

P12. *Architecture Evaluation*

2014

Sungwon Kang

12. 아키텍처의 평가

12.1 아키텍처 평가 절차

12.2 CBAM 기법 [Kazman 02]의 예

12.3 CBAM 기법을 사용한 아키텍처 평가

12.1 아키텍처 평가 절차

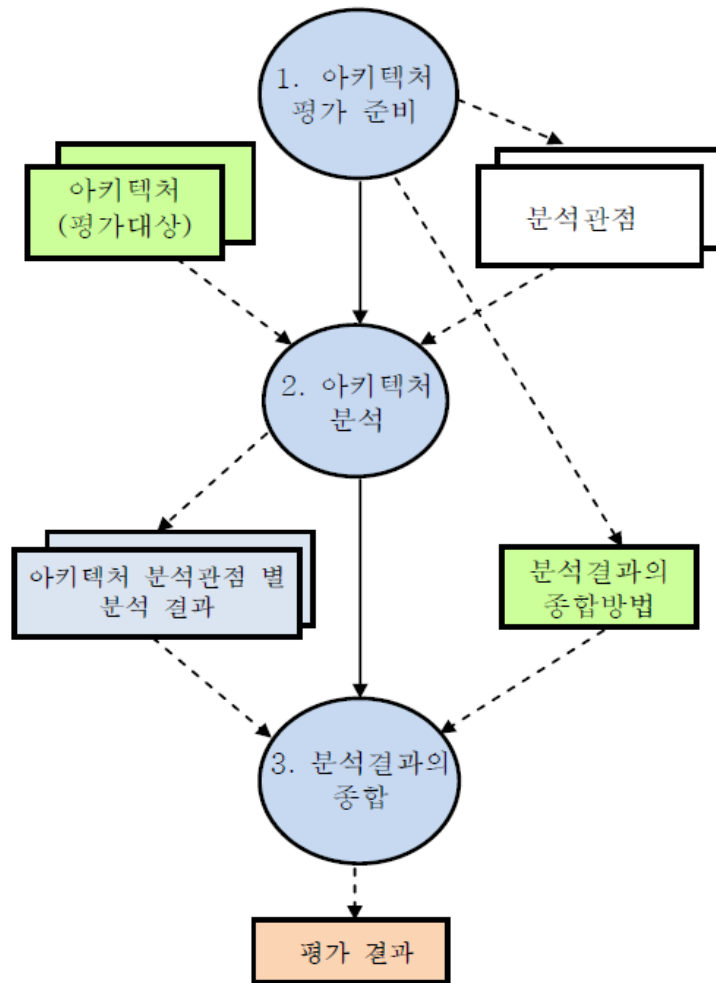


그림 12-1. 아키텍처 평가의 절차

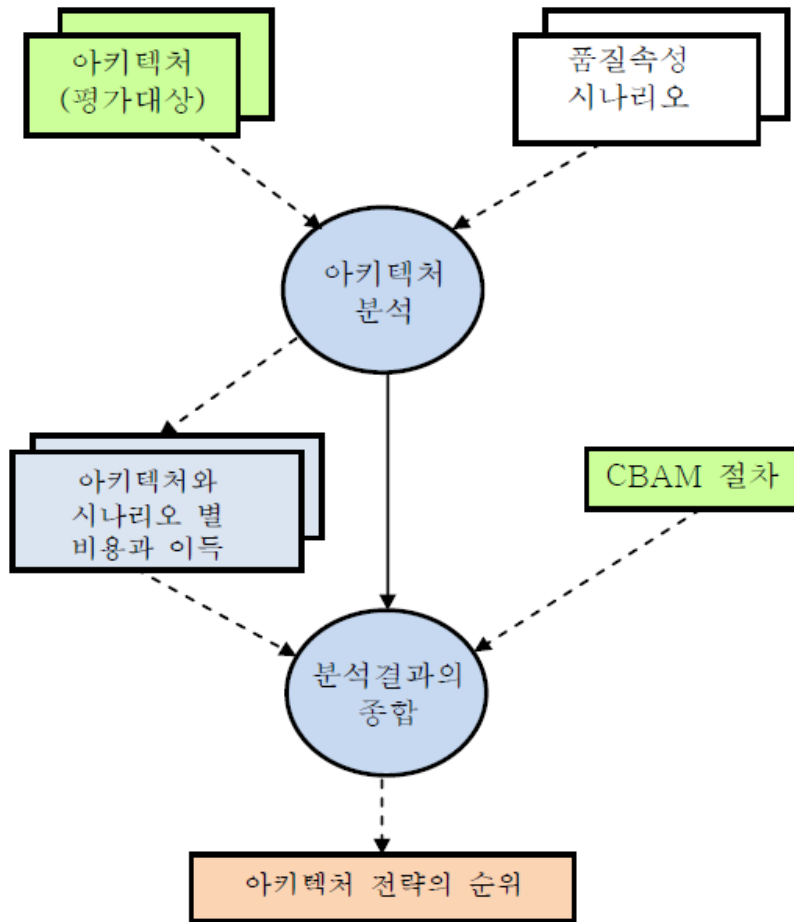


그림 12-2. CBAM 아키텍처 평가

12.2 CBAM 기법 [Kazman 02]

- 평가의 수단 제공
- 이를 이용하여 어떻게 아키텍처 전체에 대한 평가를 할 것인가를 결정하여야 한다.

CBAM Evaluation Method

- Why CBAM ?

- Can we do better than this **evaluation** ?

	VP1	VP2	VP3
AS1	O	O	X
AS2	Δ	O	O
AS3	X	Δ	O

- Things to consider for investment

- Benefit
 - Cost
 - Uncertainty – chances that something will happen or not happen
 - Risk – what possible consequences can occur

- How can we make investment decisions by quantifying such factors?

입력: 1) 아키텍처
2) 아키텍처 전략들
3) N 개의 품질속성 시나리오

출력: 각 아키텍처 전략의 순위

Iteration 1: Considers
cost and benefit
Iteration 2: Considers
uncertainty and risk

1. 시나리오 정리 (N 개의 시나리오를 $1/3 N$ 개의 시나리오로 축소)
2. 시나리오 정제
3. 시나리오 우선순위 ($1/6 N$ 개의 시나리오로 축소)
4. 각 시나리오의 응답수준별로 해당하는 효용값 부여
5. 아키텍처 전략을 개발하고 품질속성 응답수준 결정
6. 보정(interpolation)으로 기대되는 품질속성 응답수준의 효용값 결정
7. 아키텍처 전략(AS)이 주는 총이익 계산
8. 투자수익률(ROI)을 기초로 아키텍처 전략 선정

계량적 측정을 위하여
정교한 단계 구분을
가짐

반복 I 를 실행 중인 경우 반복 II 를 위하여 단계 2 로 이동한다.

반복 II 를 실행 중인 경우, 단계 9 로 이동한다.

9. 직관을 이용하여 결과 확인

그림 12-3. CBAM 절차

ECS 시스템 사례연구[Kazman 02]

- The Earth Observing System (EOS) 는 NASA 인공위성들로 구성
지구의 변화연구 등을 위한 데이터 수집
- ECS 는 EOS Data Information System (EOSDIS)의 핵심시스템(Core System)으로 ECS는 매일 수백 gigabytes 의 데이터를 처리.
=> performance 와 availability 가 중요
- EOS는 장기 프로젝트
=> modifiability 도 중요
- ATAM을 통한 사전분석에서, stakeholder로부터 많은 변경제안(ASs)이 수집됨
- PM은 제한된 예산으로 적은 수의 ASs를 선별해서 구현
=> PM은 ROI에 기초한 결정을 하기 위하여 CBAM을 사용

Iteration I

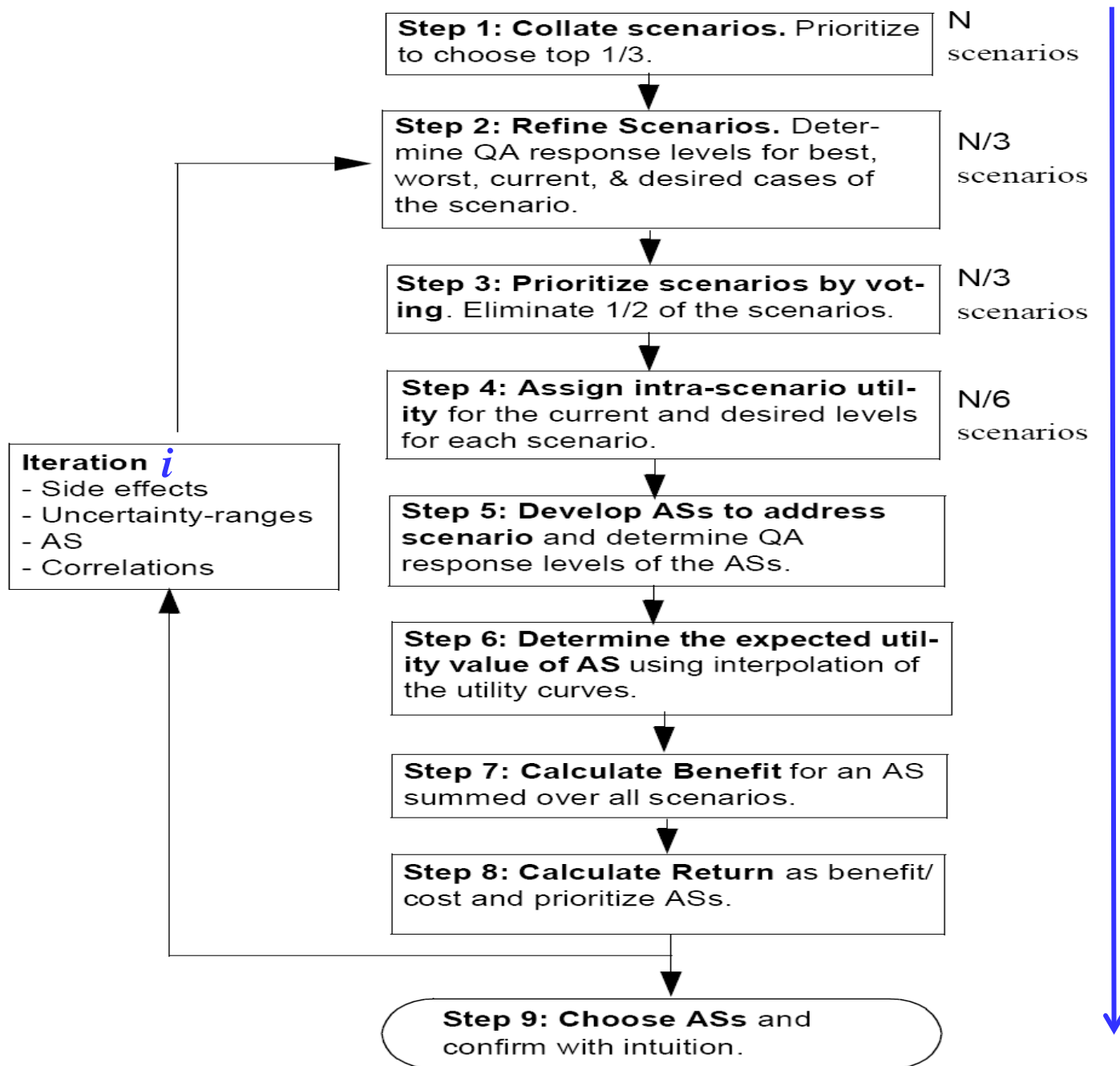


Figure 2: Process Flow Diagram for the CBAM

단계 1: 시나리오 정리

Table 1: Collected Scenarios

Scenario	Scenario Description
1	Reduce data distribution failures that result in hung distribution requests requiring manual intervention.
2	Reduce data distribution failures that result in lost distribution requests.
3	Reduce the number of orders that fail on the order submission process.
4	Reduce order failures that result in hung orders requiring manual intervention.
5	Reduce order failures that result in lost orders.
6	There is no good method of tracking ECS. Guest failed/cancelled orders without much manual intervention (e.g., spreadsheets).
7	Users need more information on why their orders for data failed.
8	Due to limitations, there is a need to artificially limit the size and number of orders.
9	Small orders result in too many notifications to users.
10	The system should process a 50 gigabyte user request in 1 day, and a 1 terabyte user request in 1 week.

단계 2: 시나리오 정제

- 단계 1의 상태의 시나리오는 측정가능한 형태가 아니므로, stimulus/response의 측정가능한 형태로 분석
- 동시에
 - worst case:
 - current case:
 - desired case:
 - best case :의 경우에 대한 목표치를 결정하여 기록
- 새로운 시스템의 개발의 경우, 응답수준 중 **Current** 수준은 불필요.

단계 2: 시나리오 정제

Table 2: Response Goals for Refined Scenarios

Scenario	Response Goals			
	Worst	Current	Desired	Best
1	10% hung	5% hung	1% hung	0% hung
2	> 5% lost	< 1% lost	0% lost	0% lost
3	10% fail	5% fail	1% fail	0% fail
4	10% hung	5% hung	1% hung	0% hung
5	10% lost	< 1% lost	0% lost	0% lost
6	50% need help	25% need help	0% need help	0% need help
7	10% get info	50% get info	100% get info	100% get info
8	50% limited	30% limited	0% limited	0% limited
9	1 per granule	1 per granule	1 per 100 granules	1 per 1000 granules
10	< 50% meet goal	60% meet goal	80% meet goal	> 90% meet goal

단계 3: 시나리오 우선순위

- 평가팀은
 - 개별적 혹은
 - 팀전체로서총 100점의 점수를 선택된 시나리오 집합에 할당
(Table 3에서 5의 배수로 점수가 주어졌으나, 이는 요구사항은 아님.)

단계 3: 시나리오 우선순위

Table 3: Refined Scenarios with Votes

Scenario	Votes	Response Goals			
		Worst	Current	Desired	Best
1	10	10% hung	5% hung	1% hung	0% hung
2	15	> 5% lost	< 1% lost	0% lost	0% lost
3	15	10% fail	5% fail	1% fail	0% fail
4	10	10% hung	5% hung	1% hung	0% hung
5	15	10% lost	< 1% lost	0% lost	0% lost
6	10	50% need help	25% need help	0% need help	0% need help
7	5	10% get info	50% get info	100% get info	100% get info
8	5	50% limited	30% limited	0% limited	0% limited
9	10	1 per granule	1 per granule	1 per 100 granules	1 per 1000 granules
10	5	< 50% meet goal	60% meet goal	80% meet goal	> 90% meet goal

단계 4: 각 시나리오의 응답수준별로 해당하는 **utility** 부여

- 컨센서스에 의하여 각 시나리오의 효용(utility)을 결정
 - 0: no utility
 - 100: the most utility possible.

단계 4: 각 시나리오의 응답수준별로 해당하는 **utility** 부여

Table 4: Scenarios with Votes and Utility Scores

Scenario	Votes	Utility Scores			
		Worst	Current	Desired	Best
1	10	10	80	95	100
2	15	0	70	100	100
3	15	25	70	100	100
4	10	10	80	95	100
5	15	0	70	100	100
6	10	0	80	100	100
7	5	10	70	100	100
8	5	0	20	100	100
9	10	50	50	80	90
10	5	0	70	90	100

단계 5: 아키텍처 전략을 개발하고 품질속성 응답수준 결정

- 앞의 시나리오 요구사항에 기초하여, 10개의 AS가 ECS 아키텍트에 의하여 도출
- 하나의 AS는 여러 개의 시나리오에 영향을 줄 수 있음
- 각 AS가 달성하는 QA응답수준을 각 시나리오 별로 결정

단계 5: 아키텍처 전략을 개발하고 품질속성 응답수준 결정

Table 5: Architectural Strategies and Scenarios Addressed

AS	AS Name	AS Brief Description	Scenarios Affected	Current Response	Expected Response
1	Order persistence on submission	Store order as soon as it arrives in system.	3	5% fail	2% Fail
			5	<1% lost	0% lost
			6	25% need help	0% need help
2	Order chunking	Allow operators to partition large orders into multiple small orders.	8	30% limited	15% limited
3	Order bundling	Combine multiple small orders into one large order.	9	1 per granule	1 per 100
			10	60% meet goal	55% meet goal
4	Order segmentation	Allow an operator to skip items that cannot be retrieved due to data quality or availability issues.	4	5% hung	2% hung
5	Order reassignment	Allow an operator to reassign the media type for items in an order.	1	5% hung	2% hung

Table 5: Architectural Strategies and Scenarios Addressed (Continued)

AS	AS Name	AS Brief Description	Scenarios Affected	Current Response	Expected Response
6	Order retry	Allow an operator to retry an order or items in an order that may have failed due to temporary system or data problems.	4	5% hung	3% hung
7	Forced order completion	Allow an operator to override an item's unavailability due to data quality constraints.	1	5% hung	3% hung
8	Failed order notification	Ensure that users are notified only when part of their order has truly failed and provide a detailed status of each item; user notification would occur only if the operator okays such a notification; the operator may edit the notification.	6	25% need help	20% need help
			7	50% get info	90% get info
9	Granule-level order tracking	An operator and user can determine the status for each item in their order.	6	25% need help	10% need help
			7	50% get info	95% get info
10	Links to user information	An operator can quickly locate a user's contact information. The server will access science data server (SDSRV) information to determine any data restrictions that might apply and will route orders/order segments to the appropriate distribution capabilities, including data distribution (DDIST), PDS, external subsetters, and data processing tools.	7	50% get info	60% get info

단계 6: 보정으로 기대되는 품질속성 응답수준의 utility 결정

- 각 AS의 효과(Expected Responses)가 해당 AS에 관련된 시나리오들에 대하여 결정되었다.
- 따라서, 각 시나리오 별로 현재 응답수준과 기대응답수준을 참조하여 utility 값을 결정

단계 6: 보정으로 기대되는 품질속성 응답수준의 utility 결정

Table 6: Architectural Strategies and Their Expected Utility

AS	AS Name	Scenarios Affected	Current Utility	Expected Utility
1	Order persistence on submission	3	70	90
		5	70	100
		6	80	100
2	Order chunking	8	20	60
3	Order bundling	9	50	80
		10	70	65
4	Order segmentation	4	80	90
5	Order reassignment	1	80	92
6	Order retry	4	80	85
7	Forced order completion	1	80	87
8	<u>Failed order notification</u>	6	80	85
		7	70	90
9	Granule-level order tracking	6	80	90
		7	70	95
10	Links to user information	7	70	75

단계 7: 아키텍처 전략(AS)이 주는 총이익 계산

- AS_i 이 주는 총이익을 Eq. 1 을 사용하여 계산:

$$B_i = \sum_j (b_{i,j} \times W_j) \quad \text{Eq. 1}$$

$b_{i,j}$: benefit of AS_i for scenario j

w_j : weight for scenario j (= 단계 3의 votes)

(j 는 AS_i 가 영향을 주는 시나리오의 인덱스이다.)

단계 7: 아키텍처 전략(AS)이 주는 총이익 계산

Table 7: Total Benefit of Architectural Strategies

Strategy	Scenario Affected	Scenario Weight	Raw AS Benefit	Normalized AS Benefit	Total AS Benefit
1	3	15	20	300	
1	5	15	30	450	
1	6	10	20	200	950
2	8	5	40	200	200
3	9	10	30	300	
3	10	5	-5	-25	275
4	4	10	10	100	100
5	1	10	12	120	120
6	4	10	5	50	50
7	1	10	7	70	70
8	6	10	5	50	
8	<u>7</u>	5	20	100	150
9	6	10	10	100	
9	<u>7</u>	5	25	125	225
10	<u>7</u>	5	5	25	25

= Expected Benefit - Current Benefit

x

=

단계 8. 투자수익률(ROI)을 기초로 아키텍처 전략 선정

- 각 AS에 대하여 **비용**을 추정
- 그러면, AS별 ROI가 계산되고, 전체 ASs에 대하여 순위를 결정할 수 있음

$$\text{ROI} = \frac{\text{Benefit}}{\text{Cost}}$$

단계 8. 투자수익률(ROI)을 기초로 아키텍처 전략 선정

Table 8: ROI of Architectural Strategies

AS	Cost	Total AS Benefit	AS ROI	AS Rank
1	1200	950	0.79	1
2	400	200	0.5	3
3	400	275	0.69	2
4	200	100	0.5	3
5	400	120	0.3	7
6	200	50	0.25	8
7	200	70	0.35	6
8	300	150	0.5	3
9	1000	225	0.22	10
10	100	25	0.25	8

ROI(AS8)

$$= \frac{150}{300}$$

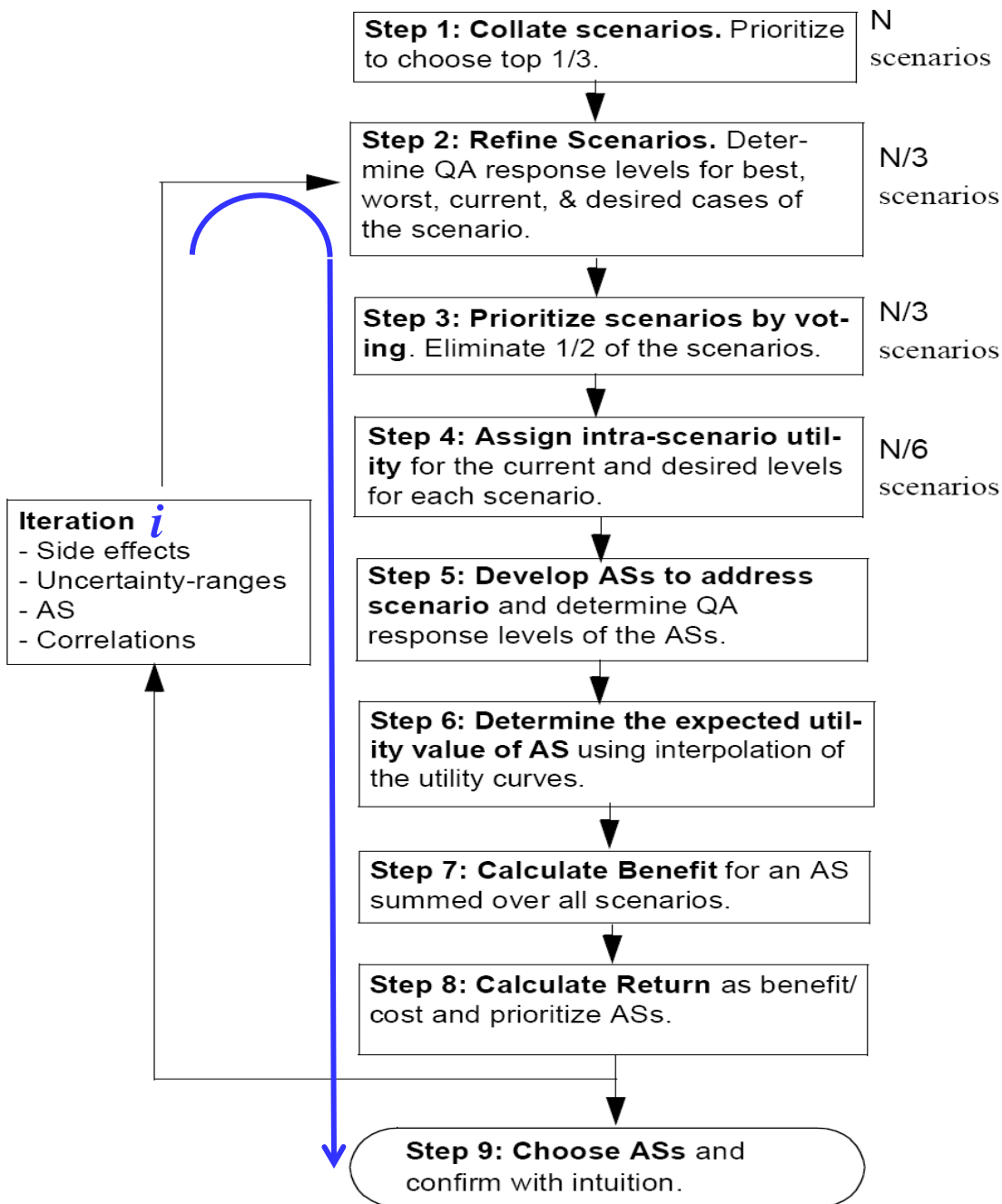
$$= 0.5$$

단계 9. 직관을 이용하여 결과 확인

- Consider whether chosen ASs seem to align with the organization's business goals. If not, consider issues that may have been overlooked. If significant issues exist, **perform another iteration** of these steps.
- 문제점
 - 직관이 잘못되었을 경우, 잘못된 직관을 받아들이게 됨.
 - 직관이 옳은지 그른지 알 수 없으므로, **CBAM**의 결과를 받아들여야 할지 알 수 없음.
 - **CBAM**을 ‘직관’의 보조수단으로 격하시킴.
 - **Question: Isn't this, though, the way we discover new facts?**
 - **Answer:** 새로운 발견에서 직관의 역할은 탐구의 방향을 제시하고, 실험을 위한 가설을 제공하나, **CBAM**에서는 그것을 **CBAM** 방법을 성공적으로 종료시키는 판단기준으로 하는데 문제가 있음.
- ☞ **CBAM**을 직관의 검증수단으로 사용한다면 문제가 없으나, 직관을 **CBAM**의 검증수단으로 사용하는 데에는 문제가 있음.

Iteration II

- Step 9의 문제점 이외에 Iteration I은 어떤 문제가 있는가?
- 위험요인, 위험의 발생확률, 불확실성



- AS 별 risk가 식별됨
- 각 risk에 대해서 다음 사항들을 평가
 - 발생확률
 - 발생할 경우의 impact
 - 응답수준의 변경
 - Utility 값의 변경
 - 완화전략(mitigation strategies)

Figure 2: Process Flow Diagram for the CBAM

Table 9: Risks for the Various Architectural Strategies

ID	<u>Risk/ AS Description</u>	Probability	Impact	Mitigation Strategy
AS1: Order persistence on submission				
T4	Selected database management system (DBMS) may become unavailable. (T4)	0.4	High, cost increase of up to 50%	Replace DBMS; cost increase equivalent to 50% source lines of code (SLOC).
T5	Ability to store complex order objects in DBMS may not support spatial attributes as advertised. (T5)	0.5	Medium, operability and user satisfaction decrease	Do not support spatial attributes.
T6	DBMS vendor change may significantly alter new object model performance characteristics. (T6)	0.3	High, cost increase of up to 70%	Write code to use non-object elements of DBMS; cost increase equivalent to 70% SLOC.
S1	New object interface to DBMS may require far more code than anticipated to develop strategy. (S1)	0.6	High, cost increase by up to 70%	none
S2	DBMS vendor may issue patch that must be incorporated late during strategy testing. (S2)	0.7	Low, cost increase by 25%	none
AS2: Order chunking		none		
AS3: Order bundling		none		

Table 9: Risks for the Various Architectural Strategies (Continued)

ID	Risk/ AS Description	Probability	Impact	Mitigation Strategy
AS4: Order segmentation				
T2	The implementation could result in extensive operations overhead to segment orders, thus reducing operability.	0.5	High, operability decrease	Redesign interface; 50% new SLOC.
AS5: Order reassignment				
T1	Fewer than expected hung orders may benefit from the ability to reassign the order. (T1)	0.7	Medium, 3% hung	Place more emphasis on forced order completion.
AS6: Order retry				
S3	S3: Benefit of retrying orders may change due to changing archive media over next two years.	0.3	Low, current may change to goal	none
AS7: Forced order completion				
AS8: Failed order notification				
AS9: Granule level order tracking				
T3	Team may not have adequate user interface experience to provide an acceptable user interface without extensive iterations with users. (T3)	0.8	High, cost increase of up to 50%	Add specialized UI expertise; cost increase equivalent to 20% SLOC.
AS10: Links to user information				
T7	Users may decide they want links to information not currently planned in strategy.	0.5	Low to Medium, cost increase by 20 to 50%	Write code to support additional information; cost increase of 20 to 50%, depending on number of items.
C1	Web server licensing scheme may change to require a per-user license.	0.4	High, cost could quadruple	Modify system to use alternate Web server; cost equivalent to 200% of SLOC.
C2	Advanced Web development expertise may be required.	0.5	Medium, cost increase of 50%	Add specialized UI expertise; cost increase equivalent to 20% SLOC.

- Risk의 유형(Type):
 - I. 고려되는 전략의 **cost** 추정에 영향을 주는 risk
 - II. AS의 시나리오의 **stimulus-response** 특징 혹은 AS의 **utility** 값에 영향을 주는 risk
 - III. 다른 시나리오 혹은 QA의 **stimulus-response** 특징에 영향을 주는 risk (AS의 **side effect**에 해당).
 - IV. 프로젝트관리, **schedule** 에 관련된 risk
- 경우에 따라, risk 는 주효과(primary effect)와 **2차적효과 (secondary effect)**를 가진 것으로 분류
(risk ID 에 “s” 를 붙여서 표시).

Risk를 cost-benefit으로 어떻게 반영하는가?

