

## CHAPTER - 2

### INCEPTING ENTERPRISE APPLICATION

Determine the problem in hand. Identify need of an E.A

Scope of E.A

Analysis the feasibility

Road map for furthered phase of E.A

#### Activity

1. Inception Analysis.
2. Business modeling.
3. Requirement Elicitation and Analysis.
4. Requirement Validation.
5. Planning and Estimation.

#### (1) Enterprise Analysis

- Identify Business Opportunities.

- Identify Stakeholder across Business Unit.

- Collect and priority to the requirement.

- Define Business roadmap with scope.

- High Level Investment.

- Risk Analysis and Competitive Analysis.

## ② Business Modeling

1. Creating something new (

Development Pgm (or) Project)

2. Extension / Change to something

which already exist (Pgm / Project)

Re-engineering

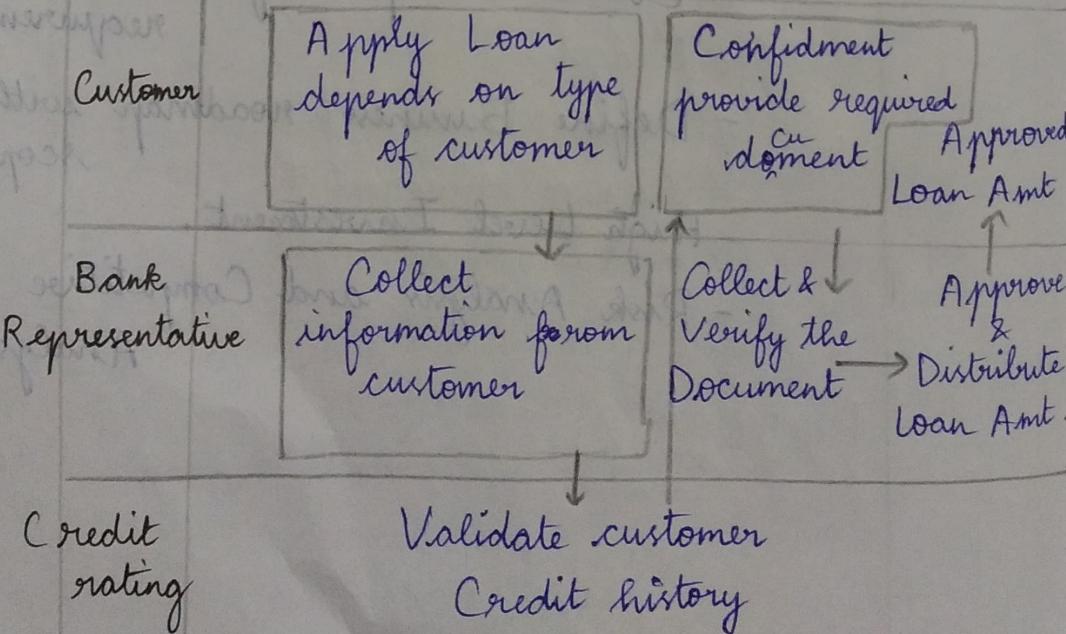
\* E-commerce application existing business model called Brick and Mortal.

### RE-ENGINEERING PGM:

- Re-writing existing business model into new E.A.
- Create entire business process and validation from existing pgm.
- May few new requirement.
- No standard definition for requirement.

### LOAN MANAGEMENT AS-IS MODEL

#### Loan Management Process AS-IS



## REQUIREMENT VALIDATION:

- Compress of three activities
- 1. Ensuring of all business needs identify.
- 2. Ensuring requirement documentation by Subject Matter Expert (SME).
- 3. Feasibility of requirement to extent possible.
  - \* ) Requirement Elicitation to be validate using Requirement Traceability Matrix (RTM).
  - \* ) User Acceptance Testing.
  - \* ) Proof Of Concept (POC).

## PLANNING AND ESTIMATION:

- Enterprise Analysis Phase leads to development of Business case.
- Business  $\xrightarrow{\text{I/P}}$  High level of planning document.
  - ↓ Called Project Charter.
  - (done by Project manager, Business Analysis Architect).
- Detailed Planning Involved.

