

# Ingeniería de Software II

## Primer Cuatrimestre de 2010

Clase 19 – Evaluación de Arquitecturas con ATAM y Walkthroughs

Buenos Aires, 28 de Junio de 2010

# ¿Por qué evaluar una arquitectura?

- ▶ Para tomar mejores decisiones! Y Además...
- ▶ Por razones económicas
- ▶ Fuerza la preparación de material para la revisión
- ▶ Captura las motivaciones detrás de la arquitectura
- ▶ Detección temprana de problemas
- ▶ Validación de requerimientos
- ▶ Arquitecturas mejoradas

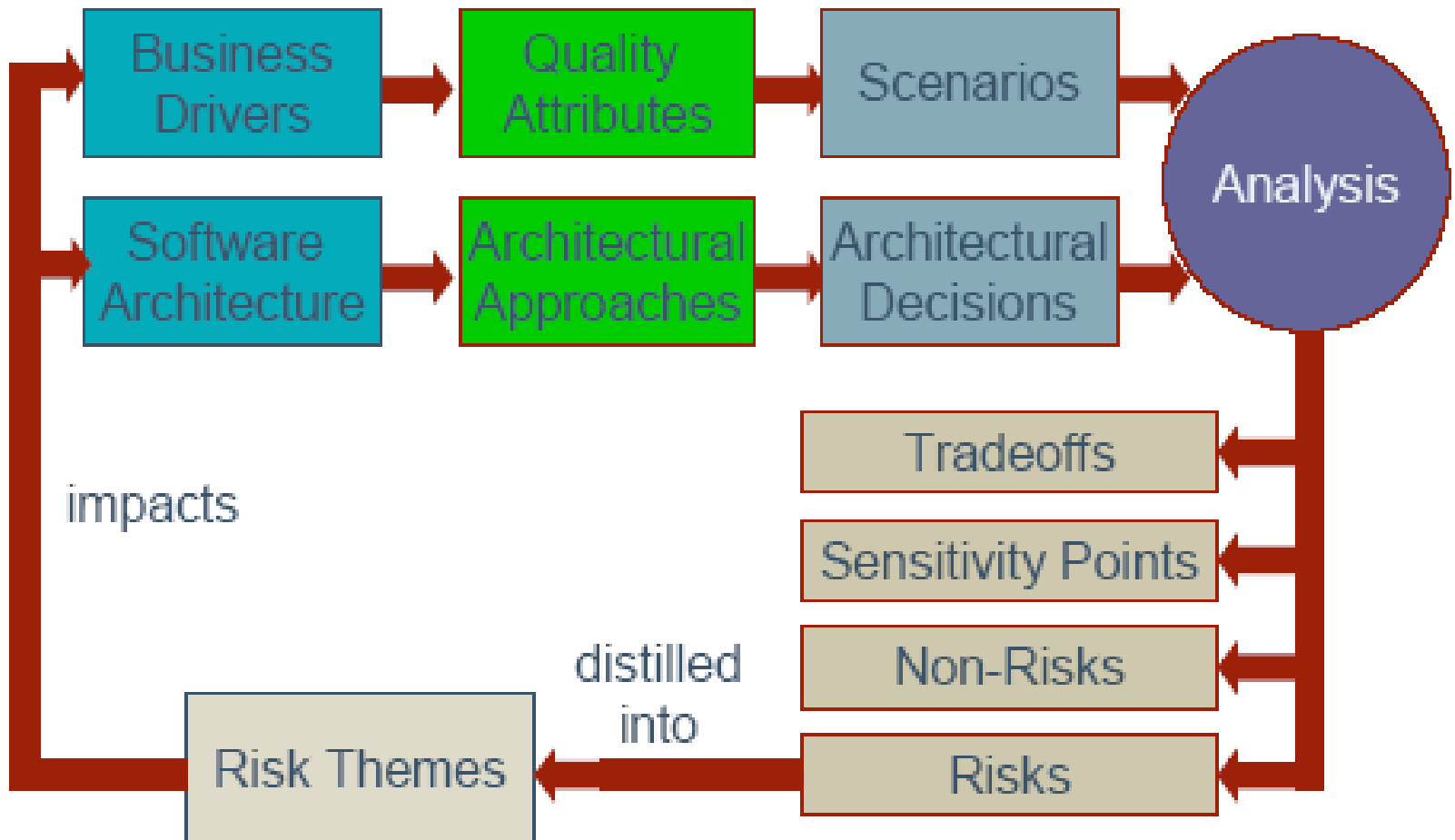
# Architecture Tradeoff Analysis Method (ATAM)

