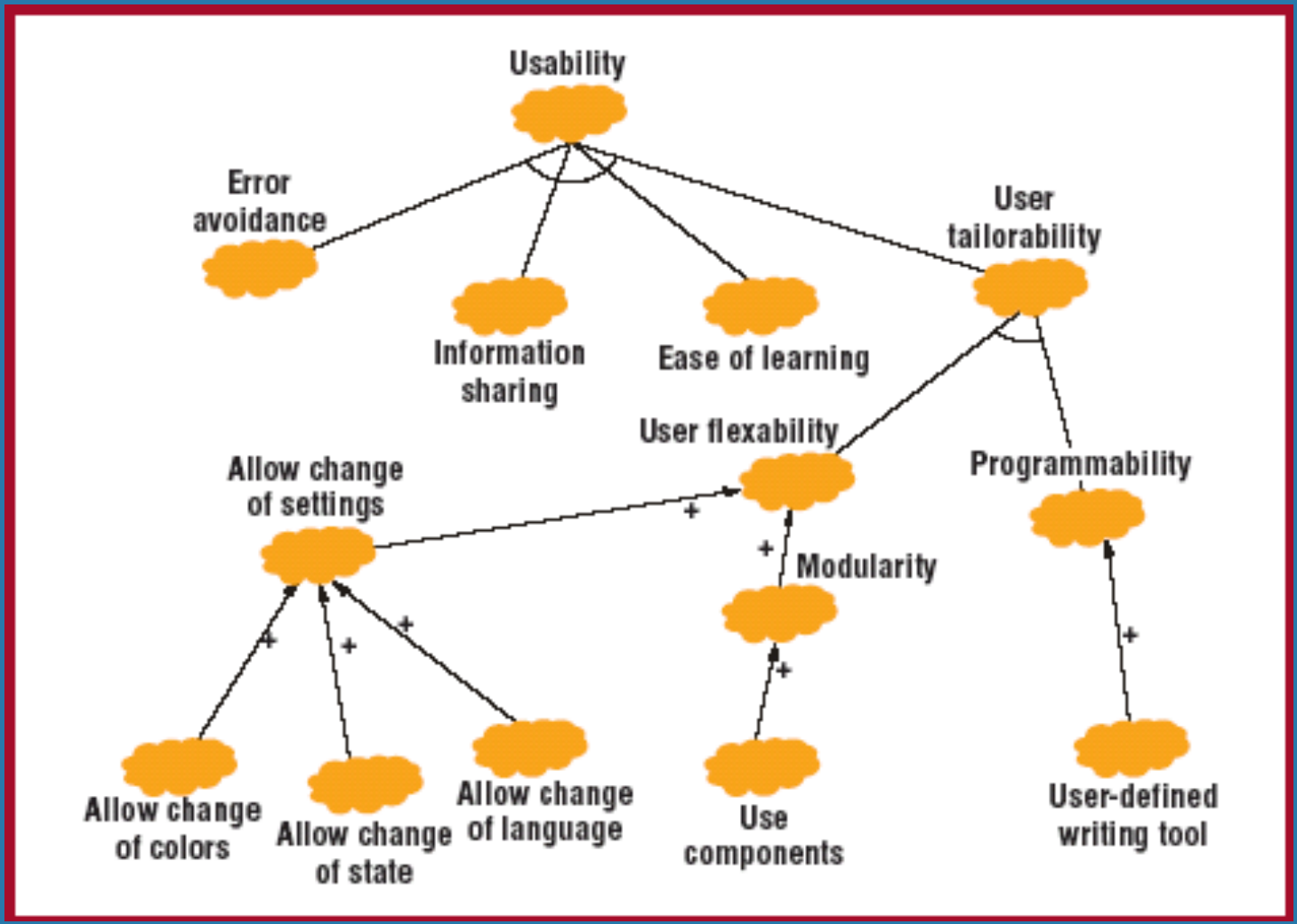
Notation: **Goal** **Root Goals** supported by checked leaf goals;

Softgoal **AND** **OR**

Softgoals [3]

- Softgoal: usualmente NFR/QA, representa metas mal-definidas (ill-defined) y sus interdependencias
- Un softgoal es satisfecho cuando hay suficiente evidencia positiva a favor de él y poca evidencia negativa en contra
- Se puede hacer jerarquías de softgoals indicando qué se podría hacer para satisfacer (o apoyar) un softgoal dado

Softgoals [4]

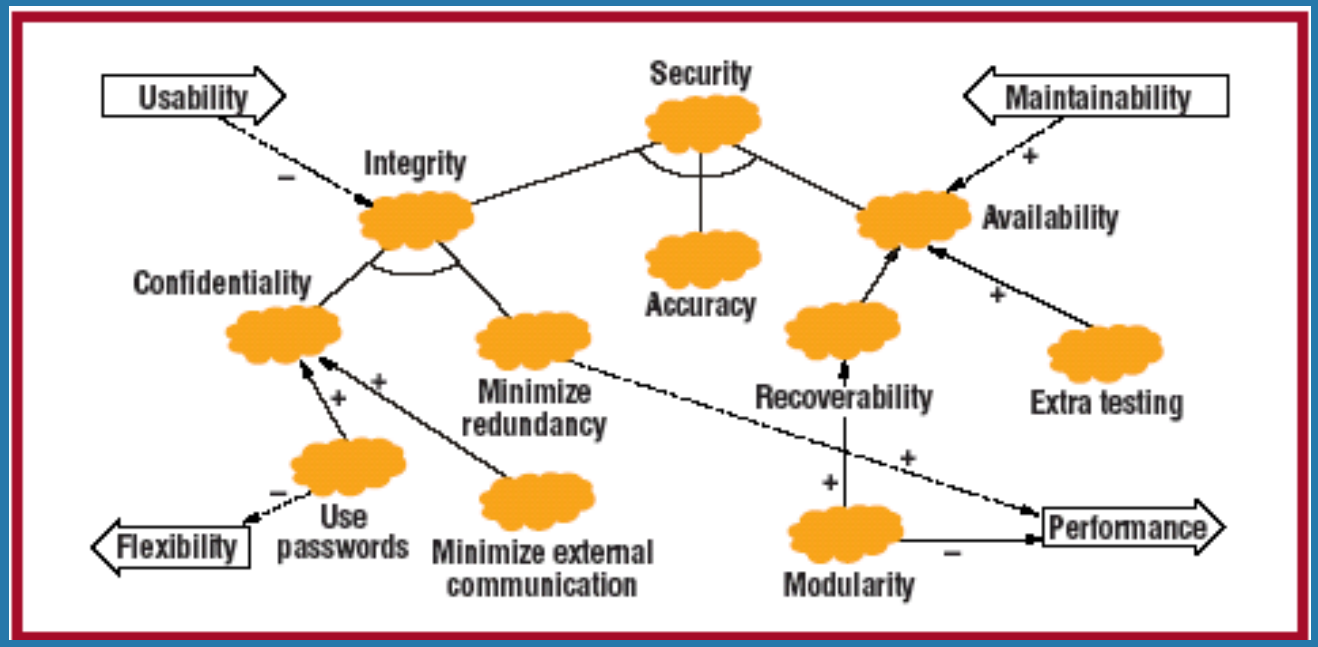

Usability
Error avoidance
Information sharing
Ease of learning
User tailorability
User flexability
Allow change of settings
Modularity
Programmability
Allow change of colors
Allow change of state
Allow change of language
Use components
User-defined writing tool
+

Softgoals [5]

