VVIT
VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY

SOFTWARE ARCHITECTURE AND DESIGN PATTERNS

UNIT-II

Dr. T. K. RAO

VVIT

VVIT
VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY

contents

- **ANALYZING ARCHITECTURES:**
  - Architecture Evaluation - ATAM
  - Architecture design decision making - CBAM
- **MOVING FROM ONE SYSTEM TO MANY:**
  - Software Product Lines
  - Building systems from off the shelf components
  - Software architecture in future

Architecture Evaluation - ATAM

- A thorough and comprehensive way to evaluate a s/w architecture is Architecture Trade off Analysis Method (ATAM)
- It reveals how well the architecture satisfies particular quality goals
- It provides insight into how quality goals interact – that is how they trade off
- ATAM is designed to elicit the business goals for the system as well as for the architecture

Participants in the ATAM

The evaluation team.

- External to the project whose architecture is being evaluated.
- Three to five people; a single person may adopt several roles in an ATAM.
- They need to be recognized as competent, unbiased outsiders.

Project decision makers.

- These people are empowered to speak for the development project or have the authority to mandate changes to it.
- They usually include the project manager, and if there is an identifiable customer who is footing the bill for the development, he or she may be present (or represented) as well.
- The architect is always included – the architect must willingly participate.

Architecture stakeholders.

- Stakeholders have a vested interest in the architecture performing as advertised.
- Stakeholders include developers, testers, integrators, maintainers, performance engineers, users, builders of systems interacting with the one under consideration, and, possibly, others.
- Their job is to articulate the specific quality attribute goals that the architecture should meet.
- Expect to enlist 12 to 15 stakeholders for the evaluation of a large enterprise-critical architecture.

Roles of Evaluation-Team

Role	Responsibilities	Desirable characteristics
Team Leader	Sets up the evaluation; coordinates with client, making sure client's needs are met; establishes evaluation contract; forms evaluation team; sees that final report is produced and delivered (although the writing may be delegated)	Well-organized, with managerial skills; good at interacting with client; able to meet deadlines
Evaluation Leader	Runs evaluation; facilitates elicitation of scenarios; administers scenario selection/prioritization process; facilitates evaluation of scenarios against architecture; facilitates onsite analysis	Comfortable in front of audience; excellent facilitation skills; good understanding of architectural issues; practiced in architecture evaluations; able to tell when protracted discussion is leading to a valuable discovery or when it is pointless and should be re-directed

Role	Responsibilities	Desirable characteristics
Scenario Scribe	Writes scenarios on flipchart or whiteboard during scenario elicitation; captures agreed-on wording of each scenario, halting discussion until exact wording is captured	Good handwriting; stickler about not moving on before an idea (scenario) is captured; can absorb and distill the essence of technical discussions
Proceedings Scribe	Captures proceedings in electronic form on laptop or workstation, raw scenarios, issue(s) that motivate each scenario (often lost in the wording of the scenario itself), and resolution of each scenario when applied to architecture(s); also generates a printed list of adopted scenarios for handout to all participants	Good, fast typist; well organized for rapid recall of information; good understanding of architectural issues; able to assimilate technical issues quickly; unafraid to interrupt the flow of discussion (at opportune times) to test understanding of an issue so that appropriate information is captured
Timekeeper	Helps evaluation leader stay on schedule; helps control amount of time devoted to each scenario during the evaluation phase	Willing to interrupt discussion to call time

Role	Responsibilities	Desirable characteristics
Process Observer	Keeps notes on how evaluation process could be improved or deviated from; usually keeps silent but may make discreet process-based suggestions to the evaluation leader during the evaluation; after evaluation, reports on how the process went and lessons learned for future improvement; also responsible for reporting experience to architecture evaluation team at large	Thoughtful observer; knowledgeable in the evaluation process; should have previous experience in the architecture evaluation method
Process Enforcer	Helps evaluation leader remember and carry out the steps of the evaluation method	Fluent in the steps of the method, and willing and able to provide discreet guidance to the evaluation leader
Questioner	Raise issues of architectural interest that stakeholders may not have thought of	Good architectural insights; good insights into needs of stakeholders; experience with systems in similar domains; unafraid to bring up contentious issues and pursue them; familiar with attributes of concern

