**Unit 4**. Software Architecture analysis and design: requirements for architecture and the life-cycle view of architecture design and analysis methods, architecture-based economic analysis: Cost Benefit Analysis Method (CBAM), Architecture Tradeoff Analysis Method (ATAM). Active Reviews for Intermediate Design (ARID), Attribute Driven Design method (ADD), architecture reuse, Domain –specific Software architecture.

--------------------------------------------------------------------------------------------------------------------------------

**Software Architecture analysis and design:**

A system's software architecture is widely regarded as one of the most important software artifacts. Software professionals routinely make decisions that impact that architecture, yet many times that impact is not fully considered or well understood.

* Which design decisions will lead to a software architecture that successfully addresses the desired system qualities?
* How do you know if a given software architecture is deficient or at risk relative to its target system qualities?

**Requirements for architecture and the life-cycle view of architecture design and analysis methods:**

**What is Requirements Architecture?**

Requirements architecture is the organization and structure of software requirements artifacts. When trying to describe what requirements architecture is, it’s helpful to look at other types of architecture. First, think of what a building architect does -

An application architect or software architect designs software so the application is usable, stable, and appealing to the user. In software architecture, components of the software are implemented so that changes can be made easily, integrations between other systems are smooth and only built when necessary, and so that the user experience is acceptable. **One goal of good software architecture is to handle scalability of the system and minimize complexity, and is therefore much more important as a system becomes more complex and is used by a wider variety of people.**Think about the software architecture of an ERP as opposed to the software architecture of a simple script used by one person.

**Cost Benefit Analysis Method (CBAM)**

Cost Benefit Analysis Method (CBAM) helps organizations invest their resources to maximize their gains, meet their schedules, and minimize their risks.

Creating and maintaining systems involve making multiple business-critical architecture design decisions. System architects typically focus on the technical tradeoffs of their architectural designs. However, when developing or maintaining large, complex systems, the biggest tradeoffs are related to economics.

Organizations need to know how to invest their resources to maximize their gains, meet their schedules, and minimize their risks. And that is exactly what the CBAM enables you to do.

Because the resources for building and maintaining a system are finite, you must choose among architectural options with different costs, amounts and types of resources, features, and inherent risk or uncertainty. The CBAM enables you to explore the effects of these options using economic software models that account for all of these factors.

In other word the process of identifying the costs and benefits of architectural decisions that result in system qualities. Then you can consider the information you compiled and choose from multiple proposed architecture options. For example, you could decide whether to use **redundant hardware, check pointing,** or some other method to address concerns about the **system’s reliability**. Or you could choose to invest the organization’s resources in some other **quality attribute**, such as **higher performance**.

The **CBAM** process consists of the following steps:

1. Choose scenarios and architectural strategies.

2. Assess quality attribute benefits.

3. Quantify the benefits of architectural strategies.

4. Quantify the costs and schedule implications of the architectural strategies.

5. Calculate the desirability of each option.

6. Make architectural design decisions.

**Architecture Tradeoff Analysis Method (ATAM)**

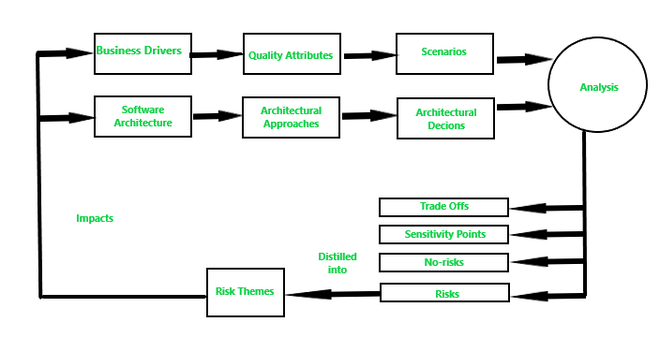
**Architecture Tradeoff Analysis :**  
This Method is a method used to evaluate the quality attributes(such as performance, availability, and security) of software architectures. ATAM is used to mitigate risks in software architectures in the early stages of the [software development life cycle (SDLC)](https://www.geeksforgeeks.org/software-development-life-cycle-sdlc/).