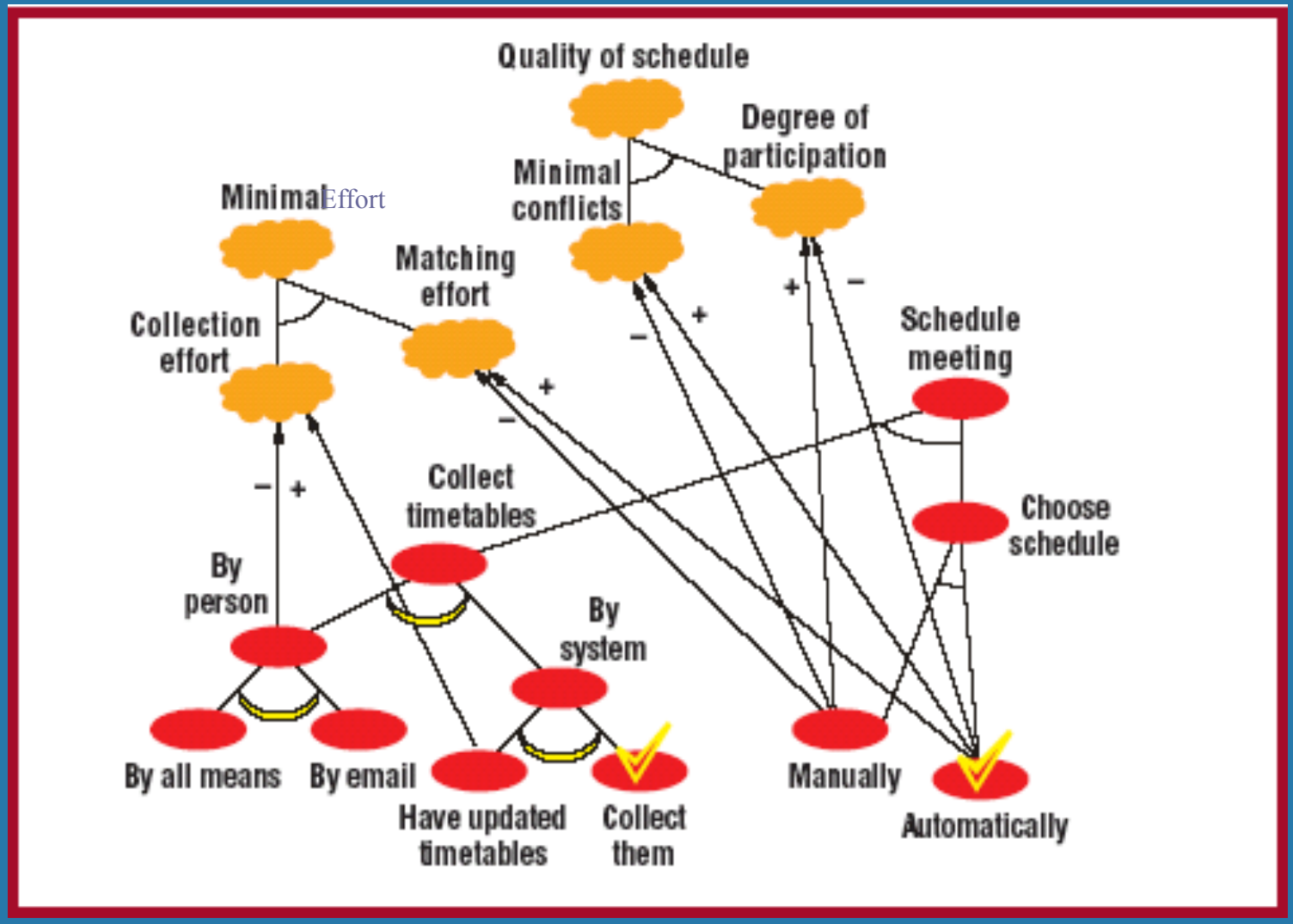
- Hay algunas descomposiciones genéricas a factores de calidad más finos para las calidades generales de software
  - Ejemplo: softgoal “desempeño rápido” puede ser descompuesto en 3 softgoals:
    - Minimizar UI
    - Usar algoritmos eficientes
    - Asegurar adecuada potencia de CPU
- Descomposiciones específicas de tarea/proyecto puede incluso ser usadas sobre acuerdos entre stakeholders

Softgoals [6]

- Metas de calidad frecuentemente están en conflicto
- Análisis de correlación permite descubrir relaciones positivas/negativas entre softgoals



Softgoals [7]


Quality of schedule
Degree of participation
Minimal conflicts
Minimal Effort
Matching effort
Collection effort
Schedule meeting
Collect timetables
Choose schedule
By person
By system
By all means
By email
Have updated timetables
Collect them
Manually
Automatically
+
–

Temas relacionados

- «Computing with words»
- Representación con valores difusos
- Ausencia de perfil UML para NFRs
- Ejemplo de NDR & browser de softgoals
- Preguntas
  - Relación tamaño-sistemas v/s árbol.utilidad /NFR/ATAM
  - NFR FW tiene demasiadas NFrs (¿incluyen métricas?)

Recursos

- Non-Functional Requirements in Software Engineering
  - Lawrence Chung, Brian A. Nixon, Eric Yu, John Mylopoulos
  - Springer (1999) ISBN-13: 978-0792386667
- Tutorial: “NFR Framework”
  - http://www.utdallas.edu/~supakkul/tools/RE-Tools/NFR-Framework.html
- “Creating an NFR Framework Softgoal Interdependency Graph” (video)
  - www.youtube.com/watch?v=HNeZOs19cws

Árboles de Utilidad

Priorizando escenarios (funcionales y extra-funcionales)

Árbol de utilidad de atributos de calidad

- Levantar requisitos para “utilidad” del sistema
  - Funcionalidades
  - Propiedades (rendimiento, disponibilidad, seguridad, modificabilidad, usabilidad…)
- Especificados hasta nivel de escenarios
- Priorizados

Ej.: árbol de utilidad

		Latencia	Minimizar acceso DB a 200ms
	Rendimiento		Entregar video en tiempo real
Utilidad		Throughput	Maximizar throughput de autenticación
		Confidencialidad	BD clientes 99.9% up
	Seguridad		
		Integridad	Link Transbank up 99.5%

Escenarios

- Casos de uso & user stories
  - (conocidos)
- Escenarios de crecimiento
  - Cambios típicos anticipados
  - (Ej.: agregar acceso Servicios Web)
- Escenarios exploratorios
  - Exponen los límites o condiciones extremas del diseño actual
  - (Ej.: agregar interfaz con realidad virtual)

Priorización

- Atributos en el árbol de utilidad son priorizados según…
  - importancia de cada escenario para el éxito del sistema
  - grado de dificultad para lograrlo
- Escala: A (alta), M (media) , B (baja)

Ej.: árbol de utilidad priorizado

		Latencia	Minimizar acceso DB a 200ms [M,B]
	Rendimiento		Entregar video en tiempo real [A,M]
Utilidad		Throughput	Maximizar throughput de autenticación [M,M]
		Confidencialidad	BD clientes 99.9% up [B,A]
	Seguridad		
		Integridad	Link Transbank up 99.5% [B,A]

Requisitos para sistemas críticos

Sistemas críticos

- NFRs juegan un importante rol en los sistemas críticos
- Un sistema es crítico si una "falla" produce un daño significativo en la economía, físico, a seres humanos, organizaciones, gente…
  - Daño inaceptable
- Los principales NFRs que se debe considerar en un sistema crítico son [Cremers et al., Organizational Requirements Engineering]:
  - Reliability (confiabilidad)
  - Performance (rendimiento)
  - Securidad
  - Usabilidad

Confiabilidad [1]

- Confiabilidad ("reliability") es la capacidad de un sistema para realizar sus funciones requeridas bajo las condiciones establecidas por un período específico de tiempo
  - Típicamente, la capacidad de manejar los tiempos de ejecución
- Se puede considerar bajo dos puntos de vista
  - Disponibilidad: Fracción de tiempo que el usuario necesita usar el sistema y está disponible
  - Tasa de falla: frecuencia con que el sistema falla por unidad de tiempo
- No son recíprocos: la disponibilidad depende de la tasa de falla y tiempo de recuperación

Confiabilidad [2]

- Hay atributos similares
  - "Dependability"
    - Habilidad para entregar el servicio que puede ser justificablemente confiable a los usuarios finales
  - Reputación
    - La opinión general del público a través de una persona o un grupo de gente

Rendimiento [1]

- Se preocupa de la velocidad de las operaciones de un sistema
- Algunos tipos de Rendimiento
  - Requisitos de respuesta: cuán rápido el sistema reacciona cuando un usuario ingresa una entrada
  - Requisitos de producción: lo mucho que se demora un sistema dentro de un periodo de tiempo
  - Requisitos de disponibilidad: si el sistema está disponible cuando un usuario final realiza una petición
- Atributos similares
  - Escalabilidad: Capacidad de un sistema para crecer su producción según crecen sus recursos

Rendimiento [2]

- Algunos ejemplos
  - El servicio del sistema X debe tener una disponibilidad de 999/1000 o 99%.
  - El sistema Y debe procesar al menos 8 transacciones por segundo
  - El sistema Z no deberá demorarse más de 30 segundos al momento de procesar productos.