Project decision makers

- These people are empowered to speak for the development project or have the authority to mandate changes to it
- They usually include the project manager, and, if there is an identifiable customer who is footing the bill for the development
- The architect is always included—a cardinal rule of architecture evaluation is that the architect must willingly participate
- Finally, the person commissioning the evaluation is usually empowered to speak for the development project

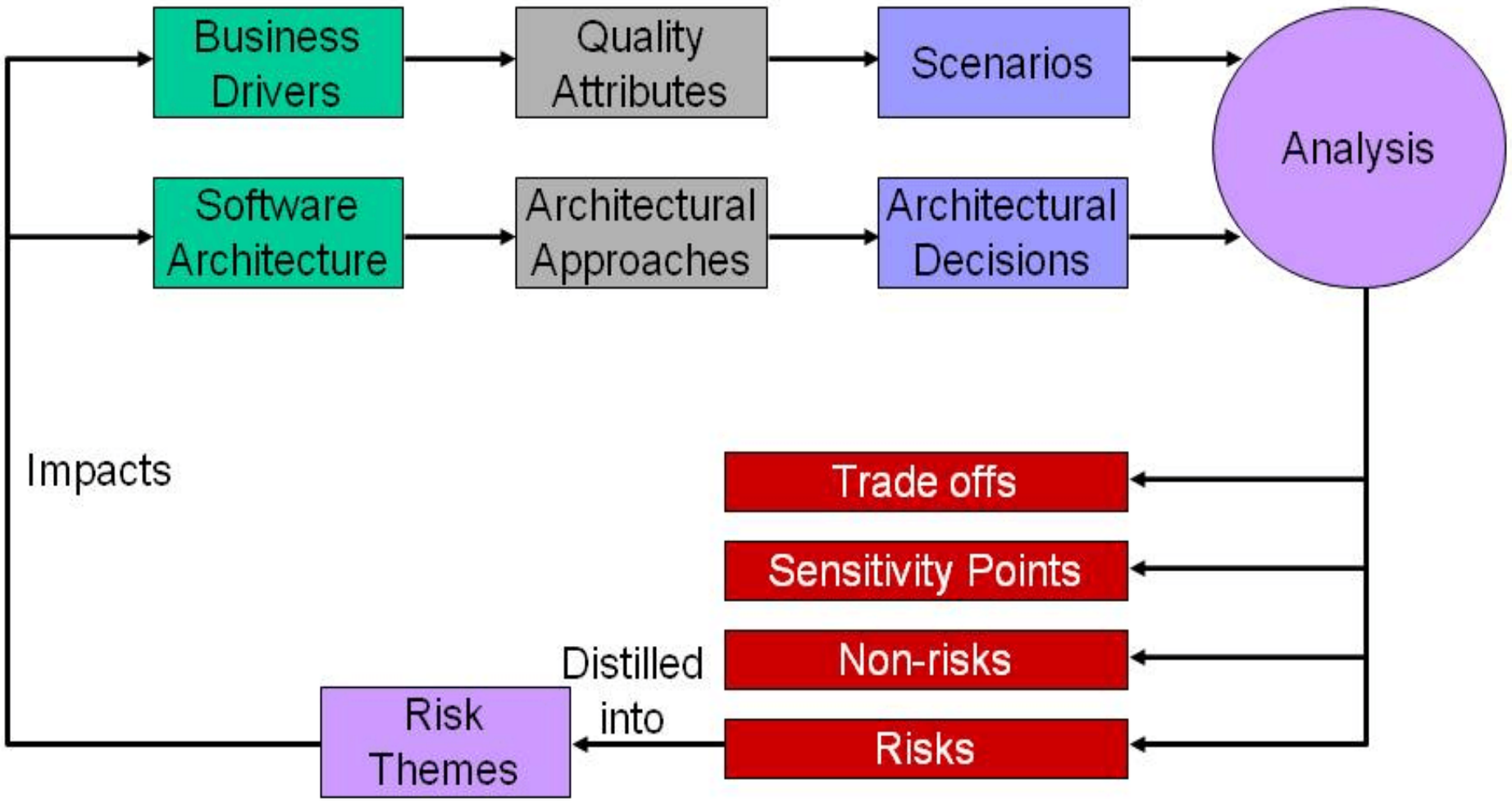
Architecture stakeholders

- Stakeholders have interest in architecture being evaluated
- Stakeholders are developers, testers, maintainers, users...
- Their job during an evaluation is to articulate the specific quality attribute goals that the architecture should meet
- A rule of thumb is that we should expect to enlist the services of twelve to fifteen stakeholders for the evaluation

ATAM method consists of 9 steps:

1. Present the ATAM.
2. Present business drivers.
3. Present architecture.
4. Identify architectural approaches.
5. Generate quality attribute utility tree.
6. Analyze architectural approaches.
7. Brainstorm and prioritize scenarios.
8. Analyze architectural approaches.
9. Present results.

