Economic Evaluation through Cost-Benefit Analysis (CBAM)

Outline

1. **Decision-Making Context**
2. **Basis for Economic Analyses**
3. **Utility-Response Curves**
4. **Putting Theory into Practice: The CBAM**
5. **Case Study: The NASA ECS Project**
6. **Summary**

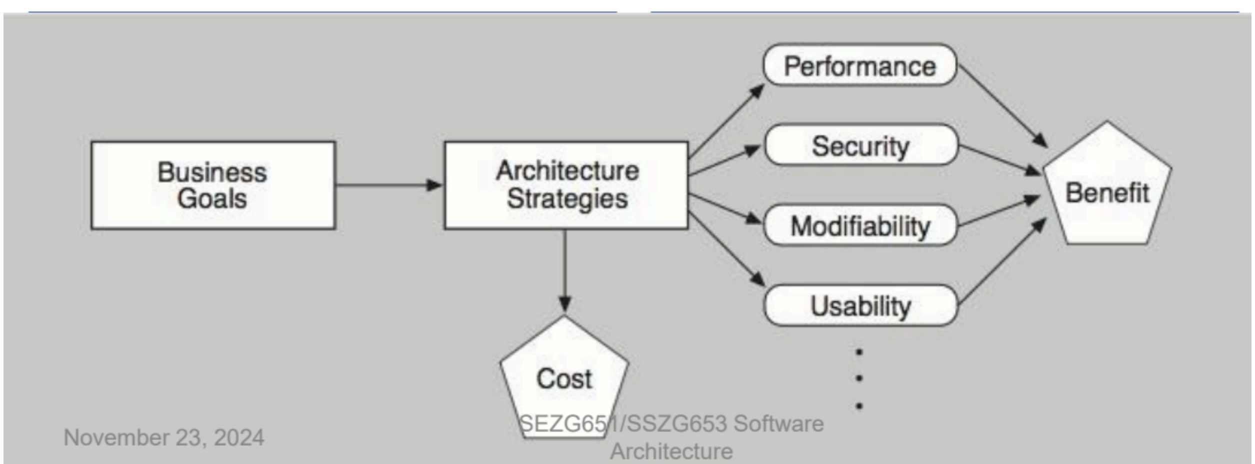
---

1. Decision-Making Context

In software architecture, every architectural decision has economic implications, influencing the project's overall cost and benefit.

- **Example:** Choosing to use redundant hardware for higher availability incurs additional costs but may significantly enhance system reliability.
- **Use Case:** When building a high-availability system, the architect compares different approaches to determine which provides the best value at an acceptable cost.

Diagram: Decision-making Context



2. Basis for Economic Analyses

Economic analyses begin with scenarios derived from requirements and architectural evaluations. Each scenario reflects different architectural strategies with associated costs and quality attribute impacts.

1. **Scenario Generation:** Scenarios represent different project requirements (e.g., performance, availability).
2. **Utility Assignment:** Each scenario is given a utility score based on its value to stakeholders.

- **Example:** A high-performance scenario might have a high utility for a real-time application.