Seguridad

- Los requisitos de seguridad aseguran…
  - Que el acceso no autorizado al sistema y los datos no sea permitido
  - Asegurar la integridad del sistema de daños accidentales o maliciosos
- Algunos ejemplos de requisitos de seguridad
  - “Los permisos de acceso para los datos del sistema sólo deben ser cambiados por el administrador del sistema”
  - “Todo el sistema de software debe ser respaldado cada 24 horas y las copias deben ser almacenadas en un lugar seguro que no esté en el mismo edificio del sistema”
  - “Todas las comunicaciones externas entre el servidor de datos del sistema y los clientes deben ser encriptadas”

Usabilidad

- Usabilidad mide cuán fácil es para un usuario aprender a operar, preparar entradas e interpretar salidas de un sistema o componente
- La usabilidad depende de (entre otros)…
  - Manuales de usuario bien estructurados
  - Información de mensajes de error
  - Facilidades de ayuda
  - Interfaces de usuario bien definidas

Prueba de NFRs

Prueba de NFRs

- Las pruebas de NFRs se concentran en aspectos no funcionales del sistema (características no relacionadas con la funcionalidad)
  - ¿Cuál es el rendimiento bajo situaciones normales?
  - ¿Cómo se comporta cuando muchos usuarios ingresan al mismo tiempo?
  - ¿Puede manejar el stress?
  - ¿Cuán segura es?
  - ¿Cuán fácil es portarla a otra plataforma?
- Pregunta clave
  - ¿Aporta este tipo de pruebas a la calidad de la aplicación?
  - Las prueba respaldan al cliente que el sistema de software cumple con los requisitos establecido al inicio del proyecto

Prueba de Rendimiento

- Evalúa el rendimiento global del sistema
- Algunos elementos claves a considerar
  - Validar que el sistema cumpla con los tiempos de respuesta
  - Evaluar los elementos significativos de la aplicación
  - Realizar pruebas de integración
  - Realizar pruebas a partes del sistema
- Beneficios
  - Determinar la velocidad, escalabilidad y estabilidad del sistema
  - Saber si el usuario está satisfecho con el rendimiento del sistema
  - Identificación de problemas entre rendimiento versus expectativas
  - Optimización de esfuerzo

Prueba de Carga

- Evalúa si el rendimiento del sistema es el esperado en condiciones normales y esperadas
- Algunos elementos claves a considerar
  - Validar el rendimiento del sistema cuando usuarios concurrentes acceden a la aplicación
  - Verificar si los tiempos de respuestas son los establecidos al realizar el punto anterior
  - Al momento de realizar este testing, se deben utilizar datos reales
- Beneficios
  - Tomar las acciones pertinentes al momento de aumentar la producción del sistema
  - Detectar problemas de concurrencia

Prueba de Stress

- Evalúa si el rendimiento del sistema es como se esperaba cuando está escaso de recursos
- Algunos elementos claves a considerar
  - Probar el sistema con poca memoria o poco espacio en el disco en los servidores del cliente para obtener problemas que no se pueden encontrar en condiciones normales
  - Múltiples transacciones en el mismo dato
  - Múltiples clientes conectados a los servidores
- Beneficios
  - Se determina si los datos se pueden corromper si existe overstressing del sistema
  - Ayuda a determinar qué tipos de fallas son más abordables

Prueba de Capacidad

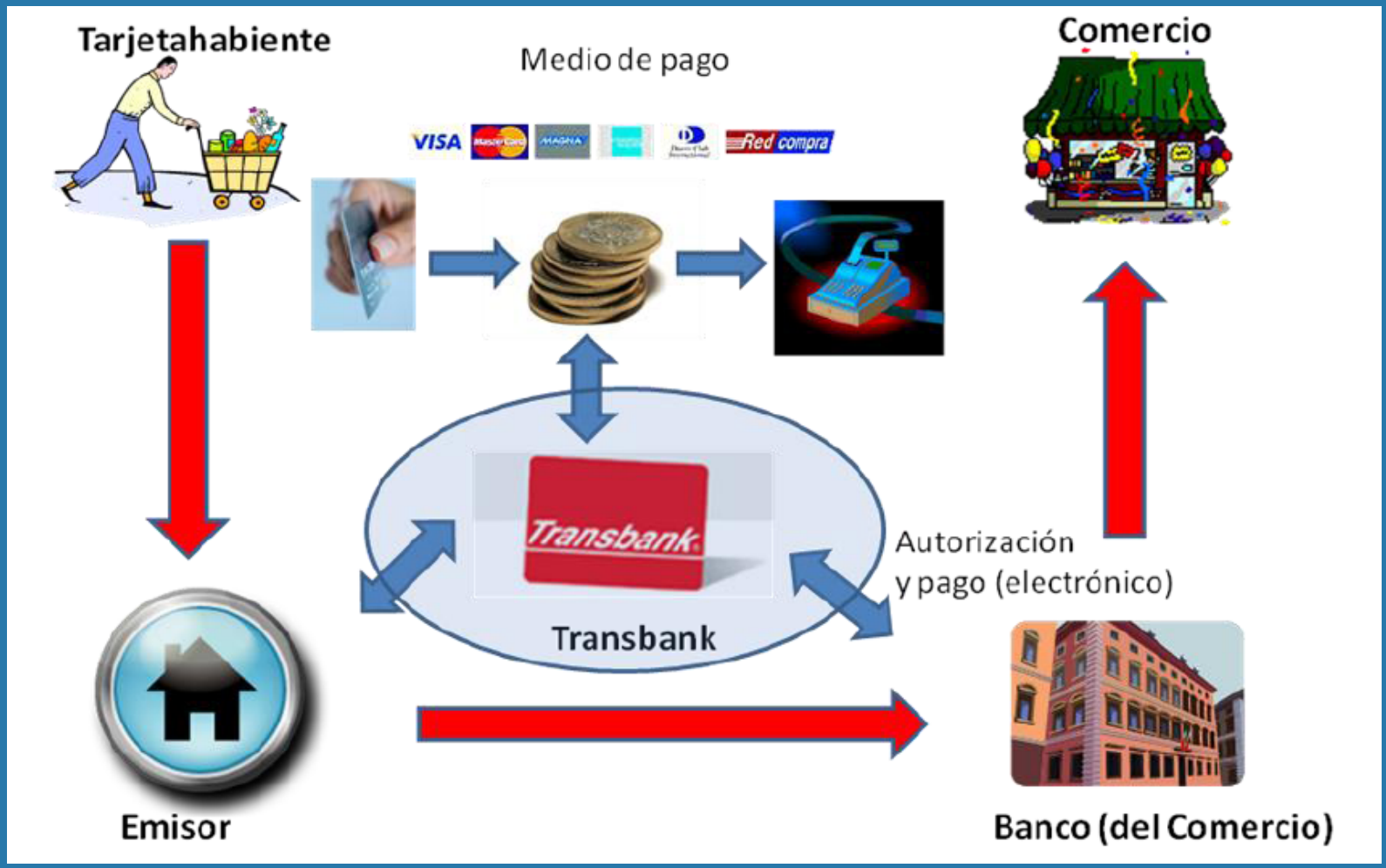
- Determina cómo muchos usuarios y/o transacciones dados al sistema serán abordados y cómo se cumplirá el rendimiento establecido
- Algunos elementos claves a considerar
  - Se debe considerar el aumento del volumen de usuarios y datos
  - Las pruebas de capacidad ayudan a la escalabilidad del sistema
- Beneficios
  - Provee información sobre cómo el trabajo del sistema puede ser manejado
  - Determina el uso actual y la capacidad existente del sistema
  - Provee información necesaria para ayudar en el plan de capacidad del sistema