Architecture Trade off Analysis Method



Phases of the ATAM

Phase 0

- activity: preparation
- participants: evaluation team leadership and key project decision makers
- typical duration: proceeds informally as required, perhaps over a few weeks

Phase 1

- activity: evaluation (steps 1-6)
- participants: evaluation team and project decision makers
- typical duration: 1 day followed by a hiatus of 2 to 3 weeks

Phase 2

- activity: evaluation (steps 7-9)
- participants: evaluation team, project decision makers and stakeholders
- typical duration: 2 days

Phase 3

- activity: follow-up
- participants: evaluation team and evaluation client
- typical duration: 1 week

Outputs of the ATAM

- An ATAM-based evaluation will produce at least the following outputs:
  - A concise presentation of the architecture
    - Architecture documentation is often thought to consist of the object model, a list of interfaces and their signatures and so on
    - An architectural presentation that is both concise and, usually, understandable that can be presented in one hour
  - Articulation of the business goals
    - The business goals presented in the ATAM are being seen by some of the development team for the first time
  - Quality requirements in terms of a collection of scenarios
    - Some of the important quality requirements are captured in the form of scenarios

Contd.

- Mapping of architectural decisions to quality requirements
  - Architectural decisions can be interpreted in terms of the qualities that they support or hinder
  - For each quality scenario examined during an ATAM, those architectural decisions that help to achieve it are determined
- A set of identified sensitivity and tradeoff points
  - These are architectural decisions that have a marked effect on one or more quality attributes
    - Adopting a backup database affects reliability (positively), and so it is a sensitivity point with respect to reliability
  - However, keeping the backup current consumes system resources and so affects performance negatively
    - Hence, it is a tradeoff point between reliability and performance.

Contd.

- A set of risks and non-risks
  - A risk is defined in ATAM as architectural decision that may lead to undesirable consequences considering quality attribute requirements
  - A nonrisk is an architectural decision that is deemed safe
  - The identified risks can form the basis for an architectural risk mitigation plan
- A set of risk themes
  - At the end, the evaluation team will examine the full set of discovered risks to look for over-arching themes that identify systemic weaknesses in the architecture, process, and team
    - If left untreated, these risk themes will threaten the project's business goals

- The outputs are used to build final written report that
  - recaps the method, summarizes the proceedings, captures the scenarios and their analysis, and catalogs the findings
- There are intangible results of an ATAM-based evaluation
  - A palpable sense of community on the part of the stakeholders, open communication channels between the architect and the stakeholders, a better overall understanding on the part of all participants of the architecture and its strengths and weaknesses
  - While these results are hard to measure, they are no less important than the others and often are the longest-lasting

Phases of the ATAM

- Activities in ATAM are spread out over four phases;
  - Phase 0 - "Partnership and Preparation“
    - The evaluation team leadership and key project decision makers informally meet to work out the details of the exercise
    - Project representatives brief the evaluators about project
    - The two groups agree on the time and place of meetings , who brings the flipcharts, and who supplies the donuts and coffee
    - They also agree on a preliminary list of stakeholders and they negotiate on when the final report is to be delivered and to whom
    - They work out delivery to the evaluation team of whatever architectural documentation exists and may be useful
    - Evaluation team leader explains what information manager and architect will be expected to show during phase 1

- Phase 1 and 2 – “Evaluation Phase”
  - By now:
    - the evaluation team will have studied the architecture documentation
    - will have a good idea of what the system is about
    - the overall architectural approaches taken, and
    - the quality attributes that are of paramount importance.
  - In phase 1, the evaluation team meets project decision makers to begin information gathering and analysis
  - In phase 2, the architecture's stakeholders join the proceedings and analysis continues, typically for two days

- Phase 3 – “Follow-up Phase”
  - The evaluation team produces and delivers a written final report
  - The essence of this phase, however, is team self-examination and improvement
  - They study the surveys handed out to participants during phase 1 and phase 2, and the process observer makes his or her report
  - Team members look for improvements in how they carry out their functions so that the next evaluation can be smoother or more effective
  - The team catalogs how much effort was spent during the evaluation, on the part of each of the three participating groups

Phase	Activity	Participants	Typical Duration
0	Partnership and preparation	Evaluation team leadership and key project decision makers	Proceeds informally as required, perhaps over a few weeks
1	Evaluation	Evaluation team and project decision makers	1 day followed by a hiatus of 2 to 3 weeks
2	Evaluation (continued)	Evaluation team, project decision makers, and stakeholders	2 days
3	Follow-up	Evaluation team and evaluation client	1 week

Steps in ATAM Evaluation Phases

- The ATAM Evaluation Phases consists of 9 steps
  - Steps 1 to 6 are carried out in **Phase 1** and 7 to 9 in **Phase 2**
- Evaluation steps are nominally carried out in sequential order
- Some times team returns briefly to an earlier step, jumps forward to a later step, or iterates among steps, as the need dictates
- Step 1—Present the ATAM
  - The first step calls for the evaluation leader to present the ATAM to the assembled project representatives
  - Using a standard presentation, the leader will describe the ATAM steps in brief and the outputs of the evaluation.

- Step 2—Present Business Drivers
  - A project decision maker (ideally the project manager or the system's customer) presents a system overview from a business perspective including following points;

1. The system's most important functions
2. Any relevant technical, managerial, economic, or political constraints
3. The business goals and context as they relate to the project
4. The major stakeholders
5. The architectural drivers **(that is, the major quality attribute goals that shape the architecture)**

- Step 3—Present Architecture
  - The lead architect makes a presentation describing the architecture at an appropriate level of detail
    - how much of the architecture has been designed and documented
    - how much time is available
    - the nature of the functional and quality requirements
  - Most important, the architect describes the architectural approaches (patterns) used to meet the requirements
  - During the presentation, the evaluation team asks for:
    - clarification based on their phase 0 examination documentation and
    - their knowledge of the business drivers

- Step 4—Identify Architectural Approaches
  - The ATAM focuses on analyzing an architecture by understanding its architectural approaches
  - The evaluation team will have a good idea of what patterns and approaches the architect used in designing the system
  - In that step, architect is asked to explicitly name the patterns & approaches used, but the team should also be adept at spotting ones not mentioned
  - In this short step, the evaluation team simply catalogs the patterns and approaches that are evident
  - The list is publicly captured by the scribe for all to see and will serve as the basis for later analysis

- Step 5—Generate Quality Attribute Utility Tree
  - An architecture is either suitable or unsuitable with respect to its ability to deliver particular quality attributes in the system being built
  - In this step, the quality attribute goals are articulated in detail via a mechanism known as the utility tree
  - The evaluation team works with project decision makers to identify, prioritize, and refine system's most important quality attribute goals
  - A utility tree begins with utility as the root node
    - Utility is an expression of the overall "goodness" of the system.
  - Quality attributes form the second level because these are the components of utility (Obtained in Step 3)
  - Under each of these quality attributes are specific quality attribute refinements e.g. data latency, transaction throughput (Performance)

- Step 6—Analyze Architectural Approaches
  - Here the evaluation team examines the highest-ranked scenarios one at a time
    - the architect is asked to explain how the architecture supports each one
  - Team members (especially the questioners) probe for architectural approaches that the architect used to carry out the scenario
  - The team documents relevant architectural decisions and identifies and catalogs their risks, nonrisks, sensitivity points, and tradeoffs.
  - For well-known approaches, the team asks how the architect overcame known weaknesses in the approach or how the architect gained assurance that the approach sufficed
  - The key is to elicit sufficient architectural information to establish some link between the architectural decisions that have been made and the quality attribute requirements that need to be satisfied

- Hiatus and Start of Phase 2
  - The evaluation team retreats to summarize what it has learned and interacts informally (usually by phone) with the architect during a hiatus of a week or two
  - More scenarios might be analyzed during this period, if desired, or questions of clarification can be resolved
  - When the project's decision makers are ready to resume and the stakeholders are assembled, phase 2 commences
  - First, step 1 is repeated so that the stakeholders understand the method and the role they are to play
  - Then the evaluation leader recaps the results of steps 2 through 6, and shares the current list of risks, nonrisks, sensitivity points, and tradeoff points

- Step 7—Brainstorm and Prioritize Scenarios
  - The purpose of scenario brainstorming is to take the pulse of the larger stakeholder community
  - The prioritized list of brainstormed scenarios is compared with those from the utility tree exercise
    - If they agree, it indicates good alignment between what the architect had in mind and what the stakeholders actually wanted
    - If they disagree, this may itself be a risk showing that there was some disagreement in goals between the stakeholders and the architect
  - The evaluation team asks the stakeholders to brainstorm scenarios that are operationally meaningful with respect to their roles
  - Once the scenarios have been collected, they must be prioritized
    - The evaluation team needs to know where to devote its limited analytical time

- Step 7—Brainstorm and Prioritize Scenarios
  - First, stakeholders are asked to merge scenarios they feel represent the same behavior or quality concern
  - Then they vote for those they feel are most important.
  - Each stakeholder is allocated a number of votes equal to 30% of the number of scenarios
  - So, if there were twenty scenarios collected, each stakeholder would be given six votes
  - Scenarios above the threshold are considered for evaluation
- Step 8—Analyze Architectural Approaches
  - Once the scenarios have been collected and prioritized, the evaluation team guides the architect in the process of carrying out the highest ranked scenarios from step 7
  - The architect explains how relevant architectural decisions contribute to realizing each one
  - In this step the evaluation team performs the same activities as in step 6, mapping the highest-ranked, newly generated scenarios onto the architectural artifacts uncovered thus far

- Step 9—Present Results
  - Finally, the collected information from the ATAM needs to be summarized and presented once again to stakeholders
  - This presentation typically takes the form of a verbal report accompanied by slides, but it might be a written report
    - The architectural approaches documented
    - The set of scenarios and their prioritization from the brainstorming
    - The utility tree
    - The risks discovered
    - The nonrisks documented
    - The sensitivity points and tradeoff points found
  - The evaluation team adds value by grouping risks into risk themes, based on some common underlying concern or systemic deficiency

Architecture design decision making - CBAM

- Cost Benefit Analysis Method (CBAM) is applicable when:
  - an organization considers a major upgrade to an existing system and they wanted to understand the utility and value for cost of making the upgrade
  - the organization wanted to choose between competing architectural strategies for the upgrade
  - new systems as well, especially for helping to choose among competing strategies

The Basis for the CBAM

- The CBAM uses scenarios as a way to concretely express and represent specific quality attributes, just as in the ATAM
- As in ATAM, scenarios are structured into three parts:
  - stimulus (an interaction with the system),
  - environment (the system's state at the time), and
  - response (the measurable quality attribute that results).
- However, there is a difference between the methods
- CBAM uses a *set* of scenarios (generated by varying the values of the responses) rather than individual scenarios as in the ATAM
- This leads to the concept of a utility-response curve.

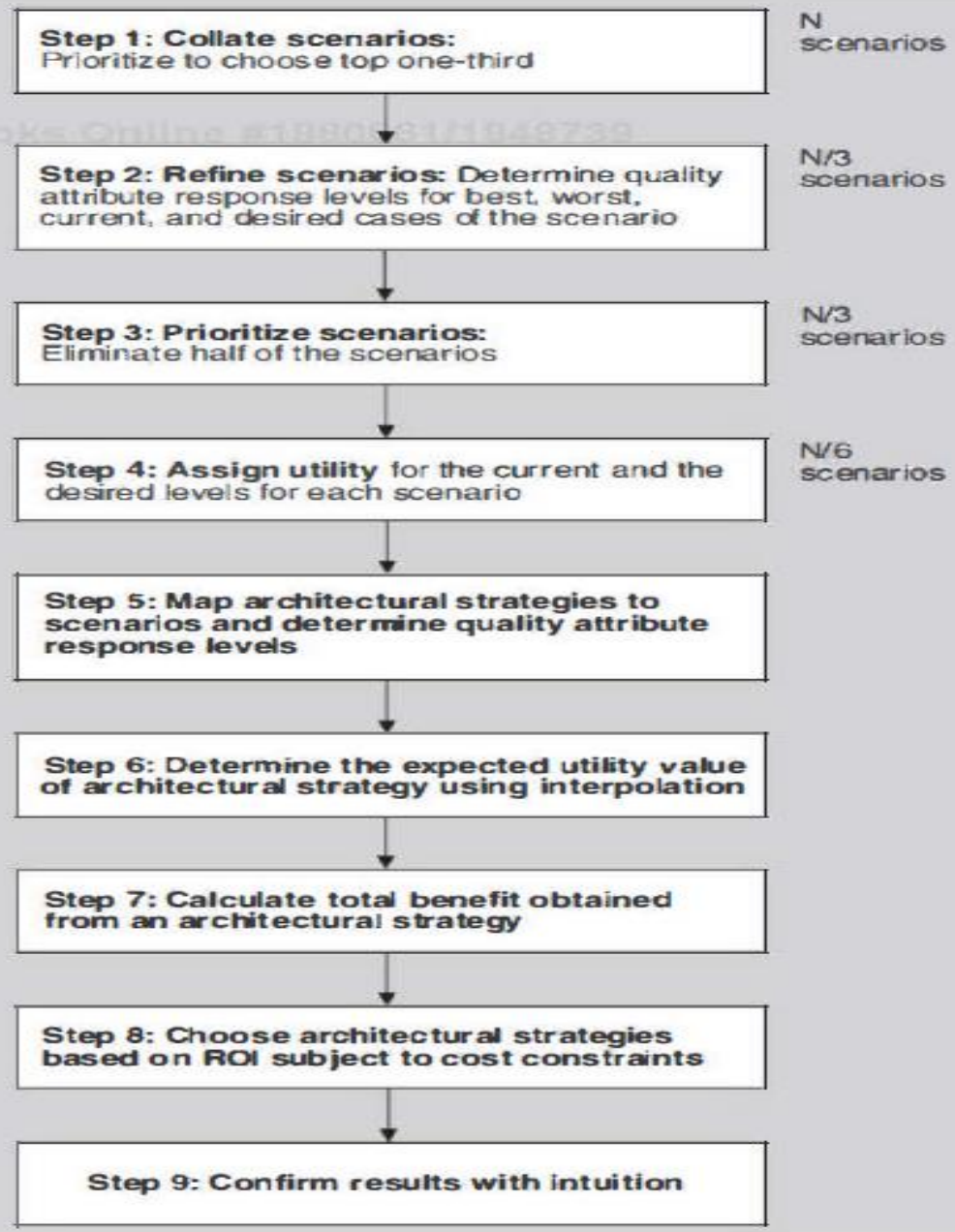


Figure 23.3. Process flow diagram for the CBAM

1 . Collate (arrange) scenarios:

- Give the stakeholders a chance to contribute new scenarios
- Ask the stakeholders to prioritize the scenarios based on satisfying the business goals of the system
- This can be an informal prioritization using a simple scheme such as "high, medium, low" to rank the scenarios
- Choose the top one-third for further study

2. Refine scenarios

- Refine the scenarios chosen in step 1, focusing on their stimulus-response measures
- Elicit the worst-case, current, desired, and best-case quality attribute response level for each scenario
- E.g. a refined performance scenario's worst-case performance for our system's response is 12 sec., the best case is 0.1 sec., our desired response is 0.5 sec.
- Our current architecture provides a response of 1 .5 seconds:

Scenario	Worst Case	Current	Desired	Best Case
Scenario #17: Response to user input	12 seconds	1.5 seconds	0.5 seconds	0.1 seconds
...				

3. Prioritize scenarios:

- Prioritize the refined scenarios, based on stakeholder votes
- Give 100 votes to each stakeholder & have them distribute the votes among the scenarios, where voting is based on desired response value for each scenario
- Total the votes and choose the top 50 percent of the scenarios for further analysis
- Assign a weight of 1.0 to the highest-rated scenario; assign the other scenarios a weight relative to the highest rated
- This becomes the weighting used in the calculation of a strategy's overall benefit.
- Make a list of the quality attributes that concern the stakeholders

4. Assign utility

- Determine the utility for each quality attribute response level (worst-case, current, desired, best-case) for scenarios from step 3.
- Capture these utility curves in a table (one row for each scenario, one column for each of the four quality attribute response levels)
- From step 2, this step would assign utility values from 1 to 1 00 for each of the latency values elicited for this scenario in step 2:

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

5. Map architectural strategies to scenarios and determine their expected quality attribute response levels

- For each architectural strategy under consideration, determine the expected quality attribute response levels that will result for each scenario.