- ▶ El propósito de ATAM es evaluar las consecuencias de decisiones arquitectónicas a partir de requerimientos de atributos de calidad
- ▶ Es un método de identificación de riesgos
- ▶ ATAM es un método que ayuda a los “stakeholders” a generar las preguntas correctas para descubrir decisiones de arquitectura potencialmente problemáticas
- ▶ El propósito de ATAM no es proveer un análisis preciso. El propósito es descubrir los riesgos creados por decisiones arquitectónicas
- ▶ Queremos encontrar tendencias: correlación entre decisiones de arquitectura y predicciones de propiedades del sistema
- ▶ Puede ser realizado temprano en el proceso de desarrollo

# El Método ATAM

- Interacción breve y facilitada entre “stakeholders” que llevan a la identificación de riesgos, puntos sensibles y tradeoffs:
  - Riesgo: decisión arquitectónica potencialmente problemática
  - Punto sensible: propiedad de uno o más componentes (y/o relaciones entre componentes) que es crítica para poder alcanzar un atributo de calidad determinado
  - Punto de “tradeoff”: propiedad que afecta más que un atributo y es un punto sensible para más de un atributo
  - “Non risks” son buenas decisiones de arquitectura que normalmente están implícitas en la arquitectura
- Riesgos: foco de actividades de mitigación, por ejemplo profundizar el diseño o el análisis, realizar un prototipo
- Puntos sensibles y tradeoffs: pueden ser documentados explícitamente

# Flujo conceptual de ATAM



# Fases en ATAM



# ATAM - Participantes

- ▶ Equipo de Evaluación
  - ▶ 3 a 5 personas
  - ▶ Externo al grupo de proyecto
- ▶ Project *Decision-Makers*
  - ▶ Autoridad para aprobar cambios
  - ▶ El arquitecto está incluido en este grupo
- ▶ Stakeholders de la arquitectura
  - ▶ Interesados en Atributos de Calidad
  - ▶ Expertos del proyecto para implementar QAs

# Pasos del ATAM - Evaluación





# Pasos de ATAM (Evaluación) – Fase 1

## 1. Presentar ATAM

- ▶ Árboles de utilidad
- ▶ Escenarios ya Relevados y Análisis de Arquitectura

## 2. Presentar los “drivers” del negocio

- ▶ Requerimientos funcionales de alto nivel
- ▶ Requerimientos de atributos de calidad