**Participants in ATAM :**  
The ATAM requires the participation of three groups as follows.

1. **The evaluation team –**  
   The evaluation team consists of the members who are external to the project. This team consists of 3-5 members who play their specific roles in the team.
2. **Project decision-makers –**  
   Project decision-makers have the power to speak for the development of the project and have the authority to mandate changes.
3. **Architecture stakeholders –**  
   Any individual, team, or organization who has an interest in the realization of the architecture and is somehow related to the architecture is a stakeholder of that architecture. Stakeholders include users, maintainers, performance engineers, testers, integrators, developers.

**Process of Architecture Tradeoff Analysis Method :**  
The ATAM process starts with bringing all the stakeholders to find business drivers like system goals, constraints, system functionality, and desired non-functional properties. Then from these drivers quality attributes and business scenarios are created. Then, in conjunction with architectural approaches and architectural designs, these scenarios are used to create an analysis of trade-offs, sensitivity points, and risks (or non-risks).

This analysis is then converted to the risk themes and their impacts from where the process can be repeated. With every analysis cycle, the process gradually proceeds from the more general to the more specific. The whole architecture will be fine-tuned and risk themes will be addressed by examining the questions that were discovered in the previous cycle.



ATAM  Process

**Steps of ATAM Process :**  
There are nine steps in the ATAM process as follows.

1. **Present ATAM –**  
   Present the concept of the process to all the stakeholders of the process and answer the questions asked by the participants. This step makes the participants familiar with the process.
2. **Present Business Drivers –**  
   All the participants involved in the presentation are expected to present, understand, and evaluate the business drivers for the system.
3. **Present Architecture –**  
   A brief overview of the architecture is presented by the architect with an appropriate level of detail, i.e. at least modules and C&C views are discussed.
4. **Identify Architectural Approaches –**  
   The architect presents some specific architectural approaches to the team and then the proposed architecture is discussed.
5. **Generate Quality Attribute Utility Tree –**  
   In this step, the core business and technical requirements of the system are defined and then mapped into an appropriate architectural property. Put all these parts of evaluations, designs, and requirement elicitation into a tree.
6. **Analyze architectural approaches –**  
   Each and every scenario is compared and rated by priority, and then highly rated scenario is mapped onto the architecture.
7. **Brainstorm Scenarios –**  
   The larger stakeholders group present and contribute current scenarios and their concerns.
8. **Analyze architectural approaches –**  
   Step 6 is repeated again with added knowledge from larger stakeholders in step.
9. **Present results –**  
   At the end of the evaluation, the team reviews the existing and newly discovered risks, non-risks, sensitivities, and tradeoffs. The team discusses whether any new risk themes have arisen. Then the team provides all the documentation to the stakeholders.

**Phases of ATAM :**  
The ATAM consists of four phases as follows.

1. **Phase 0 –**  
   Preparation, planning, stakeholder recruitment, and team formation takes place in this phase. Participants are evaluation team key project decision-makers.
2. **Phase 1 –**  
   This phase consists of steps 1-6 of the evaluation process. Participants are evaluation team key project decision-makers. Its typical duration is 1 day followed by a hiatus of 2 to 3 weeks.
3. **Phase 2 –**  
   This phase consists of steps 7-9 of the evaluation process. Participants are evaluation team key project decision-makers and stakeholders. The duration of this phase is 2 days.
4. **Phase 3 –**  
   This phase is a follow-up phase. Report generation and delivery of the report is done in this phase. Any scope of improvement in the process is also looked upon. Participants are the Evaluation team and evaluation clients. Duration is 1 week.