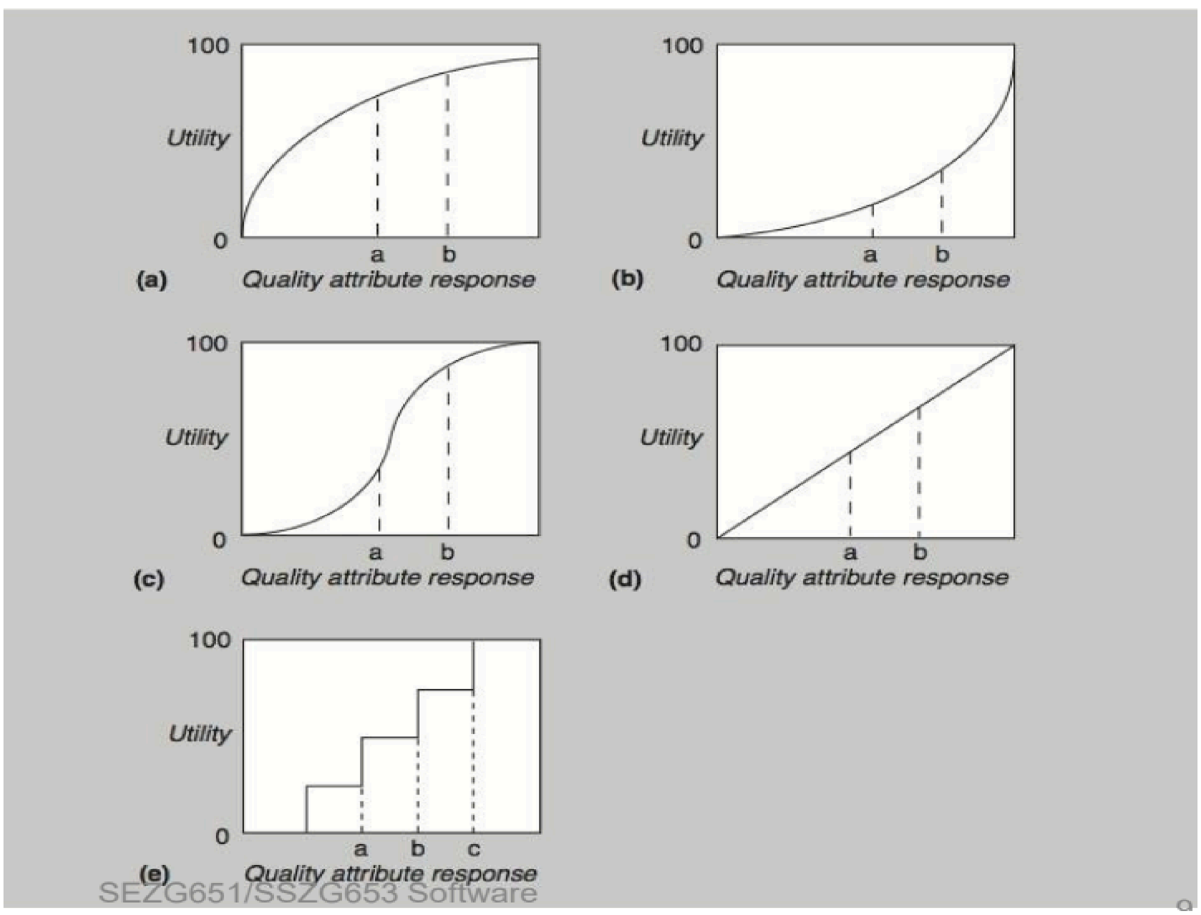
---

3. Utility-Response Curves

Utility-response curves represent the relationship between architectural decisions and their utility (value) to stakeholders. The curve helps in comparing different quality attributes, like performance vs. modifiability.

- **Example:** Stakeholders may value "99.999% availability" highly, but slightly lower availability might still be acceptable if it significantly reduces cost.

Graph: Example Utility-Response Curves



Weighting Scenarios

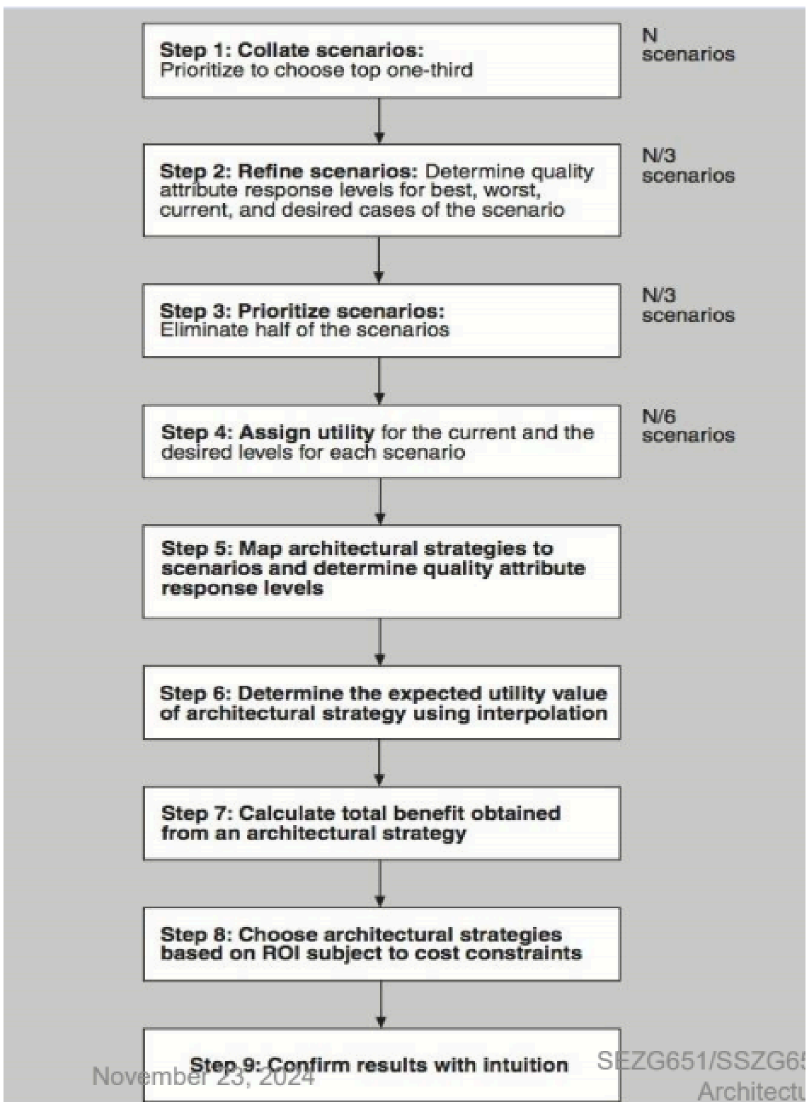
Scenarios are prioritized based on their importance to stakeholders.

- **Example:** A scenario for high security might be weighted more heavily than a scenario for ease of maintenance if security is critical for the project.

---

4. Putting Theory into Practice: The CBAM Process

The CBAM process includes several steps to assess the costs and benefits of different architectural strategies. Each step is illustrated with practical examples.

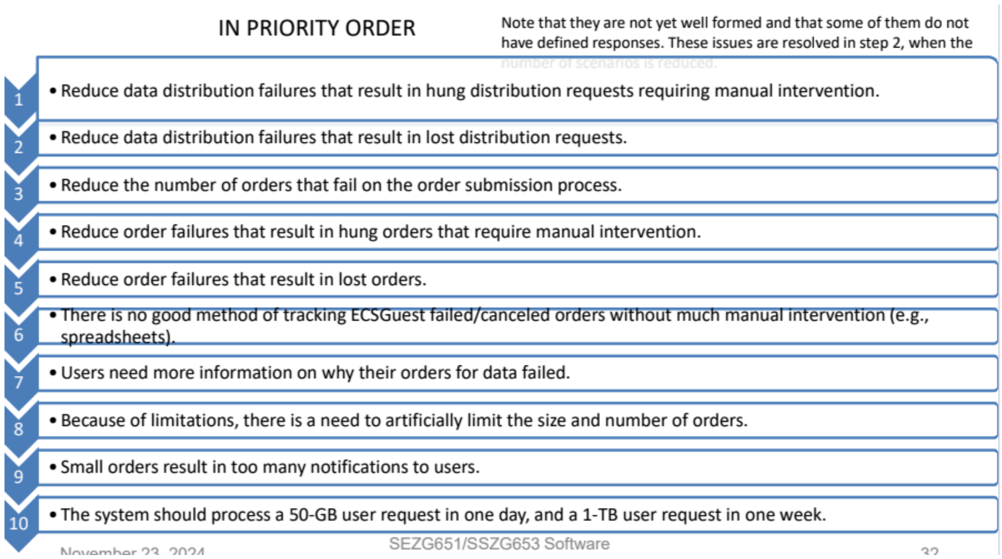


Step 1: Collate Scenarios

Gather all relevant scenarios and prioritize them.

- **Example:** List scenarios such as "reduce data processing failures" or "improve system response time."

Table: Scenarios List and Priorities



Step 2: Refine Scenarios

Refine scenarios by setting specific goals for best-case, worst-case, current, and desired responses.

- **Example:** The desired response time for a system might be 0.5 seconds, while the worst-case response could be 2 seconds.

Table: Scenario Refinement with Goals

Scenario	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 information	50% get information	100% get information	100% get information
8	50% limited	30% limited	0% limited	0% limited
9	1/granule	1/granule	1/100 granules	1/1,000 granules
10	< 50% meet goal	60% meet goal	80% meet goal	> 90% meet goal

Step 3: Prioritize Scenarios

Stakeholders vote on scenario priorities to assign weights.

- **Example:** Allocate 100 points among scenarios to represent their importance.

Table: Scenario Prioritization

Scenario	Votes	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 information	50% get information	100% get information	100% get information
8	5	50% limited	30% limited	0% limited	0% limited
9	10	1/granule	1/granule	1/100 granules	1/1,000 granules
10	5	< 50% meet goal	60% meet goal	80% meet goal	> 90% meet goal

Step 4: Assign Utility Scores

Determine the utility scores for each response level in each scenario.

- **Example:** A response time of 0.5 seconds might have a utility score of 100, while a response time of 2 seconds might score 20.

Table: Utility Scores for Response Levels

Scenario	Worst Case	Current	Desired	Best Case
Scenario #17: Response to user input	12 seconds	1.5 seconds	0.5 seconds	0.1 seconds
	Utility 5	Utility 50	Utility 80	Utility 85

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	50	50	80	90

Step 5: Develop Architectural Strategies

Identify architectural strategies and their expected impact on each scenario.

- **Example:** One strategy might be to implement load balancing to improve response time.

Table: Expected Quality Attribute Response Levels

Strategy	Name	Description	Scenarios Affected	Current Response	Expected Response
1	Order persistence on submission	Store an order as soon as it arrives in the 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
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 detailed status of each item; user notification occurs only if operator okays notification; the operator may edit notification.	6	25% need help	20% need help
			7	50% get information	90% get information
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 information	95% get information
10	Links to user information	An operator can quickly locate a user's contact information. Server will access SDSRV information to determine any data restrictions that might apply and will route orders/order segments to appropriate distribution capabilities, including DDIST, PDS, external subsetters and data processing tools, etc.	7	50% get information	60% get information

Step 6: Calculate Utility of Expected Responses

Use interpolation to calculate the utility of each expected response level.

- **Example Calculation:** For a response time of 0.7 seconds, interpolate utility between 50 (for 1.0 seconds) and 80 (for 0.5 seconds).

Table: Utility of Expected Responses

- Using the elicited utility values (that form a utility curve), determine the utility of the expected quality attribute response level for the architectural strategy.
- Do this for each relevant quality attribute enumerated in step 3.
- For example, if we are considering a new architectural strategy that would result in a response time of 0.7 seconds, we would assign a utility proportionately between 50 (which it exceeds) and 80 (which it doesn't exceed).
- The formula for interpolation between two data points (x_a, y_a) and (x_b, y_b) is:

$$y = y_a + (y_b - y_a)\frac{(x - x_a)}{(x_b - x_a)}$$

- For us, the *x* values are the quality attribute response levels and the *y* values are the utility values. So, employing this formula, the utility value of a 0.7-second response time is 74 .

Strategy	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	Failed order notification	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

Step 7: Calculate Total Benefit

Calculate the total benefit of each architectural strategy by summing the benefits across all scenarios.

- **Formula:** Bi=∑(bi,j×Wj)Bi = \sum (bi,j \times Wj)Bi=∑(bi,j×Wj)

Table: Total Benefit Calculation

Strategy	Scenario Affected	Scenario Weight	Raw Architectural Strategy Benefit	Normalized Architectural Strategy Benefit	Total Architectural Strategy 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	7	5	20	100	150
9	6	10	10	100	
9	7	5	25	125	225
10	7	5	5	25	25

Step 8: Choose Architectural Strategies Based on VFC

Evaluate strategies based on their Value for Cost (VFC) ratio, choosing those with the highest VFC scores within budget constraints.

- **Example:** If one strategy has a VFC score of 1.5 and another 2.0, prioritize the latter for its higher cost-effectiveness.

Table: VFC Scores and Ranking

Strategy	Cost	Total Strategy Benefit	Strategy VFC	Strategy 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

Step 9: Confirm Results with Stakeholders

Verify that the chosen strategies align with stakeholder expectations and business goals.

---

5. Case Study: The NASA ECS Project

NASA's Earth Observing System Data Information System (ECS) processes large volumes of environmental data. To maintain system performance and availability within budget, the ECS project team used CBAM to prioritize architectural strategies.

1. **Scenario Example:** Reduce data distribution failures to ensure scientists receive accurate data on time.
2. **Steps in Practice:** The ECS project manager prioritized scenarios, refined them, and applied utility scores, finally choosing strategies based on VFC.

Diagram: ECS Project Case Study Workflow

---

6. Summary

The CBAM process allows architects and stakeholders to make informed decisions by evaluating the costs and benefits of architectural strategies. By using structured steps and utility-response curves, teams can ensure that decisions align with project goals and stakeholder priorities.