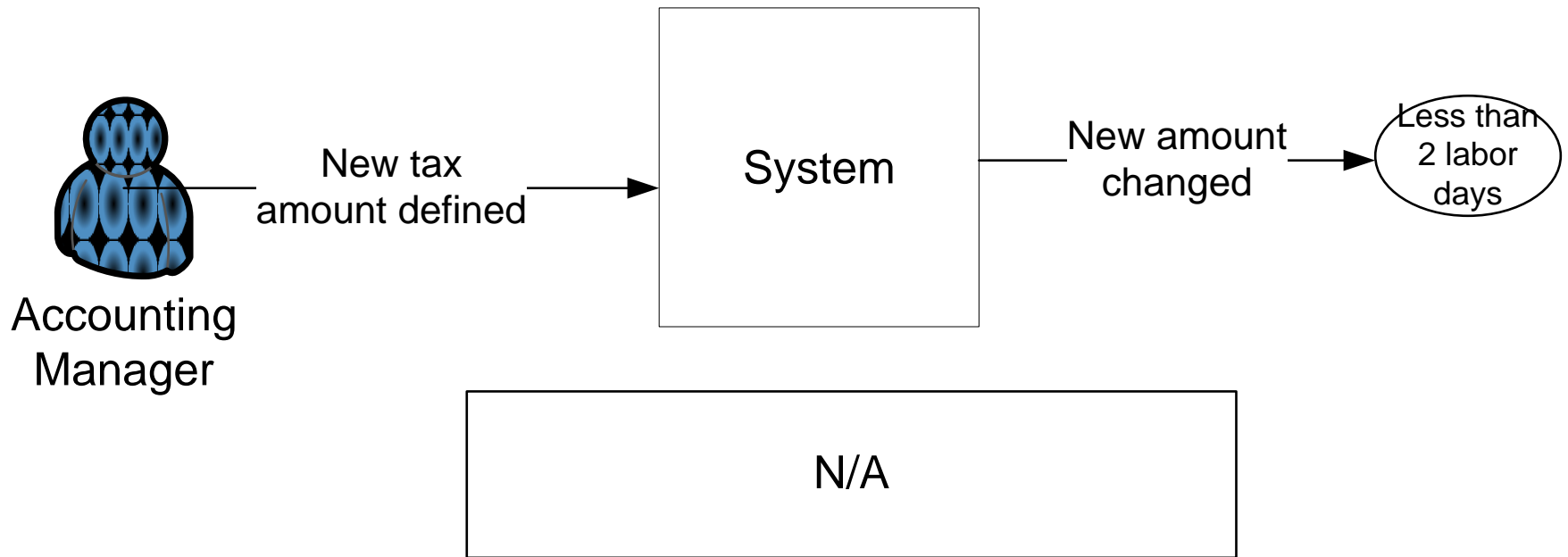
## 3. Presentar la arquitectura

- ▶ Restricciones técnicas
- ▶ Otros sistemas con los cuales interactuar

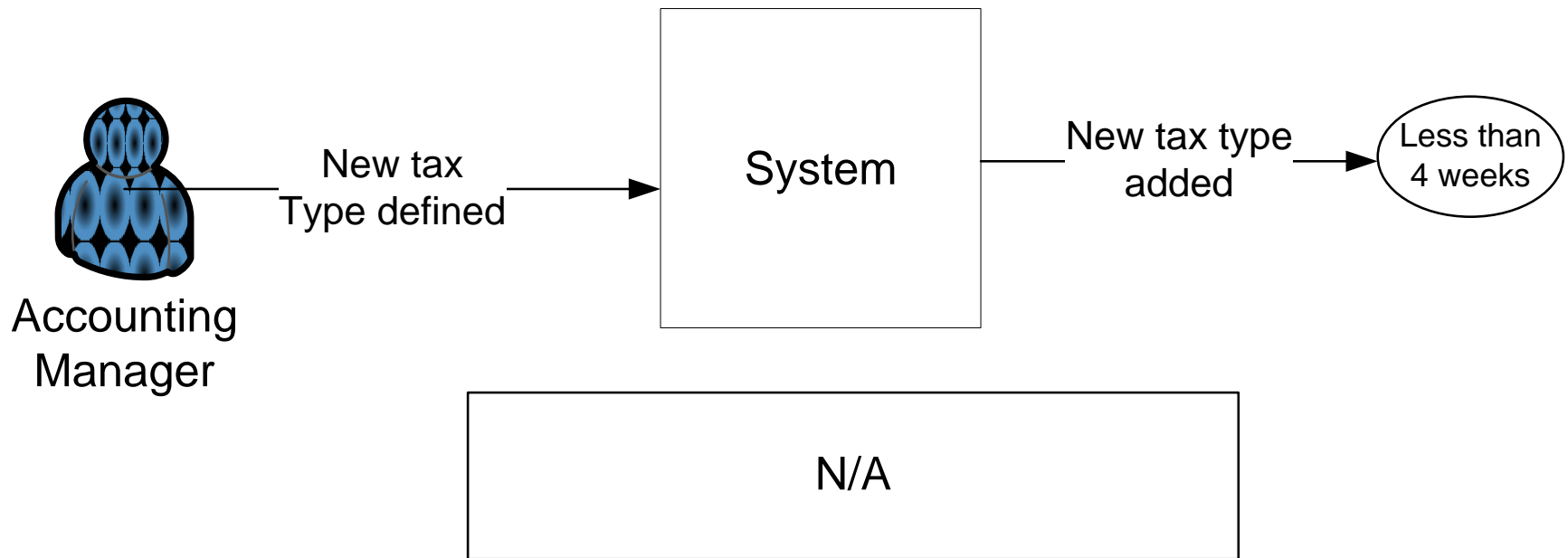
# Pasos de ATAM (Evaluación) – Fase 1

4. Identificar enfoques de arquitectura
  - Identificar aspectos clave para los QA
  - Identificar enfoques predominantes de arquitectura
5. Generar el árbol de utilidad
  - Nodos alto nivel => Objetivos Calidad
  - Hojas => Escenarios

