# Software Engineering

# Assignment 6

**Asad Zaman (p18-00304)**

**Sana Haider (p18-0011)**

## Architecture Trade-Off Analysis Method (ATAM)

**Phase 1: Presenting Phase:**

1. **Presenting Business Drivers:**
   The Business Drivers in this Banking System Software are: keeping information details of a customer in database, Card information validation and activation, Cash transactions like cash withdrawal and cash transfer, checking bank balance and printing bank statements.

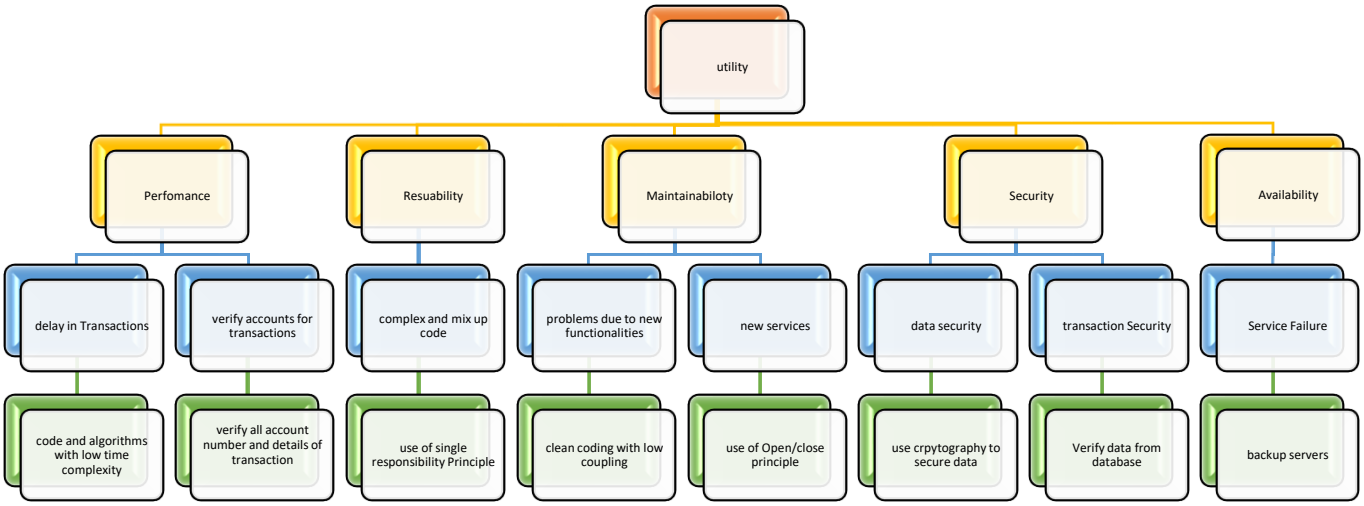
2. **Presenting Architecture:**
   The architectural style that we are going follow for this project is Service Oriented Architecture (SOA). The project will be divided into multiple services for different functional and non-functional requirements. The reason to use this architectural style is that by creating multiple services we will be able to connect the services and will be able to communicate between them properly and we'll be able to avoid the single point of failure problem. Every service will be independent from the other services so while maintaining the system in future will be a lot easier.

**Phase 2: Investigation and Analysis Phase:**

1. **Identify Architectural Approaches:**
   Doing everything from scratch will take time and a lot of hard work, so instead we will use the methods and approaches that are already present to use in order to have a good

software, Design patterns are one of the solutions to such problems, software engineers have already spent a lot of time to come up with easy and fast ways to commonly occurring problem, so all we have to do is apply them in our software/designs to make in flexible for changes in future.

2. **Generate Quality Attribute Utility Tree:**




3. **Analyze Architecture approaches:**

For good performance there should be no delay or at least minimum delay which is possible If the coding is efficient and have low time complexity, the software is said to be performing well if all the transaction are correct and it verify all the details needed for a transaction. To Reuse the services of the software in some other similar software, the coding should be in such a way that each component/class perform only one task so that it doesn't effect any other class, this is similar for maintain a software, if in future we need to add new services the coding should be done in such a way that it is open to new updates/functionalities. Security is one of the most important utility in banking

system, It is very important to keep the information of customers safe and all the details should be encrypted. There should always be a backup in some case of failure of services so the system doesn’t crash.

---