**Outputs of ATAM :**

1. A concise presentation of the architecture. The architecture is presented in one hour.
2. Articulation of business goals. This helps the new participants in understanding the business goals.
3. Prioritized quality attribute requirements expressed as quality attribute scenarios.
4. A set of risks and non-risks as follows –
   * A risk is defined as an architectural  decision that may lead to undesirable consequences in light of quality attribute requirements.
   * A non-risk is an architectural decision that may lead to undesirable consequences in light of quality attributes.
5. A set of risk themes. This set helps the evaluation team to examine all the discovered risk themes that identify systematic weakness in the architecture, process, and team.
6. Mapping of architectural decisions to quality requirements. For each quality attribute scenario examined during an ATAM, those architectural decisions that help to achieve it are determined and captured.

**Active Reviews for Intermediate Designs (ARID):**

Both Active Design Reviews and ATAM are used to evaluate preliminary designs. In the active design review, stakeholders receive detailed documentation and then complete exercise questionnaires on their own. ATAM is used to evaluate whole architecture and not a portion of it.

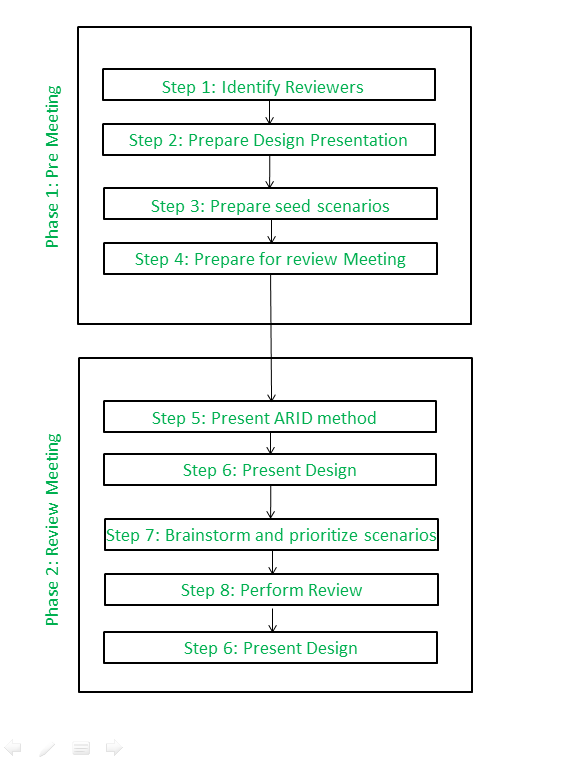
Both ATAM and ADR’s have strong qualities for evaluating software architectures and designs, but still, something was needed which could provide an early insight into design strategies. Thus, ARID was born by combining stakeholder-centric, scenario-based, architecture evaluation method like ATAM and an ARD of design specifications.

ARID is an easy, lightweight evaluation approach that is made by combining ADR’s and evaluation strategies like ATAM, which focuses on suitability and does not require complete architectural documentation.

**ARID Participants :**  
The main participants in ARID process are ARID review team (facilitator, scribe, and questioners), software architect/lead designer, and reviewers.

1. **ARID review team :** It consists of three roles :
   * **Facilitator –**  
     The facilitator works with the software architect to prepare for the review meeting and facilitates it when it takes place.
   * **Scribe –**  
     The scribe captures the issues and results in the review meeting.
   * **Questioners –**  
     One or more questioners raise issues, ask questions, and assist with creating scenarios during the review meeting.
2. **Software architect/Lead designer :**  
   The software architect(or designer) is the spokesperson for design and is responsible for preparing and presenting the design as well as participating in it.
3. **The reviewers :**  
   The reviewers are drawn from the community of stakeholders of the design, people who have an interest in its adequacy and usability, and the software engineers who are expected to use the design.

**ARID Steps :**  
An ARID process progresses across two phases that comprise of nine steps :



*ARID Steps*

**Phase-1** of ARID is carried out as a meeting between the lead designer and the review facilitator.

* + **Step-1 : Identify Reviewers –**  
    The lead designer and facilitator work together to identify the set of people who should be present at the review.
  + **Step-2 : Prepare the design presentation –**  
    Designer prepares a brief explanation of design. Goal of this step is to present design in sufficient detail so that a knowledgeable audience could use design. Here, during Phase One, the designer gives a dry run of presentation to review facilitator.
  + **Step-3: Prepare seed scenarios –**  
    Designer and review facilitator prepare a set of seed scenarios. Like seed scenarios in ATAM, these are designed to illustrate concept of a scenario to reviewers, who have opportunity to see a sample set.
  + **Step-4: Prepare for the review meeting –**  
    Copies of presentation, seed scenarios, and review agenda are produced for distribution to the reviewers during main review meeting.