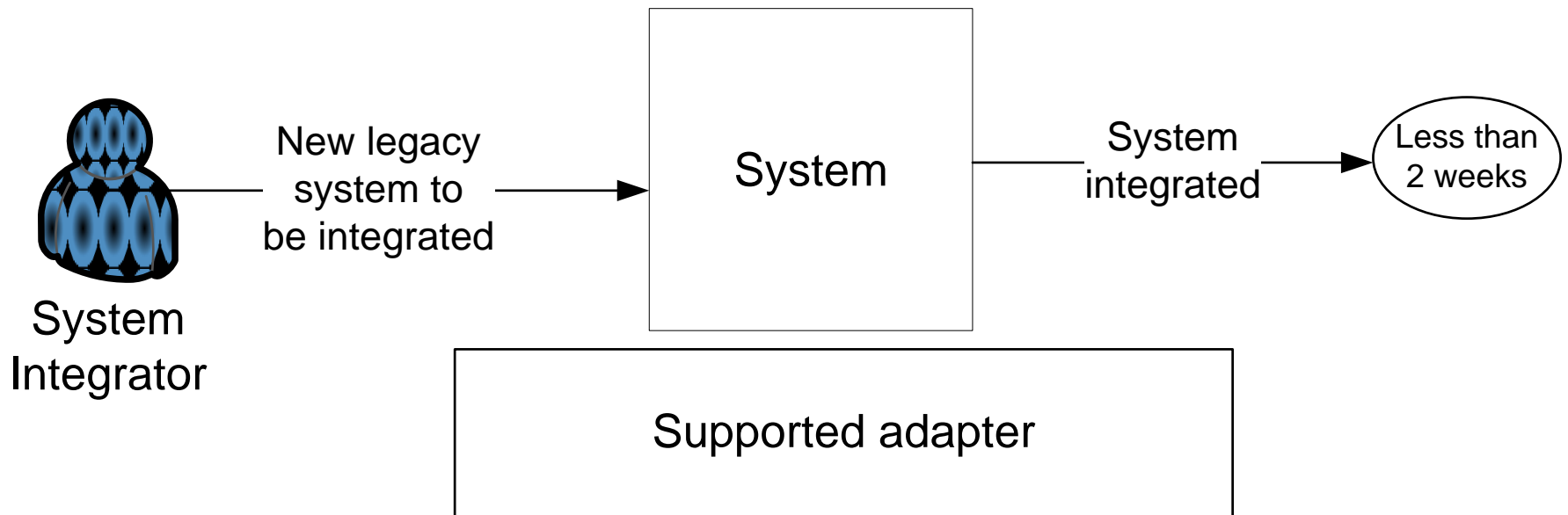
# Ejemplos de Escenarios a Evaluar



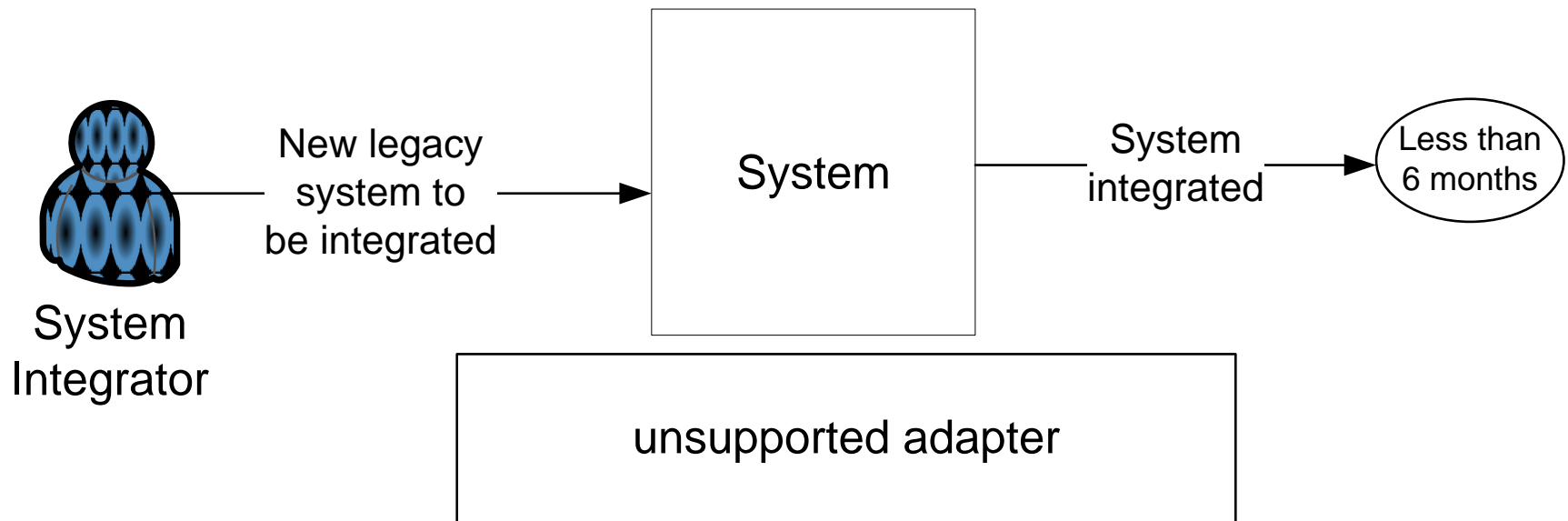
## Ejemplos de Escenarios a Evaluar (cont.)