Prueba de NFRs: recursos

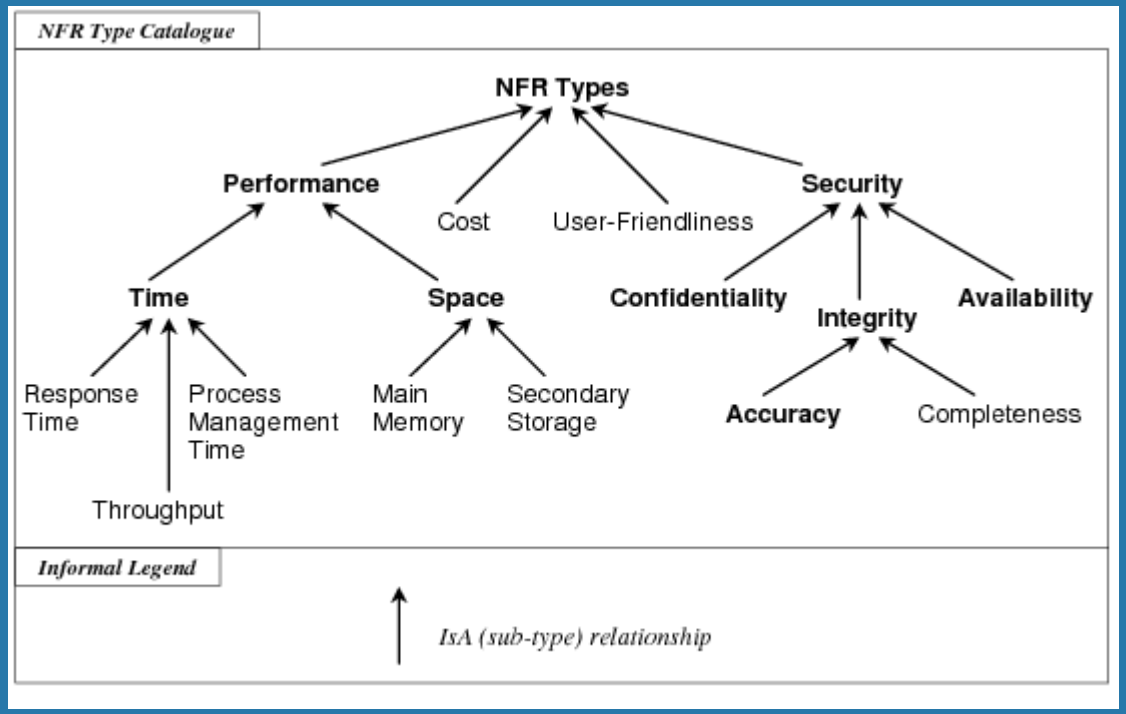
- Ver más pruebas utilizadas por la industria
  - [McEwen, IBM, 2004]
  - [Meier et al., Microsoft, 2007]

Extra: Ejemplo Softgoals

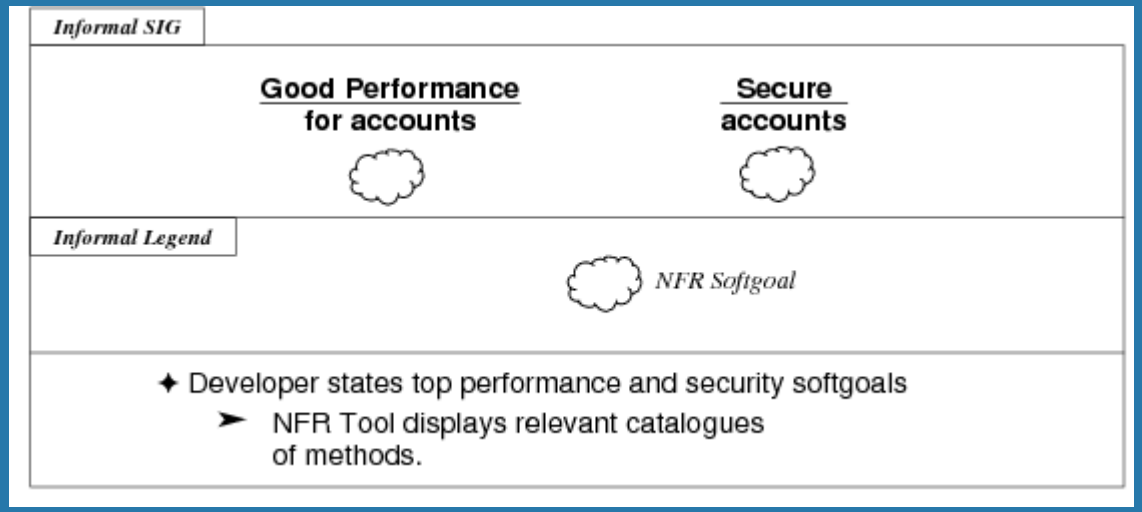
Ejemplo


Tarjetahabiente
Medio de pago
Comercio
VISA
Red compra
Transbank
Transbank
Autorización
y pago (electrónico)
Emisor
Banco (del Comercio)