Table 10: Risks and Their Categories

ID	Type	Comment
T4	I	Cost
T5	III	Benefit - Side effect
T6	I	Cost
S1	I	Cost
S2	I	Cost
T2	III	Benefit - Side effect
T1	II	Benefit
S3	II	Benefit
T3	I	Cost
<u>T3s</u>	<u>IV</u>	<u>Schedule</u>
T7	I	Cost
C1	I	Cost
C2	I	Cost
<u>C2s</u>	<u>IV</u>	<u>Schedule</u>

- The impacts of the risks and the probabilities of the events are explicitly determined by the stakeholders, and
 - Expected risk-based delta cost (the maximum cost ↑)
 - Delta benefit (the minimum benefit ↓)are calculated.
- Based on these values, the benefits and costs are re-estimated using a *pessimistic estimate*.
- In addition, for the cost estimate, a 10% factor is added to account for the error in the underlying cost estimation process.
- The minimum and maximum values are the end points of the confidence intervals.
- The expected return ($E(\text{Return})$) is the mean of the minimum and maximum return values.

$$\text{Return}_{\text{Min}} (\text{AS8}) = 135/330 = 0.409$$

$$\text{Return}_{\text{Max}} (\text{AS8}) = 165/270 = 0.611$$

AS	Cost	Total AS Benefit	AS ROI	AS Rank
8	300	150	0.5	3

Table 11: Uncertainty Estimates Using Ranges



**Revised
Ranks**

AS #	Cost		Benefit		Return		E(Return)	RANK
	Min	Max	Min	Max	Min	Max		
1	1200	7152.75	850	950	0.119	0.79	0.454	4
2	360	440	180	220	0.409	0.611	0.510	2
3	360	440	247.5	302.5	0.563	0.84	0.701	1
4	180	220	50	100	0.227	0.556	0.391	5
5	360	440	80	120	0.182	0.333	0.258	7
6	180	220	0	50	0	0.278	0.139	10
7	180	220	63	77	0.286	0.428	0.357	6
8	270	330	135	165	(0.409 + 0.611) / 2 = 0.510		0.510	2
9	900	1650	202.5	247.5	0.123	0.275	0.191	8
10	90	440	22.5	27.5	0.051	0.306	0.166	9

$$\left(\frac{B_{\max}}{C_{\min}} + \frac{B_{\min}}{C_{\max}} \right) \times \frac{1}{2} = \frac{C_{\min} B_{\min} + C_{\max} B_{\max}}{2C_{\min} C_{\max}}$$

- CBAM의 마지막 단계인 단계 9에서는 표 `12-13의 결과를 확인
 - 이를 위하여 예를 들어 두 개의 AS간의 쌍대비교 (pairwise comparison) 방법을 사용할 수도 있다.
 - 이 방법은 여러 개를 한꺼번에 비교하는 것은 매우 어려우나, 두 개의 대상을 비교하는 것은 비교적 정확히 할 수 있기 때문에 유용한 방법이다.
 - 즉, AS_i 가 모든 다른 AS들 AS_j 에 대하여 $AS_i > AS_j$ 인 확률을 얻을 수 있다면, 그 평균값의 크기로 우선순위를 결정한다.

12.3 CBAM 기법을 사용한 아키텍처 평가

- 설계가 주어진 요구사항을 충족시키는지에 대한 분석은 설계과정에 달성된 것으로 가정할 수 있음
- CBAM은 아키텍처 평가의 Tool을 제공
 - ASs에 대한 평가
 -  그러나 아키텍처 전체에 대한 직접적인 평가를 주지는 않음
-  개별적인 전략의 가치를 합산할 수 있는 방법이 필요.

예) 개발예산이 1,500 인 경우, 표 12-14의 전략들 가운데 이 예산으로 적용할 수 있는 전략의 집합은

- 우선 순위에 따라 {AS3, AS2, AS8, AS4, AS7}
 - 각각의 비용이 400, 400, 300, 200, 200
 - AS1은 우선순위가 4이지만 비용이 1,200 으로 채택되지 못함.

- **CBAM**을 활용하여 아키텍처를 평가하는 방법은 대안 아키텍처들 간에 선정된 **ASs**의 가치의 합을 평가결과로 보는 것
 - 그러나 간섭효과를 반영하는 방법에 대한 연구가 필요함
(예, AS1의 benefit이 AS2의 benefit과 단순히 합하여 지는 것인가?)
=> [Asundi 01]

Lab 5. 아키텍처 평가

- 아키텍처 문서의 **G**와 **H**를 작성

Step 1. 평가 방법 제시

- 평가 관점(= 분석관점)들 제시
- 분석데이터의 종합 방법 제시
- 결과의 형태 제시

Step 2. 설계된 아키텍처 평가

Step 3. 대안 아키텍처 제시

Step 4. 대안 아키텍처의 평가

Questions?