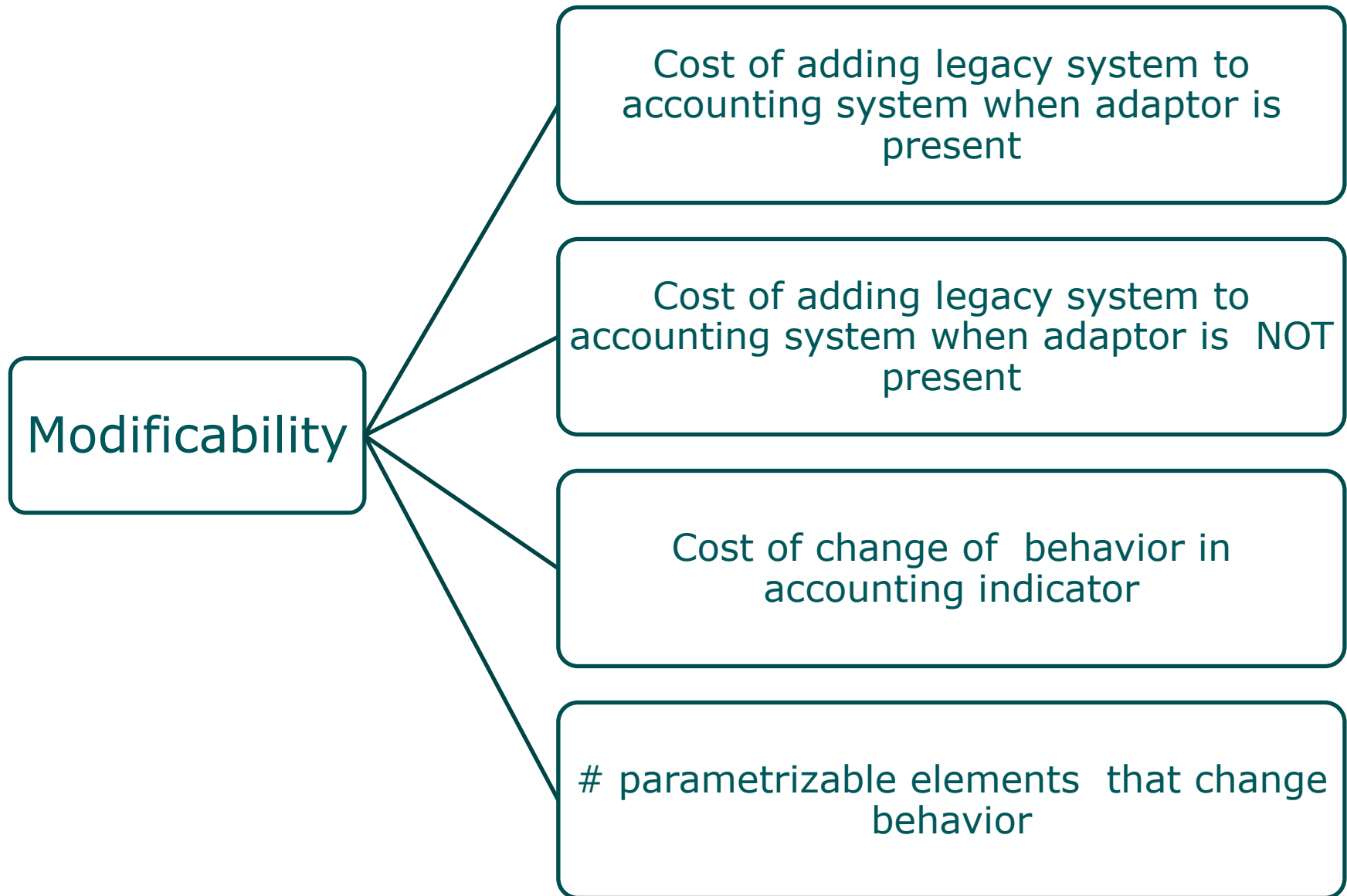
## Ejemplos de Escenarios a Evaluar (cont.)



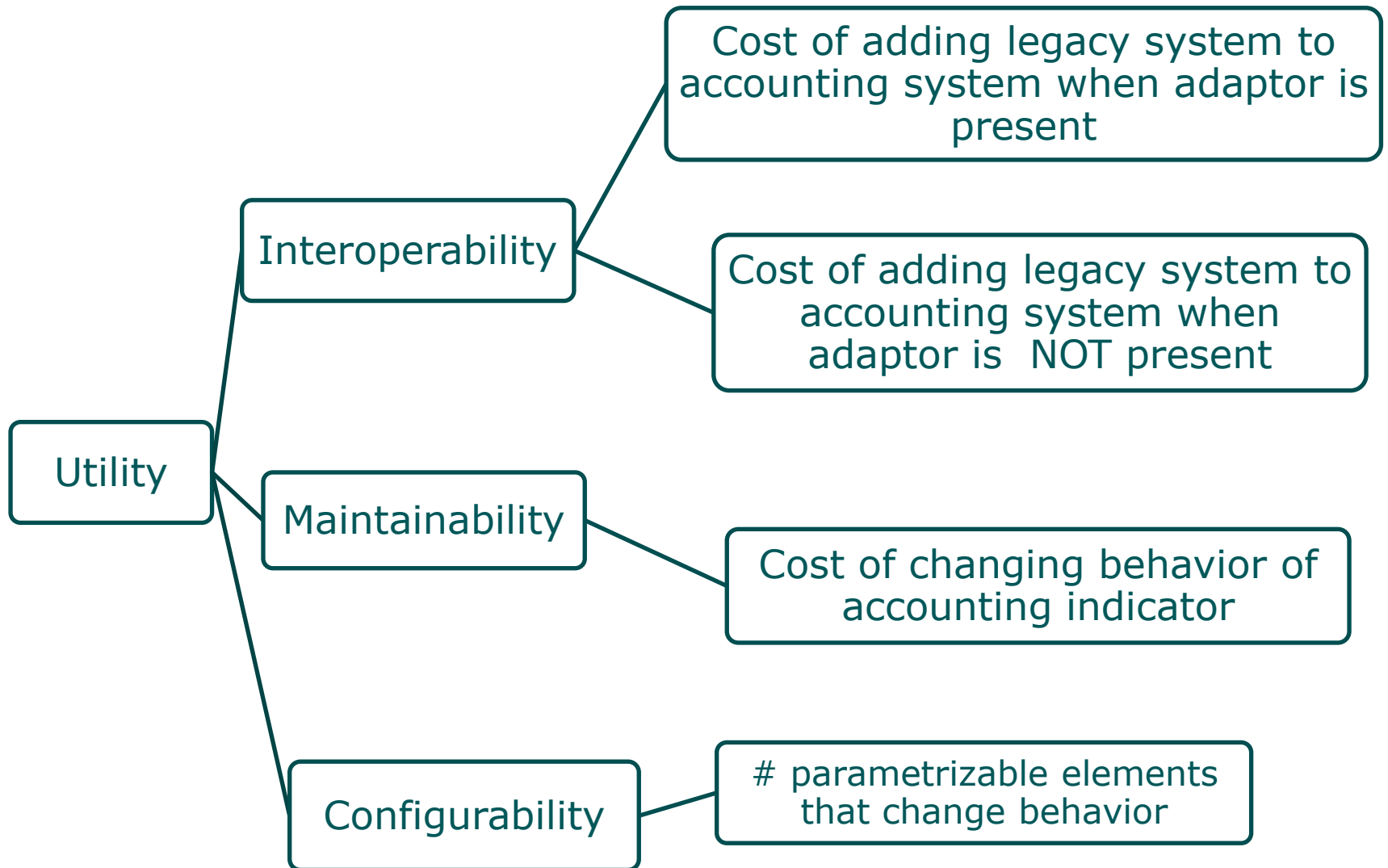
## Ejemplos de Escenarios a Evaluar (cont.)



## Ejemplos de Utility Tree

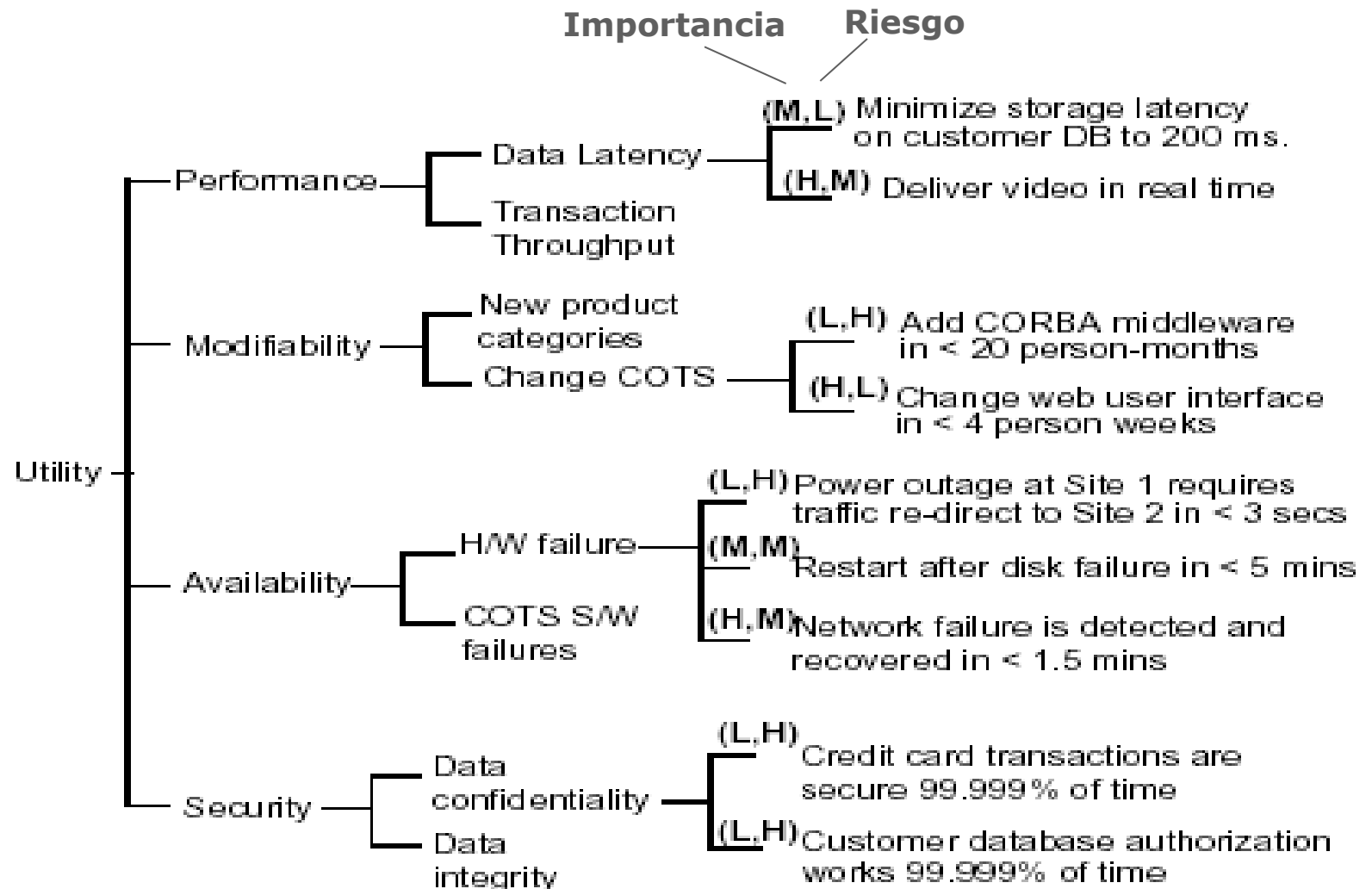


## Ejemplos de Utility Tree (cont.)





## Ejemplos de Utility Tree (cont.)



## Pasos de ATAM (Evaluación) – Fase 1

### 6. Analizar enfoques de arquitectura

- Identificar los enfoques para QA mas prioritarios
- Generar preguntas para los QA de mayor prioridad
- Identificar “riesgos”, “puntos sensibles”, “puntos de tradeoff” y “non-risks”

# Ejemplo de Análisis de Escenario

Scenario #: A12		Scenario: Detect and recover from HW failure of main switch.			
Attribute(s)	Availability				
Environment	Normal operations				
Stimulus	One of the CPUs fails				
Response	0.999999 availability of switch				
Architectural decisions		Sensitivity	Tradeoff	Risk	Nonrisk
Backup CPU(s)		S2		R8	
No backup data channel		S3	T3	R9	
Watchdog		S4			N12
Heartbeat		S5			N13
Failover routing		S6			N14
Reasoning	<p>Ensures no common mode failure by using different hardware and operating system (see Risk 8)</p> <p>Worst-case rollover is accomplished in 4 seconds as computing state takes that long at worst</p> <p>Guaranteed to detect failure within 2 seconds based on rates of heartbeat and watchdog</p> <p>Watchdog is simple and has proved reliable</p> <p>Availability requirement might be at risk due to lack of backup data channel ... (see Risk 9)</p>				
Architecture diagram	<pre>graph LR     In(( )) --&gt; P[Primar CPU OS1]     In --&gt; B[Backup CPU with Watchdog OS2]     P -- "heartbeat 1 sec." --&gt; B     P --&gt; S[Switch CPU OS1]     B --&gt; S     S --&gt; Out(( ))</pre>				

## Pasos de ATAM (Fase 2)

### 7. Brainstorm y priorizar escenarios

- Los escenarios del árbol de utilidad pueden servir como ejemplos
- Se agregan los nuevos escenarios al árbol de utilidad

### 8. Analizar enfoques de arquitectura

- Identificar los enfoques de arquitectura impactados por los escenarios generados en el paso anterior
- Este paso continua el análisis iniciado en el paso 6, usando los escenarios nuevos

### 9. Presentar resultados

## Salidas del Proceso

- ▶ Presentación concisa de la arquitectura
- ▶ Articulación de los objetivos de negocio
- ▶ Requerimientos de calidad expresados en escenarios
- ▶ Mapping entre requerimientos de calidad y decisiones de arquitectura
- ▶ Conjunto de puntos sensibles y de tradeoff identificados.
- ▶ Conjunto de riesgos y no-riesgos
- ▶ Conjunto de “risk themes”

## Cuándo usar ATAM

- ▶ ATAM puede ser usado a lo largo del ciclo de vida cuando hay una arquitectura de software para evaluar
- ▶ ATAM puede ser usado después de que una arquitectura se especificó pero hay poco o nada de código listo:
  - ▶ Para evaluar alternativas arquitectónicas
- ▶ Para evaluar la arquitectura de un sistema existente

## ATAM – Limitaciones

- ▶ **No tengo** Valuaciones de costo.
- ▶ **No considero** Variaciones de escenarios e impacto en la respuesta.
- ▶ **No es un** Método cuantitativo.

# Un “ATAM Simplificado”

		Efficiency		Reliability		Usability	
Subject	Approach	E1	F2	E3	E4	E5	E6
Infrastructure	Linux OS						
	MySQL Databse						
	Apache Tomcat						
High Level App Arch	4 Tier						
	Use of ORM						
	No SP						
Main Frameworks Used	Java Server Faces						
	Spring						
	..						
Design Patterns	Factory						
	...						
Reliability Strategy	Redundancy for ...						



# Walkthroughs

- ▶ Los Walkthroughs son un método de análisis de artefactos de desarrollo de software, muy útil para modelos gráficos
- ▶ Son una forma particular de “peer review”
- ▶ Objetivos
  - ▶ Detectar **posibles** defectos
  - ▶ Identificar oportunidades de mejora
  - ▶ Examinar alternativas
  - ▶ Aprender
- ▶ En general son usadas para revisar especificaciones de requerimientos, arquitecturas o diseños
- ▶ El concepto de “walkthrough” significa “recorrer” el sistema (qué pasa al recibir un estímulo)

## Walkthroughs: la reunión

- ▶ El presentador conoce a fondo el producto
- ▶ Los asistentes
  - ▶ son especialistas del negocio, la tecnología usada o conocedores de los sistemas donde hay impacto
  - ▶ no preparan esta actividad
- ▶ Se pueden discutir brevemente los temas planteados (problemas, sugerencias de mejora)
- ▶ Si funcionan bien: buenos resultados y buena relación calidad / esfuerzo
- ▶ Ser cuidadoso con el tiempo y el foco de la reunión!