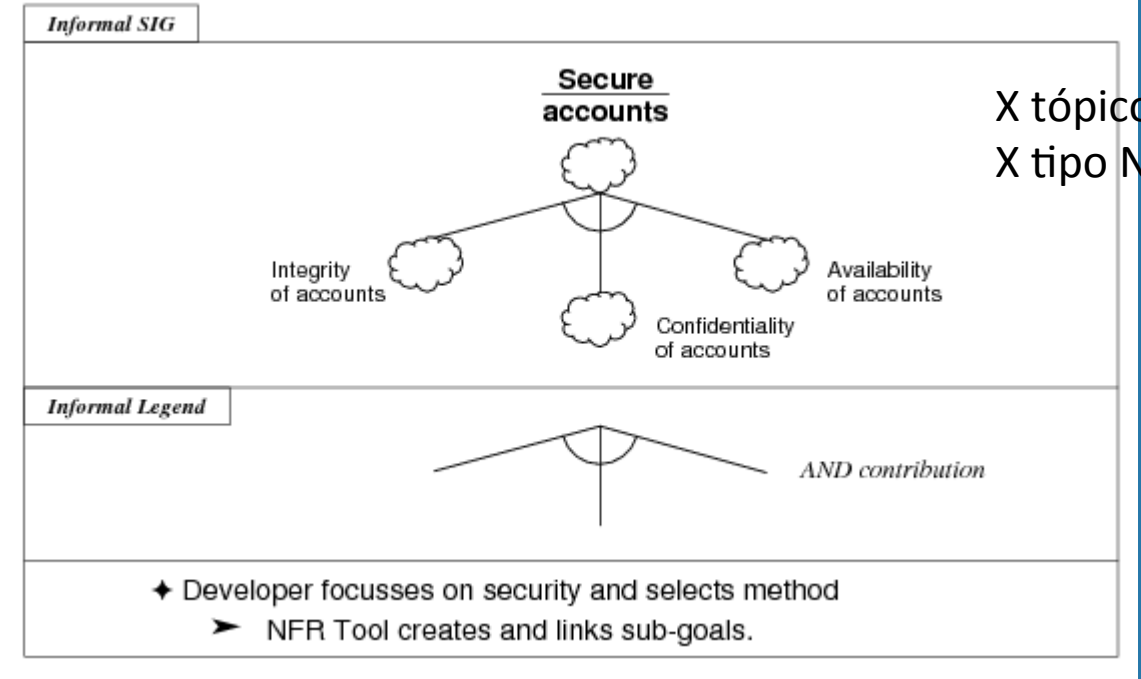
Catálogo de algunos NFRs



Identificar NFRs

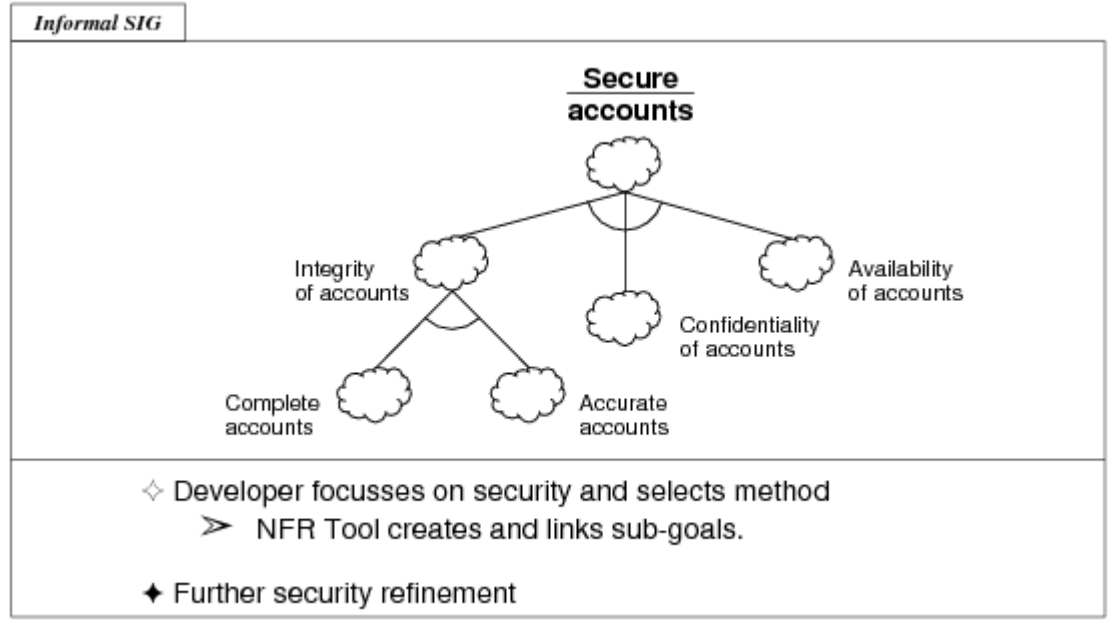


Descomponer softgoals

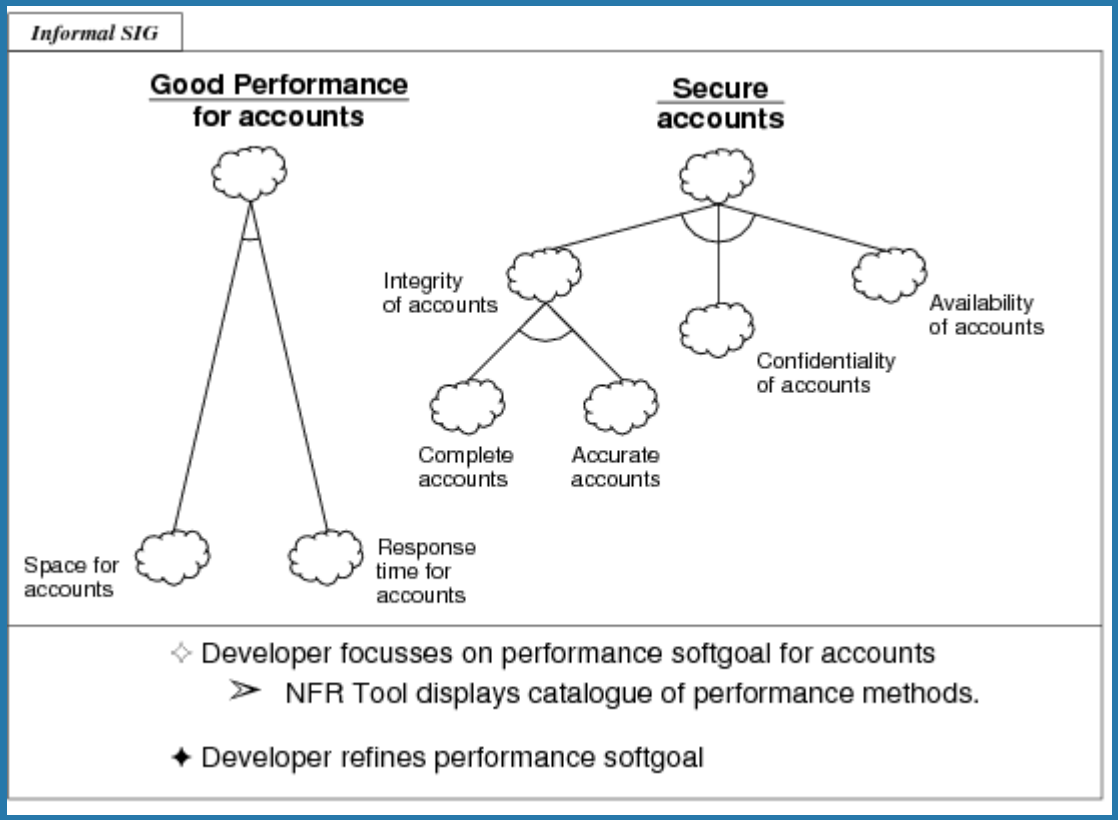


X tópico
X tipo NFRs

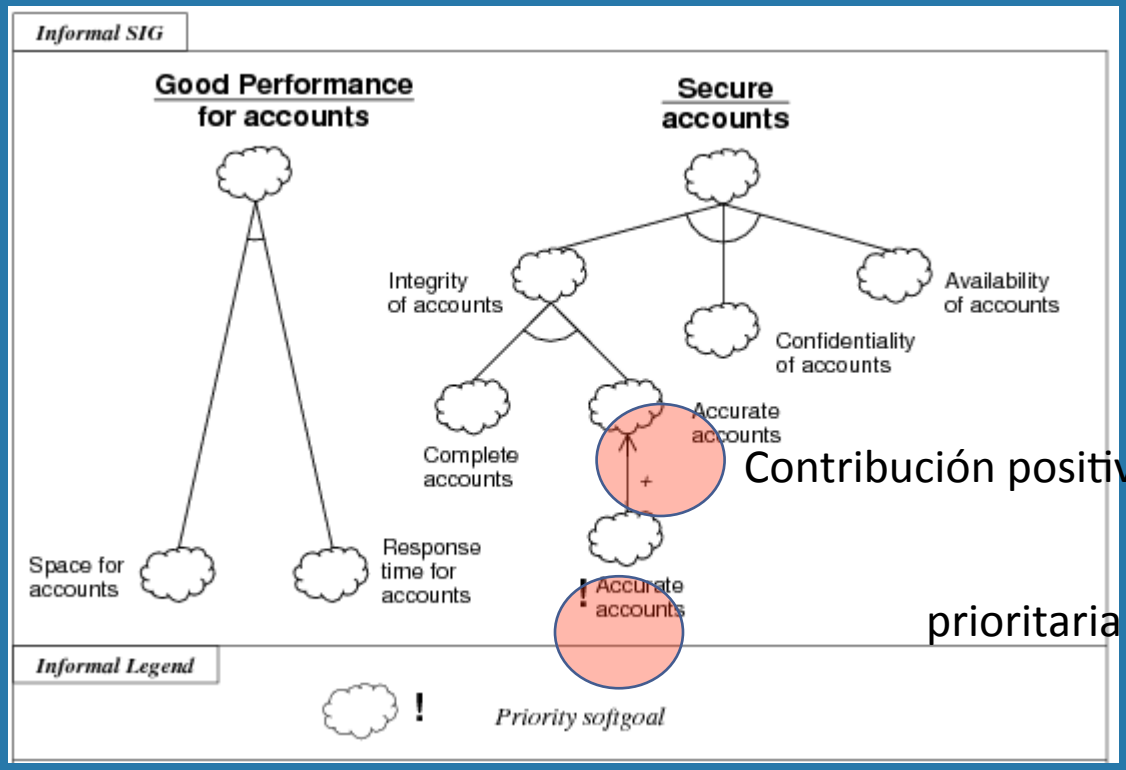
Descomponer softgoals



Descomponer softgoals



Registrar prioridades



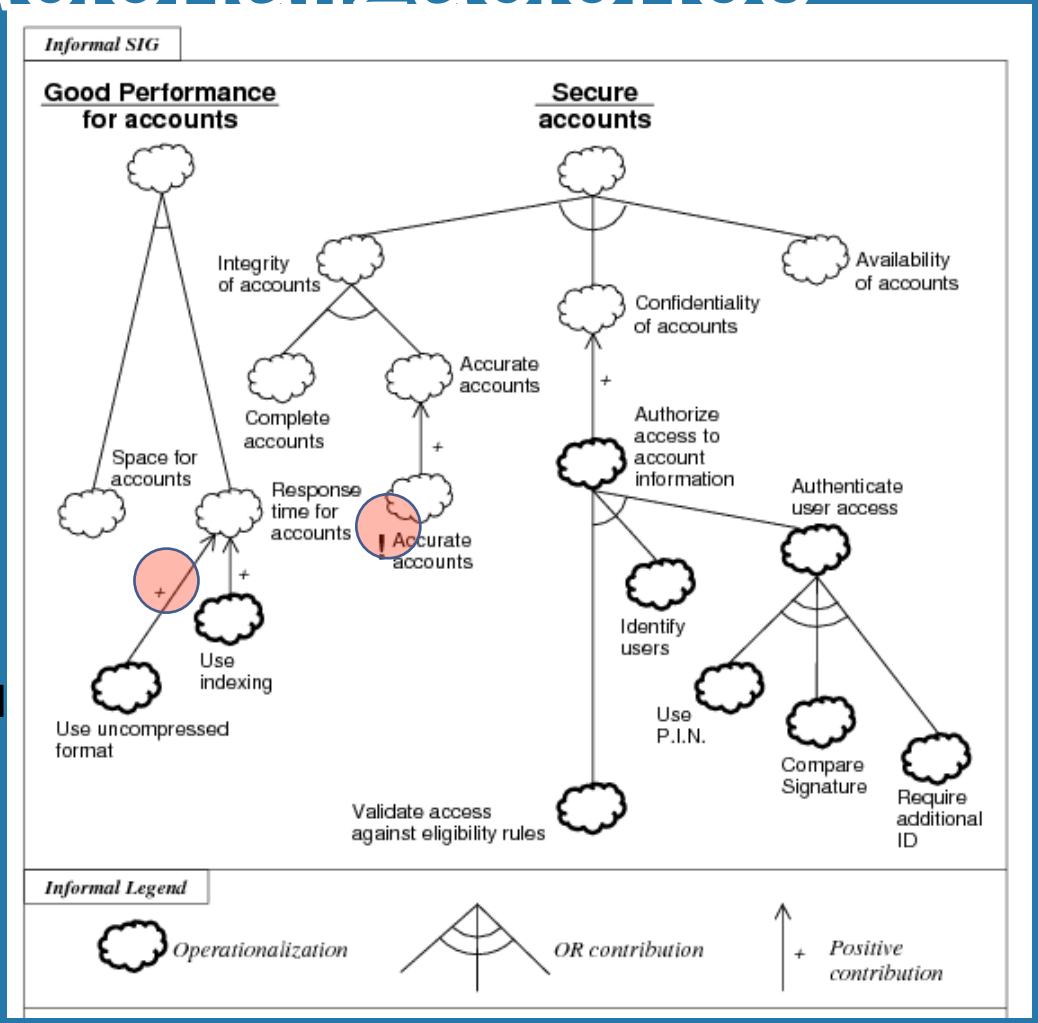
Contribución positiva.

prioritaria

Registrar prioridades

- Registra tradeoffs (“compromisos”) entre softgoals
  - Autorización rápida de ventas con tarjeta de crédito
  - Determinación rápida de los puntos más para los que usan tarjetas Cencosud

Identificar operacionalizaciones



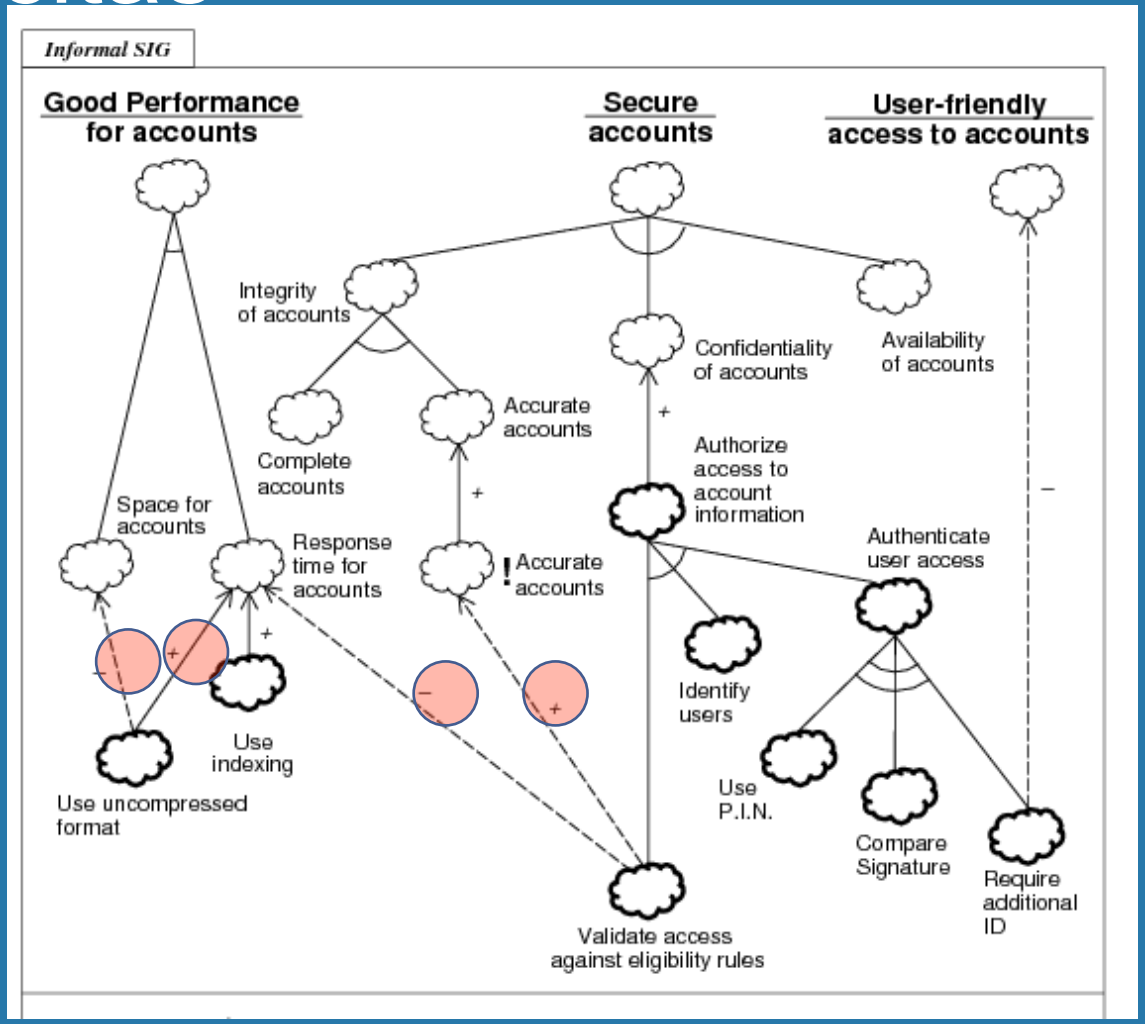
Impacto negativo en rendimiento del espacio

Cable a tierra!

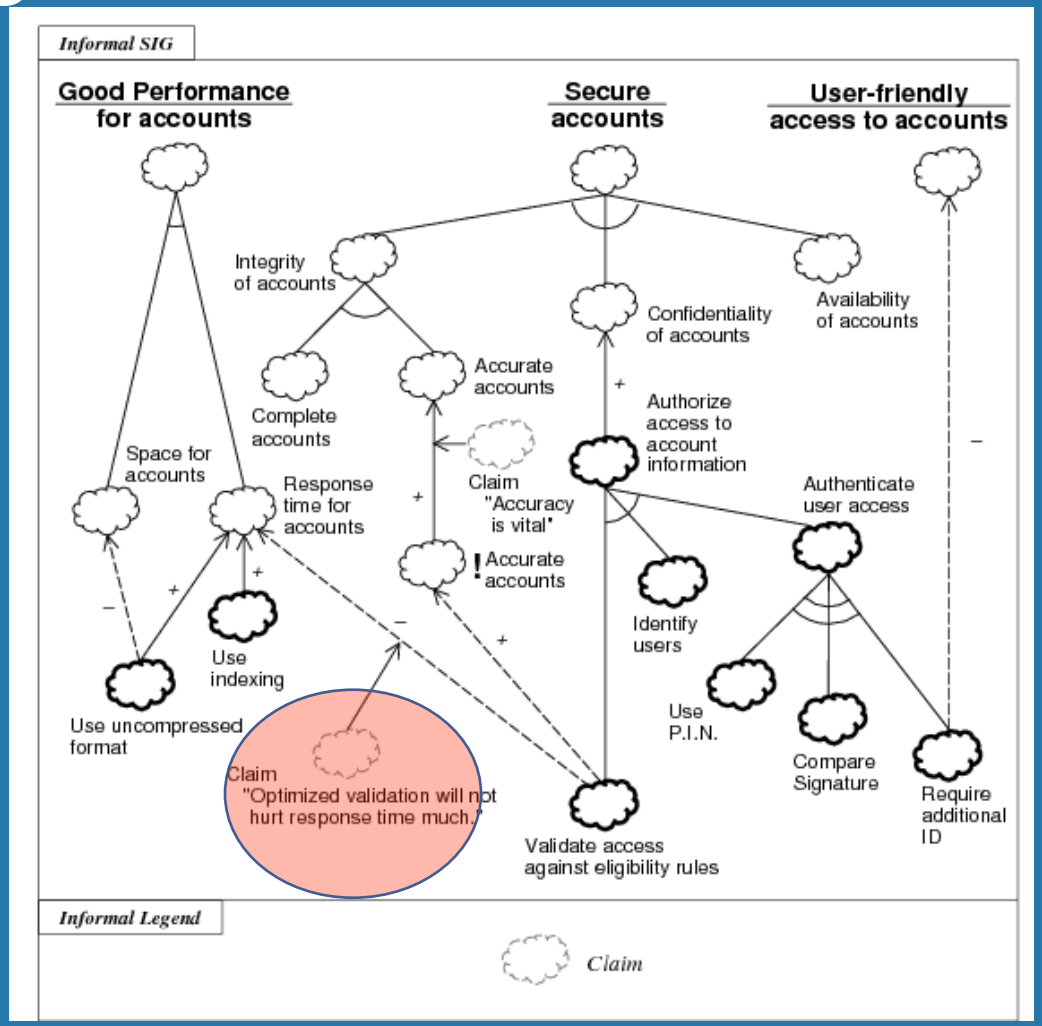
Explicitar interdependencias implícitas

- En este caso:
  - “Formato descomprimido” ayuda al tiempo de respuesta pero perjudica el rendimiento de espacio
  - “Requerir ID adicional” ayuda finalmente a confidencialidad de las cuentas pero perjudica la simplicidad de acceso

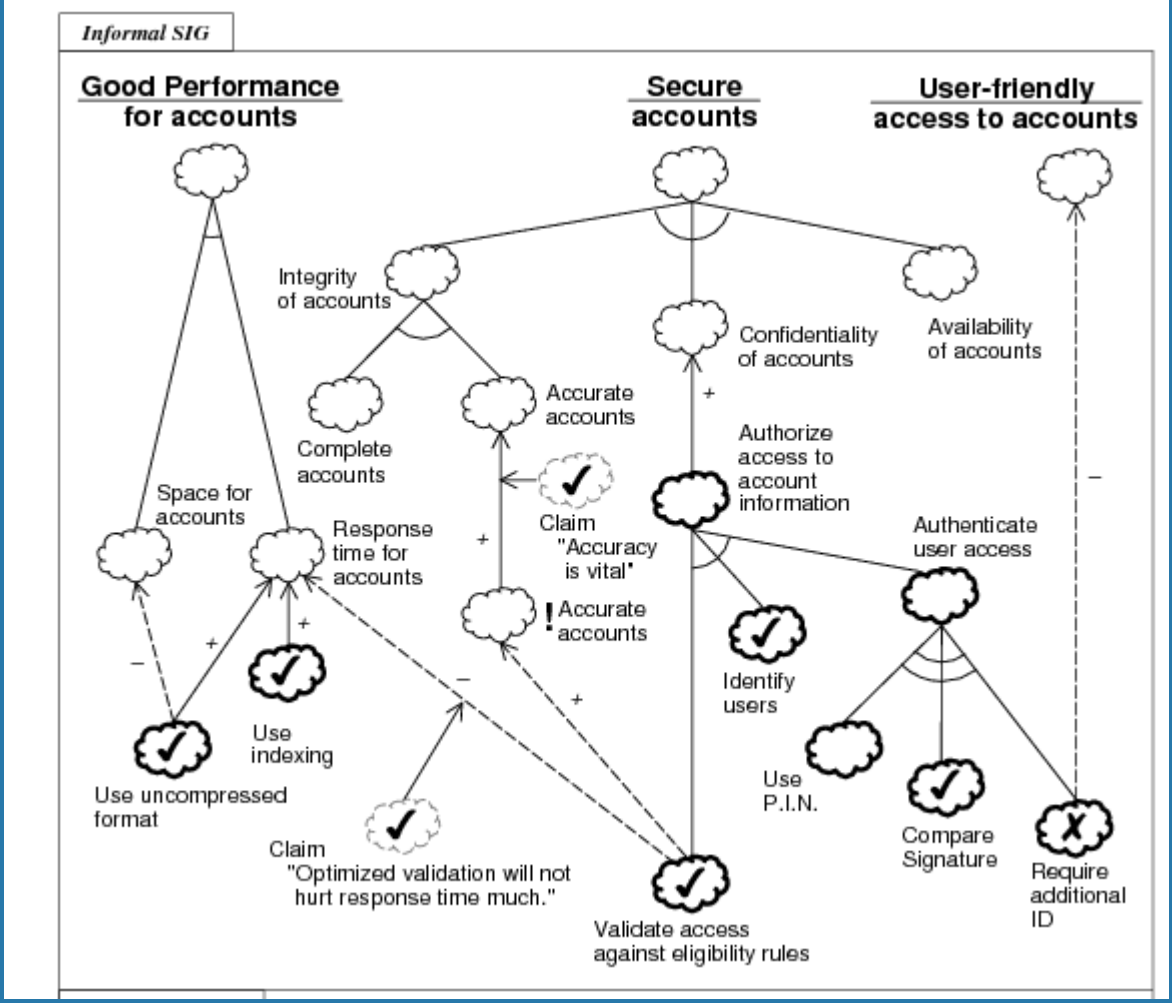
Explicitar interdependencias implícitas



Registrar fundamentación del diseño



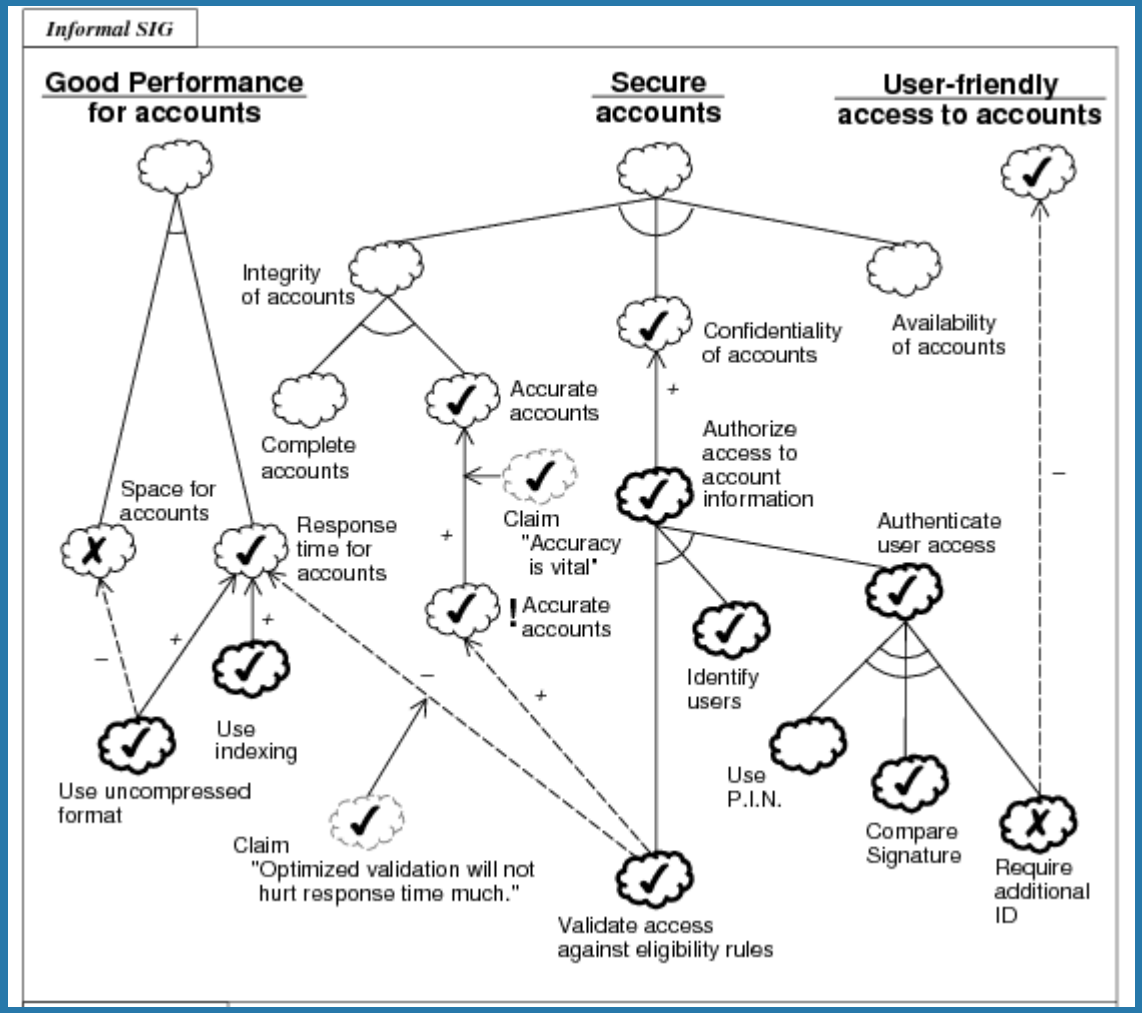
Seleccionar entre las alternativas



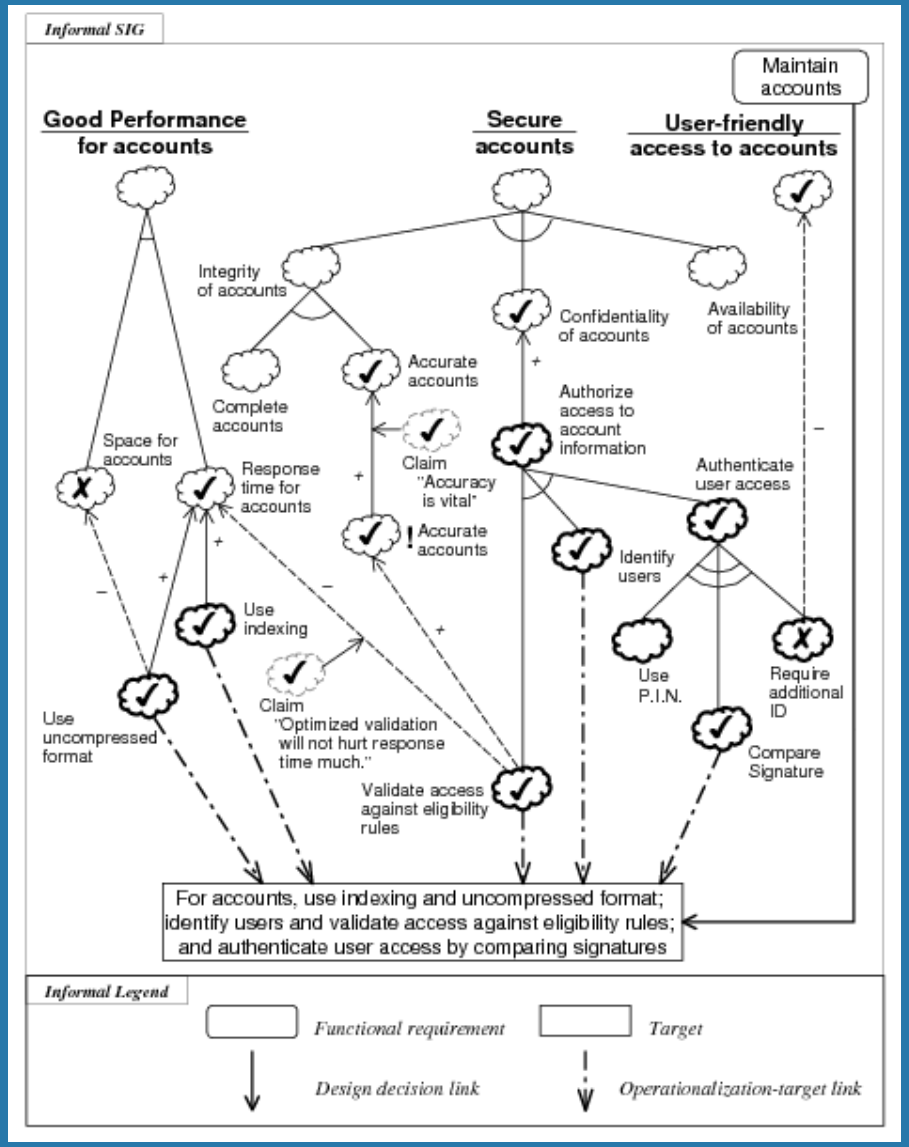
Evaluar impacto de decisiones

- En general, las soluciones encontradas son *suficientemente buenas* aunque quizás no sean las óptimas

Evaluar impacto de la decisión



Relacionar FRs a decisiones y NFRs



UNIVERSIDAD TECNICA
FEDERICO SANTA MARIA

Departamento de Informática
Universidad Técnica Federico Santa María

NFRs y Priorización

Ingeniería de Software

Hernán Astudillo & Gastón Márquez
Departamento de Informática
Universidad Técnica Federico Santa María