

# Architectural Thinking for Intelligent Systems

## Assignment 11

### Team 11

#### 1. Think about the ATAM evaluation team. Who would you like to have on the team? What insights can each team member contribute?

In a complex project, usually the core ATAM evaluation team consists of 12-16 persons, thus for our project we would have the following persons in the ATAM evaluation team:

- I. Evaluation Team Lead/ATAM Certified Expert
- II. Business Owner/Investor
- III. Engineering Director
- IV. Program Manager
- V. Project Manager
- VI. Software Architect/Team Lead
- VII. Machine/Deep Learning Specialist
- VIII. NLP Scientist
- IX. Full-Stack Developers
- X. Unit/Integration Testers

Contribution of each team member is as follows:

- **ATAM Certified Expert:** explains the overall ATAM process to everyone in the evaluation team, sets the context, analysis method and expectations for the remainder of the activities
- **Business Owner:** lists down all the basic quality attributes that he is envisioning for the system
- **Engineering Director:** supervises the selection of architectural decisions & quality attributes (at a very high level of abstraction)
- **Program Manager:** provides an overview of the existing systems & business constraints (time to market, budgeting for all the phases of project development)
- **Project Manager:** presents a system overview which covers its technical, managerial, economic & political constraints. Also covers the architectural drivers, business goals and context
- **Software Architect:** provides a high level overview of the envisioned architectural style and its associated risks to achieve the desired quality attributes set by the stakeholders (Hardware, OS, 3<sup>rd</sup> Party dependencies etc.)

- **Machine/Deep Learning Specialist:** architects the selection and development strategies of appropriate Deep Learning models and also contributes info on the risks incurred to achieve the desired attribute
- **NLP Scientist:** designs and provides detailed information about the techniques, strategies & risks involved to train a NLP system based on real-time speech data
- **Full-Stack Developers:** gives a solid overview about the implementation specific challenges/aspects of the system such as the subsystems, layers, modules that describe the system's decomposition of desired functionality, along with the objects, procedures, functions that populate these subsystems and the relationship among them (e.g. procedure calls, method invocation, callback)
- **Unit/Integration Testers:** provide an overview of the testing strategies and test coverage to capture architectural decision risks, sensitivity points, completeness of the desired quality attributes & it's associated scenarios

## 2. Review the scenarios that you created in Assignment 5 and review/refine these scenarios if necessary. Pay specific attention to stimuli and response measures.

The scenario that we had created in Assignment 5 are already **well refined**, as we had carefully made all the scenario **responses measurable** (sign of a good quality attribute) and the **stimuli conforms** to the functionalities provided by the system, thus we can easily assign weights to these individual scenarios while constructing an utility tree, later.

Scenarios that we had created (listing only for **Primary NFR**) so far are:

### a) Scenario 1:

Part of the Scenario	Description
Stimulus source	End User
Stimulus	User asks (by voice) the system about the weather forecast for the next day
Environment	Context Sensitive Intelligent Assistant
Artifact	Speech Recognition component of Intelligent System
Response	The weather forecast report summarized both by voice and graphics on GUI
Response measure	The forecast report for that particular location should be generated and summarized in less than 1000ms

### b) Scenario 2:

Part of the Scenario	Description
Stimulus source	IT Manager
Stimulus	An IT company Manager requests (either by voice or through GUI) to book a meeting room if it is vacant

Environment	Context Sensitive Intelligent Assistant
Artifact	Input Speech/Text Recognition component
Response	The meeting room is booked if it is vacant
Response measure	The room should be booked every time if it is available within 4 - 5 seconds

c) **Scenario 4:**

Part of the Scenario	Description
Stimulus source	End User
Stimulus	User asks the following to the system: "How far is München" "I want to go there"
Environment	Context Sensitive Intelligent Assistant
Artifact	Input Speech Recognition & Context Parser components
Response	Based on the context-dependent word-level attention, the system notifies all the possible modes of transportation to reach Munich
Response measure	Possible modes of transportation to reach Munich and potential roadblocks through the journey must be conveyed clearly & the accuracy of the response should be > 60% (can use MAP, Accuracy, MRR etc.)

**3. Develop a utility tree with at least 3-5 quality attributes based on the scenarios. Attach weights to the attributes. Has your understanding about the priority of the expected system qualities evolved or changed when compared to your answers given in Assignment 5?**

Yes, our priority for individual quality attribute/system qualities has evolved 2 folds after initiating ATAM. Previously, we had assumed equal priority to every attribute that was envisioned for our system. Whereas, now we prioritize more towards Performance and less towards Portability and so on.

Please find the detailed Utility Tree representation which clearly shows the weights (priorities) assigned for every attribute:

**[Please turn the page over]**



4. Identify any risks that you see in the architecture. Which architectural decision is responsible for this risk? Why can there be a potential negative impact on a quality attribute? How could a potential risk mitigation plan look like?

The following are the **major risks** and their associated architectural decisions:

a. **Risk:** Total System/Communication Failure

**Architectural Decision:** Selection of Client-Server/n-Tier/SoA architectural style

**Affected Quality Attribute:** Reliability, Availability, Performance, Accessibility

**Effect:** Negative! As we can see if there is a potential communication failure/server becomes unresponsive in Client-Server architecture (similar for n-tier & SoA too). There is an extremely high risk of an entire system failure. This has a negative consequence/impact on the above mentioned quality attributes because we have written multiple scenarios where the availability of the system, or quickness of the response is expected to be more than 95%. And when this failure occurs, this expectation set for individual quality attribute might not be achieved.

**Risk Mitigation Plan:** There are a few options we could consider to reduce the negative impact on the quality attribute:

- Change the architectural style to Edge Computing/Onion
- Utilize modern techniques such Virtualization, Containerization, Redundant Gigabit Networks, Proxy Servers, Localized Backup Compute Nodes etc. which would complement our regular architectural style
- Reduce the weight/importance of 'Availability' quality attribute
- Refine the response measures of scenarios and set a lower system uptime time of 60%

b. **Risk:** Poor Generalization & Very High Training Time for Context Sensitive queries

**Architectural Decision:** Selection of Deep Q Networks (Reinforcement Learning) instead of Deep Reinforcement Relevance Network, Seq2Seq Encoder-Decoder Networks (Un-Supervised) or any other Supervised Learning methods

**Affected Quality Attributes:** Performance, Portability, Usability, Maintainability, Efficiency

**Effect:** Negative, as we know all the above 3 mentioned methods are fundamentally different in terms of their operation, data/feedback dependency & corresponding accuracies. Selection of Deep Q Networks would drastically increase the model training/response times with good Accuracy scores.

Additionally, the 'Usability' attribute is also badly affected as we keep looking out for user feedback compared to employing context-dependent word-level attention or question-guided sentence level attention for more Accurate statement representations in Unsupervised Learning.

Even portability attribute is affected, as the model could take very long training times to achieve an accuracy of 90% whereas we had defined an expectation to deploy the system within 2 days.

**Risk Mitigation Plan:**

- Switch from **Deep Q Network** (Reinforcement) to **Deep Reinforcement Relevance Network** or **Seq2Seq Encoder-Decoder networks** (Unsupervised)
- Upgrade the Information Retrieval algorithm from Vanilla **TF\*IDF** to **BM25**

**5. Do you see a sensitivity point in the architecture? Which of your architectural decisions affects most of the quality attributes?**

Yes, there are 3 critical sensitivity points in our system architecture which may become a risk:

- Switching from Client-Server architectural style to Pipes & Filter or Edge Computing
- Using Encoder-Decoder networks instead of Deep Q Learning
- Replacing TF\*IDF with BM25 or Markov Random Walk Models

The 'Model Selection' architectural decision affects most number of quality attributes. **Quality attributes affected:** Performance, Portability, Usability, Maintainability, Modularity and Operability

**Brief:** If we use Deep Q Networks (Reinforcement), Input text from user is the state-space for the model (which is quite heavy), and the action space is every text combination available (perhaps **infinity**). Because of this, our system would take very large amount of time to generate reasonable responses to user's queries. Thus this would badly affect all the above mentioned quality attributes. This single architectural decision whole handedly causes the system to fail (w.r.t the desired quality attributes).

Hence, it is really important to switch to alternative architectural decisions like **Deep Reinforcement Relevance Networks (DRRN)** or **Seq2Seq Encoder-Decoder Networks** to achieve a right balance between all the desired system qualities.

**6. Is there a tradeoff (compromise) that you had to make? Which constraints and forces have led to this tradeoff?**

We know that, 'a tradeoff point is a property that affects more than one attribute and is a sensitivity point for more than one attribute'

We have discussed this already in the previous answer that, '**Model Selection**' is the tradeoff point that we had to make to achieve higher Accuracy (Performance attribute) at the cost of increased Training time & overall System Complexity.

We have brainstormed (because Context-aware Question Answering is too complex!!) many times & selected 'Deep Q Networks' to train our Intelligent Conversational System.

The **advantage** of this architectural decision is Great Model Accuracy [cited from Examples & Sources].

But, this negatively affects other attributes like Portability, Manageability, Usability, Testability, Availability, and Time Behavior.

As we have prioritized '**Performance - Accuracy**' over other quality attributes, we have to make this tradeoff (compromise) to achieve a sweet spot in the overall system expectation.

**Constraints:**

- Time Constraint (Training & Response times)
- Data Constraint
- Technical Constraint
- Infrastructure Constraint

**Forces:**

- The accuracy of the response to user queries should be relatively high
- Deep Learning Model should generalize better with limited data & question examples (which are also used in Encoder Networks - Unsupervised)

X-X-X