6. Determine the utility of the expected quality attribute response levels by interpolation

- Using 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
- E.g., if we are considering a new architectural strategy that would result in a response time of 0.7 sec., we would assign this a utility proportionately between 50 (which it exceeds) and 80 (which it doesn't exceed)
- Formula for interpolation between two data points (xa, Ya) and (xb, Yb) is given by:

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

- For us, the x values are the quality attribute response levels and the y values are the utility values.

7. Calculate the total benefit obtained from an architectural strategy

- Subtract the utility value of the "current" level from the expected level and normalize it using the votes elicited in step 3
- Sum the benefit due to a particular architectural strategy across all scenarios and across all relevant quality attributes

8. Choose architectural strategies based on VFC subject to cost and schedule constraints

- Determine the cost and schedule implications of each architectural strategy
- Calculate the VFC value for each as a ratio of benefit to cost
- Rank-order the architectural strategies according to VFC and choose top ones until the budget/ schedule is exhausted.

9. Confirm results with intuition

- For the chosen architectural strategies, consider whether these seem to align with the organization's business goals
- If not, consider issues that may have been overlooked while doing this analysis
- If there are significant issues, perform another iteration of these steps.

Building Systems from Off-the-Shelf Components

- Systems are being constructed with more off-the-shelf components for economic reasons
- Their design and interaction mechanisms are not under the architecture
- System might be largely integrated from the-off-shelf components
- Even then the quality attributes can be maintained in a system

- Requirements process needs to be more flexible, allowing what is available in the marketplace
- It is for modifying the requirements to provide a better overall business solution
- For systems built from off-the-shelf components, component selection involves a discovery process
- Understand how they can achieve the desired quality attributes
- Then decide whether they can be integrated the system being built

Architectural Mismatch

- Not all components work together- even if they are commercial products that claim compatibility
- Components that were not developed specifically for the system, may not meet all of requirements
- Architectural mismatch usually shows up at system integration time
- The system will not compile, will not link, or will not run

- What to do about interface mismatch?
- Besides changing your requirements there are three things:
    1. Avoid it by carefully specifying and inspecting the components for your systems
    2. Detect those cases you have not avoided by careful qualification of the components
    3. Repair those cases you have detected by adapting the components

Techniques For Repairing Interface Mismatch

- Alternative to change the code of one / both mismatched components is to insert code that reconciles their interaction in a way that fixes the mismatch
- There are three classes of repair code:
  - I. Wrappers
  - II. Bridges
  - III. Mediators

Wrappers:

- The term wrapper implies a form of encapsulation
- We can interpret interface translation as including:
  - Translating an element of a component interface into an alternative element
  - Hiding an element of a component interface
  - Preserving an element of a component's base interface without change

Bridges:

- A bridge translates some requires assumptions of one arbitrary component to some provides assumptions of another
- Bridge vs wrapper is that the repair code constituting a bridge is independent of any particular component

Mediators:

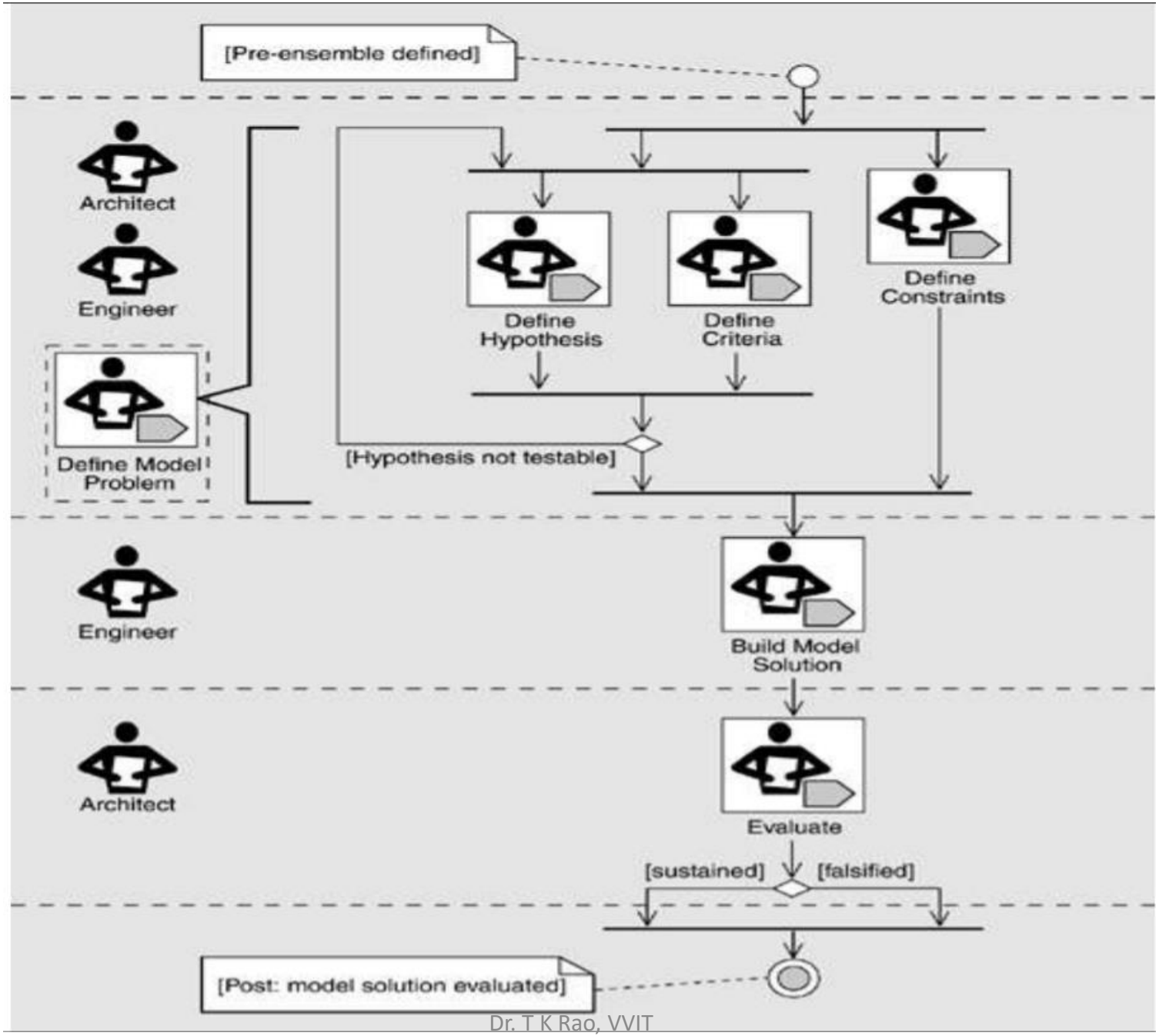
- Mediators exhibit properties of both bridge and wrappers
- The major distinction between bridges and mediators is that incorporate a planning function

Techniques For Detecting Interface Mismatch

- Discovering all of the required assumptions of the components for each of the services that will be used by the system
- Making sure that each required assumption is satisfied by some provided assumption in the system

Model problem workflow

- The process consists of the following six steps that can be executed in sequence:
  I. The architect and the engineers identify a design question
  II. The architect and the engineers define the starting evaluation criteria
  III. The architect and the engineers define the implementation constraints
  IV. The engineers produce a model solution situated in the design context
  V. The engineers identify ending evaluation criteria
  VI. The architect performs an evaluation of the model solution against the ending criteria
- An illustration of the model problem workflow is shown in Figure.


[Pre-ensemble defined]
Architect
Engineer
Define Model Problem
Define Hypothesis
Define Criteria
Define Constraints
[Hypothesis not testable]
Engineer
Build Model Solution
Architect
Evaluate
[sustained]
[falsified]
[Post: model solution evaluated]

Software architecture in future

- 1960s was the decade of subroutines
- 1970s was a concern with the structuring of programs to achieve qualities
- Data Flow analysis, ER diagrams, information hiding were the bases
- 1980s was module-based programming, information hiding, objects, inheritance
- 1990s was a standard Object based architecture, in the form of frameworks
- In current decade, middleware and IT architecture has been a standard platform

ABC revisited

- In this, four versions of ABC were identified and discussed in terms of feature research as follows:
  - The simplest case, in which a single organization creates a single architecture for single system
  - One in which a business creates not just a single system from an architecture but an entire product line of systems that are related by a common architecture and a common asset base
  - One in which, through a community-wide effort, standard architecture or reference architecture is created from which large numbers of systems flow
  - One in which the architecture becomes so pervasive that the developing organization effectively becomes the world, as in the case of the WWW