During **Phase-2,** the reviewers assemble and the meeting commences.

1. **Step-5: Present ARID method –**  
   The review facilitator spends 30 minutes explaining the steps of the ARID to the participants.
2. **Step-6: Present design –**  
   The lead designer presents two – hour overview presentation and walks through examples. During this time, a ground rule is that no questions concerning implementation or rationale are allowed, nor are suggestions about alternate designs. Goal is to see if design is usable, not to find out why things are done in a certain way, or to learn about implementation secrets behind interfaces. Questions of factual clarification are allowed and encouraged. Facilitator enforces this rule during presentation.

During this time, the scribe captures each question, or each instance where designer indicated that sort of resource (usually a kind of documentation) was on its way nut not yet available. Resulting list is summarized to show potential issues that the designer should address before design could be considered complete and ready for production.

1. **Step-7: Brainstorm and prioritize scenarios –**  
   Just as in ATAM, participants suggests scenarios for using design to solve problems they expect to face. During brainstorming, all scenarios are given a fair chance. Then seed scenarios are also kept in the pool. Then, scenarios getting the most votes will be used to test design for usability.
2. **Step-8: Perform review –**  
   Considering highest voted scenario, facilitator asks reviewers to jointly craft code that uses design services to solve problem posed by scenario. After review of considered scenarios team reach up to a conclusion.
3. **Step-9: Present conclusion –**  
   At the end, list of issues is recounted, participants are polled for their opinions regarding efficacy of the review exercise.

**Active reviews of intermediate designs** (**ARID**) is an architecture review method that combines ADR with ATAM. This hybrid method takes from the ADR approach the focus of reviewing a software architecture while it is in progress and the emphasis on active reviewer participation. It combines this with the ATAM approach of focusing on quality attribute scenarios. The goal is to provide valuable feedback into the viability of the software architecture and uncover any errors and inadequacies with it.

**Attribute Driven Design method (ADD):**

Attribute-Driven Design (ADD) method is a systematic step-by-step method for designing the software architecture of a software-intensive system. It is an approach to defining software architectures by basing the design process on the architecture's quality attribute requirements. It follows a recursive decomposition process where, at each stage in the decomposition, tactics and architectural patterns are chosen to satisfy a set of quality attribute scenarios.

Required input to ADD includes known functional requirements, quality attribute requirements, and constraints. Functional requirements may be specified with a feature list or use cases. Quality attribute requirements may be specified using quality attribute scenarios, such as result from an SEI [Quality Attribute Workshop](http://csauth-sei.sei.cmu.edu/architecture/tools/establish/qaw.cfm). Constraints are design decisions that are forced by outside factors.

**Domain specific software architecture:**

* A reference architecture, which describes a general computational framework for a significant domain of applications.
* A component library, which contains reusable chunks of domain expertise.
* An application configuration method for selecting and configuring components within the architectural to meet particular application requirements.

DSSAs involve a number of domain engineering activities. The regions of the problem space (domains) are mapped into domain-specific software architectures (DSSAs) which are specialized into application-specific architectures and then implemented.

The three key factors of DSSA are:

1. **Domain**: Must have a domain to constrain the problem space and focus development
2. **Technology**: Must have a variety of technological solutions like tools, patterns, architectures & styles, legacy systems to bring to bear on a domain
3. **Business**: Business goals motivate the use of DSSE Minimizing costs: reuse assets when possible Maximize market: develop many related applications for different kinds of end users

These three factors together apply technology to domain-specific goals which is made firm by business knowledge. The overview of the DSSA process is given below:

