Ingeniería de Software II

Segundo Cuatrimestre de 2011

Clase 20 – Evaluación de Arquitecturas con ATAM y Walkthroughs

Buenos Aires, 3 de Noviembre de 2011

¿Por qué evaluar una arquitectura?

- Detección temprana de problemas
- Validación de requerimientos
- Comprensión

- Captura las motivaciones detrás de la arquitectura
- Para tomar mejores decisiones
- Arquitecturas mejoradas
- Fuerza la preparación de material para la revisión

Análisis de arquitectura

- Dos enfoques distintos
- 1) Basado en herramientas
  - Consistencia / Propiedades más duras
  - Requiere formalización
  - Uso de herramientas
- 2) Basado en inspecciones / revisiones / walkthroughs
  - Basado en asistencia de expertos y stakeholders
  - Requieren un «proceso»

Análisis basado en herramientas

- **Consistencia**
  - ¿Las partes enganchan correctamente?
- **Completitud**
  - ¿Falta alguna parte?
- **Corrección y conformidad**
  - ¿La arquitectura satisface la especificación?
  - ¿La implementación "conforma" a la arquitectura?
- **Refinamiento**
  - ¿Puede una arquitectura reemplazar a otra?
- **Comportamiento/Performance/Confiabilidad del sistema**
  - ¿Cuales el comportamiento agregado del sistema dado el de sus partes?
- **Evaluación de decisiones de diseño y compromisos**
  - ¿Cómo decidir que decisiones arquitectónicas hacer?

Analizando Arquitecturas

- Parte “estática”
  - Topología
  - Propiedades (eg: latencia, performance)
  - Tipos de elementos (estilos)
- Parte “dinámica”
  - Comportamiento (ej: protocolos, implementaciones, etc.)
  - Reconfiguración (ej: nuevos attachs, aparición de componentes, etc.)

Arquitectura en ACME

```
System simpleCS = { …

   // simple rule requiring a primary server
   Invariant exists c : server in self.components |
                c.isPrimaryServer == true;

   // simple performance heuristic
   Heuristic forall s : server in self.components |
                s.transactionRate >= 100;

   // do not allow client-client connections
   Function no-peer-connections(sys : System) =
       forall c1, c2 in sys.components |
             connected(c1, c2) ->
                   !(declaresType(c1,clientT)
                   and declaresType(c2, clientT));
… };
```

Chequeando arquitecturas en ACME

- El chequeo de arquitectura se reduce a un chequeo de tipos
  - Reglas de tipado clásicas
  - + Verificación de invariantes
- Opciones:
  - Demostradores de teoremas
    - Automático (pero incompleto)
    - Asistidos
  - Refutadores
    - En vez de probar buscan contraejemplos
    - Alloy

Cómo se puede automatizar el análisis

- Usando model checking
- Traducir CSP a FSMs
- Para deadlock
  - Chequear la composicion paralela
- Otros:
  - Ver si cierto componente es alcanzable

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

Fuente: ***ATAM: Method for Architecture Evaluation.*** **Rick Kazman, Mark Klein, Paul Clements. August 2000. TECHNICAL REPORT CMU/SEI-2000-TR-004. ESC-TR-2000-004**

Objetivos

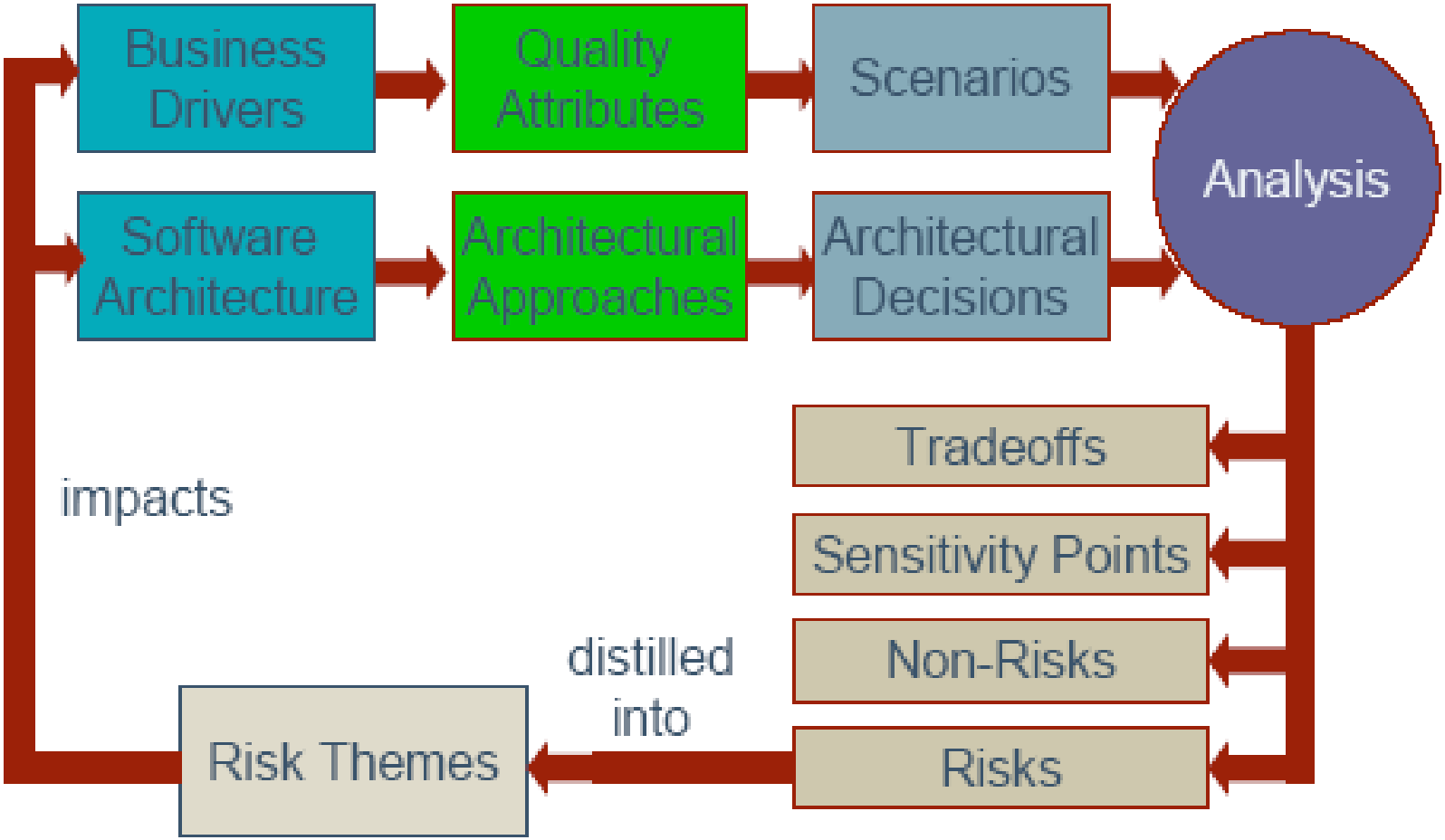
- Obtener y/o refinar una descripción precisa de los atributos de calidad que afectan a la arquitectura
- Obtener y/o refinar una descripción sobre las decisiones de arquitectura principales
- Dadas estas dos:
  - Evaluar las decisiones de arquitectura para determinar si estás resuelven adecuadamente los atributos de calidad

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

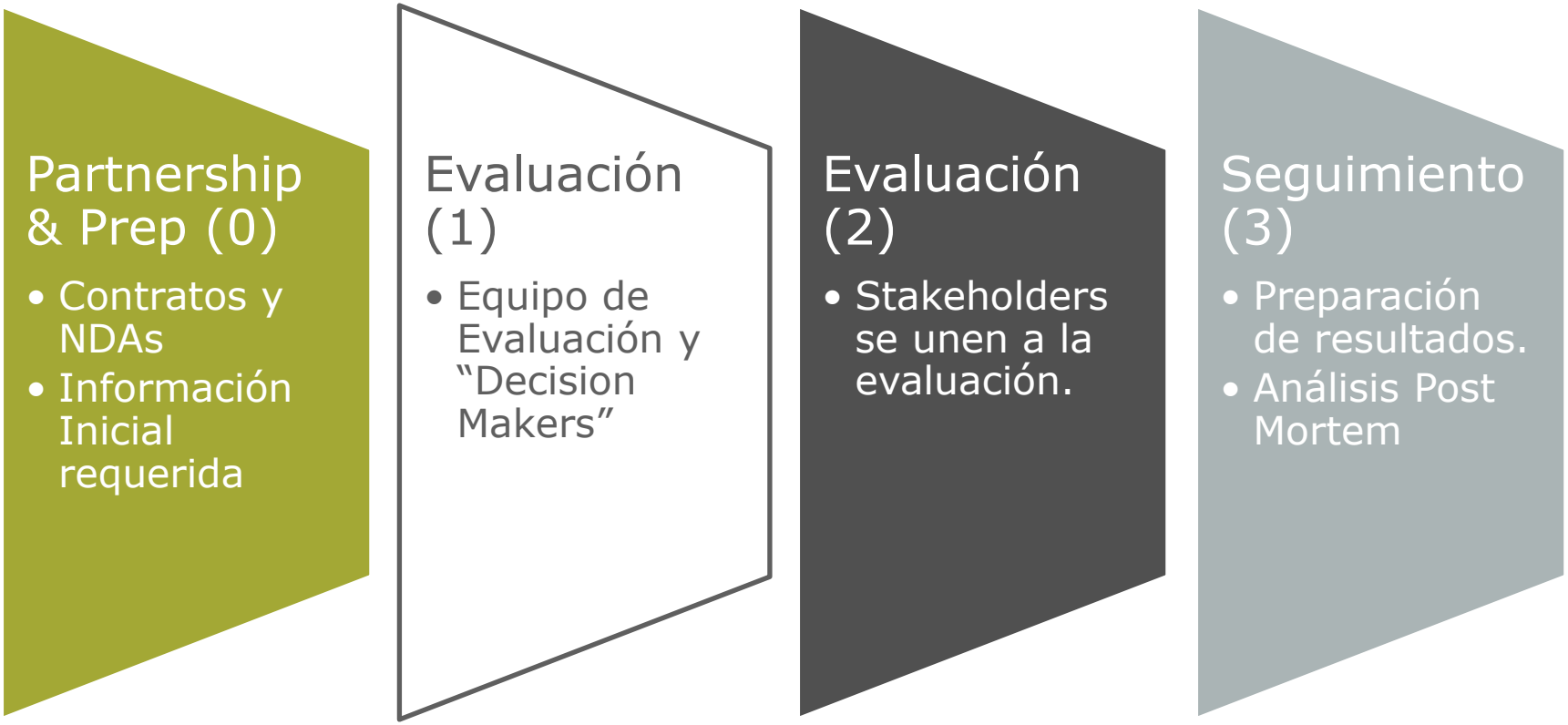
Fuente: ***ATAM: Method for Architecture Evaluation.*** **Rick Kazman, Mark Klein, Paul Clements. August 2000. TECHNICAL REPORT CMU/SEI-2000-TR-004. ESC-TR-2000-004**

Flujo conceptual de ATAM



Fuente: ***ATAM: Method for Architecture Evaluation.*** **Rick Kazman, Mark Klein, Paul Clements. August 2000. TECHNICAL REPORT CMU/SEI-2000-TR-004. ESC-TR-2000-004**

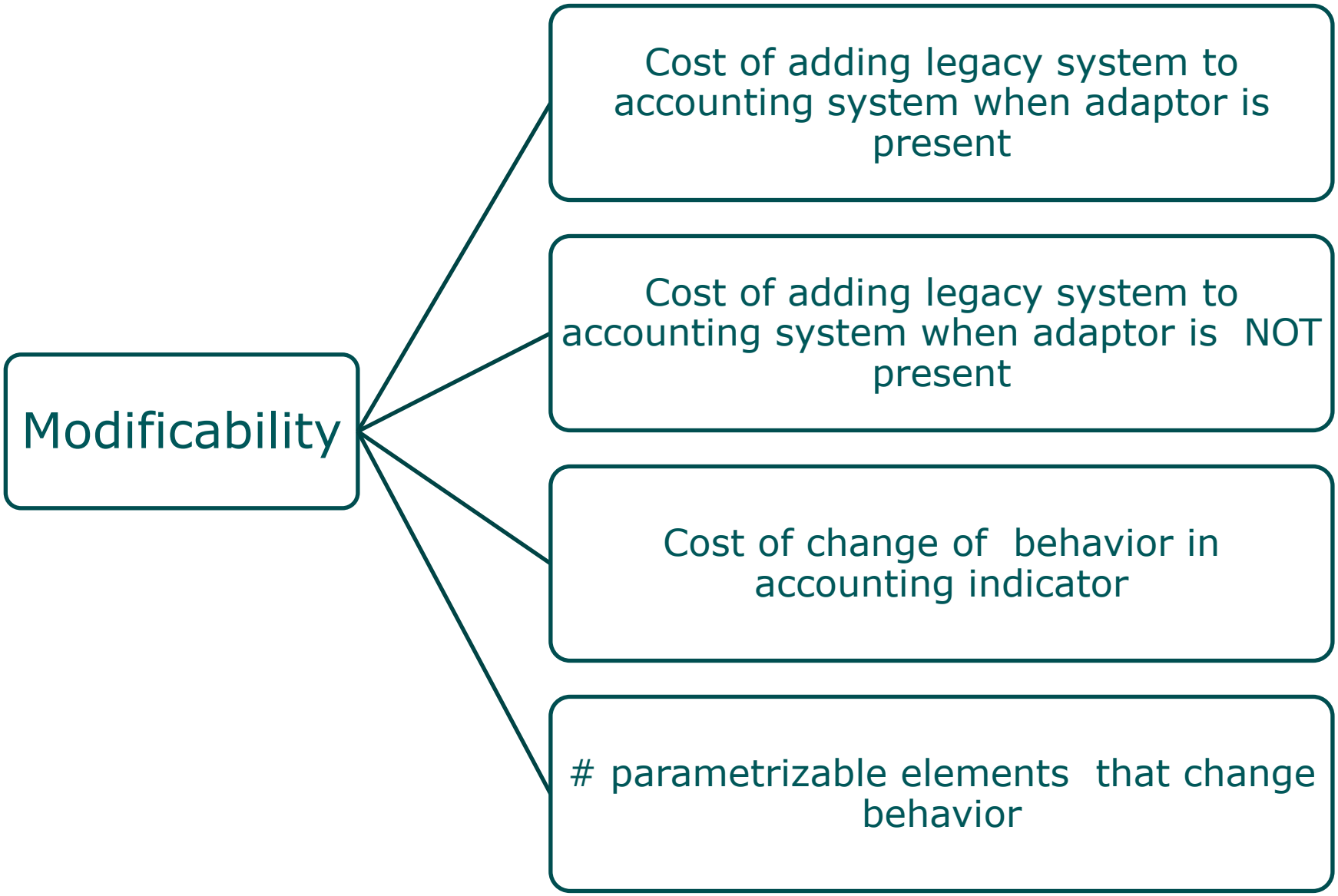
Fases en ATAM



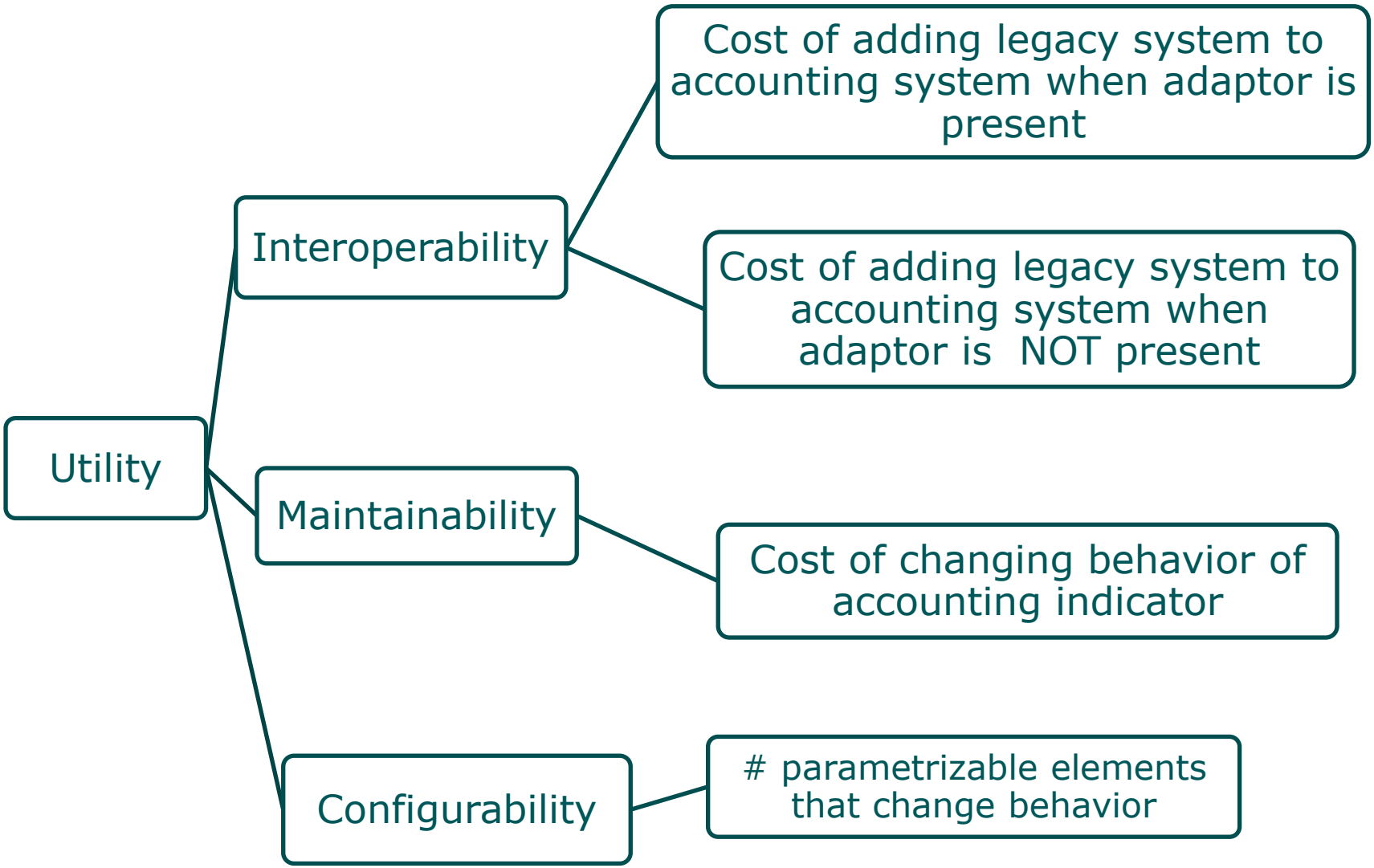
Pasos del ATAM - Evaluación

- 1.Presentar ATAM
- 2. Presentar "business drivers"
- 3. Presentar la arquitectura
- 4. Identificar "architectural approaches"
- 5. Generar "utility tree" de atributos de calidad
- 6. Analizar "architectural approaches"
- 7. Hacer brainstorm y dar prioridades a los escenarios
- 8. Analizar "architectural approaches"
- 9. Presentar los resultados

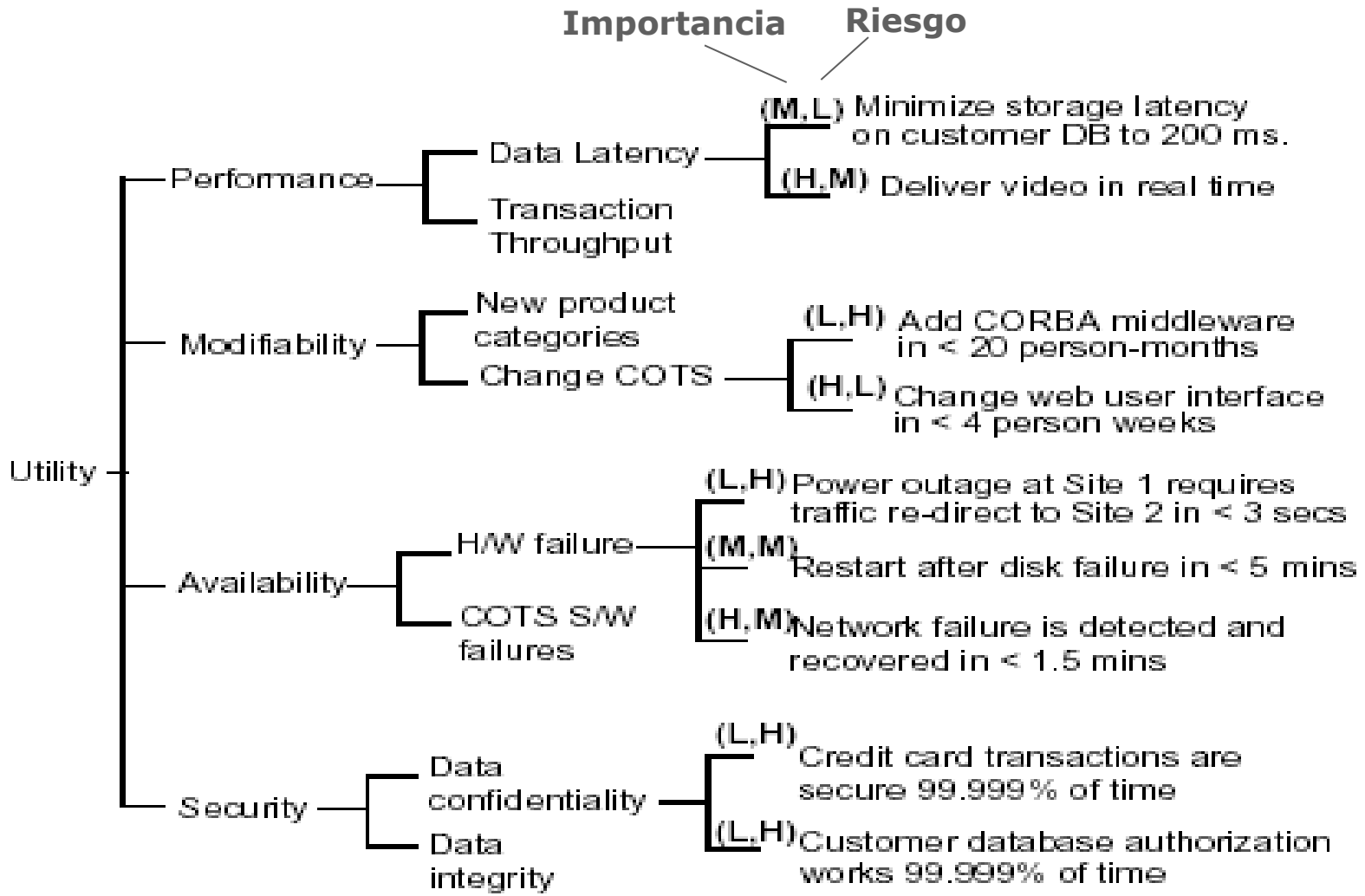
Ejemplos de Utility Tree



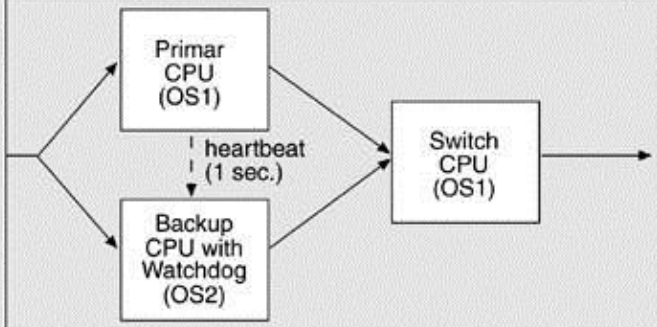
Ejemplos de Utility Tree (cont.)



Ejemplos de Utility Tree (cont.)



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	Ensures no common mode failure by using different hardware and operating system (see Risk 8) Worst-case rollover is accomplished in 4 seconds as computing state takes that long at worst Guaranteed to detect failure within 2 seconds based on rates of heartbeat and watchdog Watchdog is simple and has proved reliable Availability requirement might be at risk due to lack of backup data channel ... (see Risk 9)			
Architecture diagram				

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”
  - (Asuntos que si no se atienden pueden comprometer los objetivos del negocio)

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.

Ejemplo: analizando el File System de Google

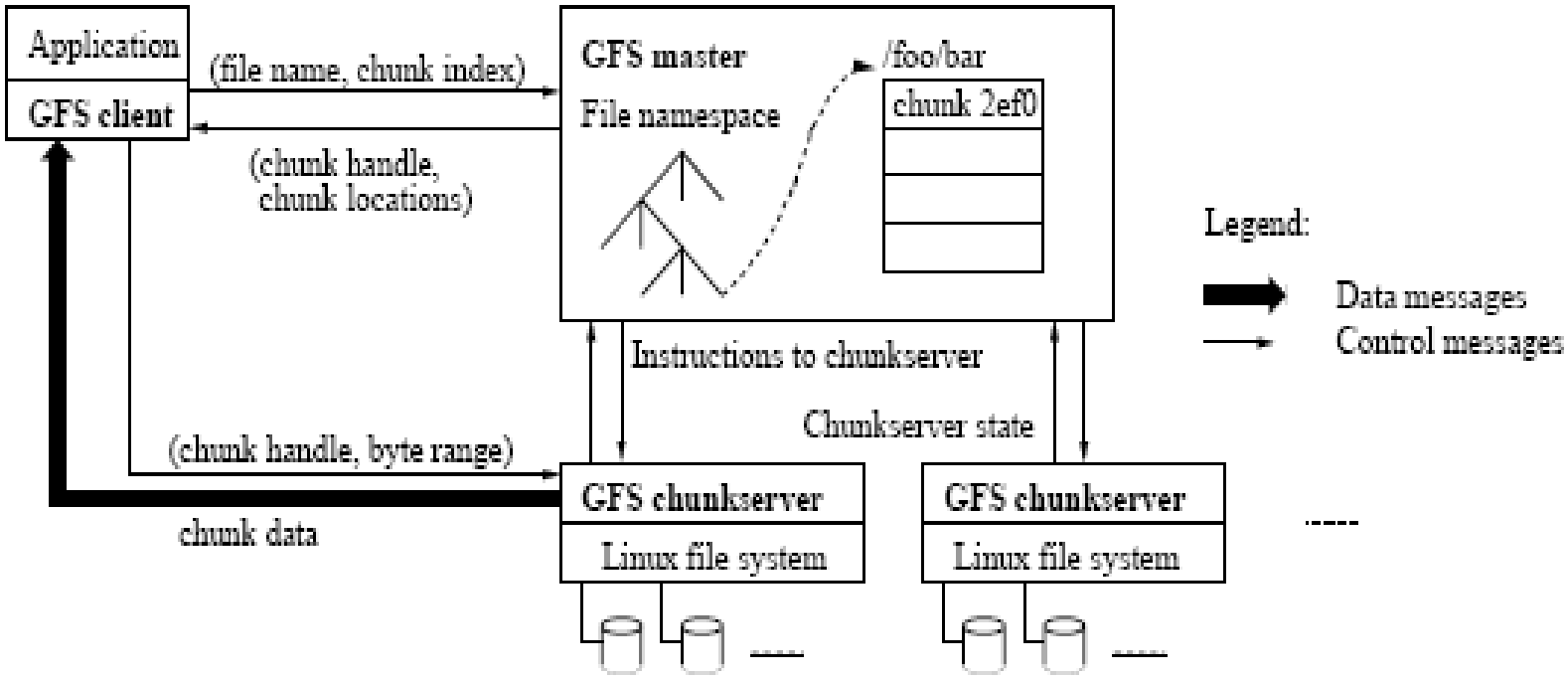


Figure 1: GFS Architecture

Source: "The Google File System" Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung

Un “ATAM Simpificado”

		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 sofware, 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 “walkthough” 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!