Identifying the cost, schedule, size, and effort involved called Estimation.

- Estimation strengthen Planning.
- Estimation done by Project Manager along with Business Analyst, Technical Architect.

## Estimation Tool

- Ball Park Estimation
- Use Case Point Estimation
- Function Point Estimation

## ARCHITECTURE DESCRIPTION

- Concrete artifact to depict the architecture.

- It organised in the form of Views and Viewpoint.

VIEWPOINT: (where you are looking from)

- General represent stakeholder view of Enterprise Application.

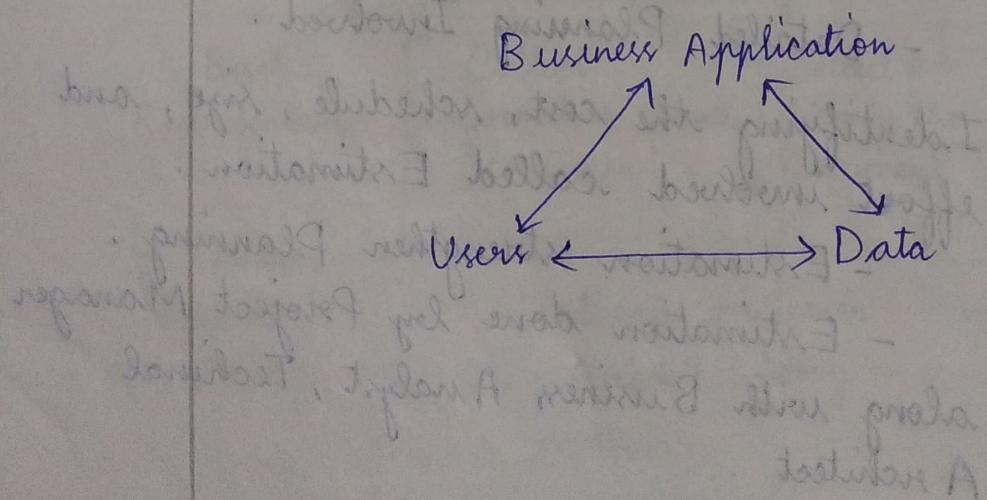
VIEW: (what you get you see)

- How the enterprise application look like from selected view point.

View and Viewpoint together constitute comprehensive architecture.

## ENTERPRISE TRIANGLE

↳ Co-existence of user business application and data and their interaction.



## ENTERPRISE ARCHITECTURE:

- Explain building blocks of organisation.
- Interaction b/w building blocks.

E.A helps

- represent
- understand
- analysis business scenario

## ENTERPRISE ARCHITECTURE FRAMEWORK:

- Specifies framework.
- Comprises of principles, standard, guideline, approach, services, design.
- TOGAF (The Open Group Architecture Framework)
  - 1. Business Architecture.
  - 2. Application Architecture.
  - 3. Data Architecture.
  - 4. Technology Architecture.

## BUSINESS ARCHITECTURE

- Overall strategy and governance of Business process and other key element.
- Business Goal.
- Business Location.
- Business Organisation.
- Business Event.

## 2. APPLICATION ARCHITECTURE :

- Structure of interconnection and interaction of all application and maps underlying process.

## 3. DATA (OR) INFORMATION ARCHITECTURE :

- Data model .
- Data life cycle management .
- Data security .
- Data integrity .

## 4. TECHNOLOGY ARCHITECTURE :

- Ensure consistent , user experience , scalability , availability .
- These are determined by
  - IT infrastructure .
  - N/W
  - Inter N/W
  - Communication protocol .

## 5. LOGICAL ARCHITECTURE :

- \* ) Represent business process in the form of logical building blocks .
- \* ) Maps building blocks into S/W .
- \* ) Bridge between requirement capture by Business analysts and technical solution will be implemented by development team .
- \* ) Independent of any technology .
- \* ) Doesn't required any tools and framework to realize logical Architect .

\* ) Organised in the form of layered fashion.

\* ) Each layer host related set of logical component

\* ) Modular approach  
desirable properties

Loose Coupling.  
Scalability.  
Reusability.  
Modularity.

Presentation Layer

Service Layer

Business Layer

Data Access Layer

Integration Layer

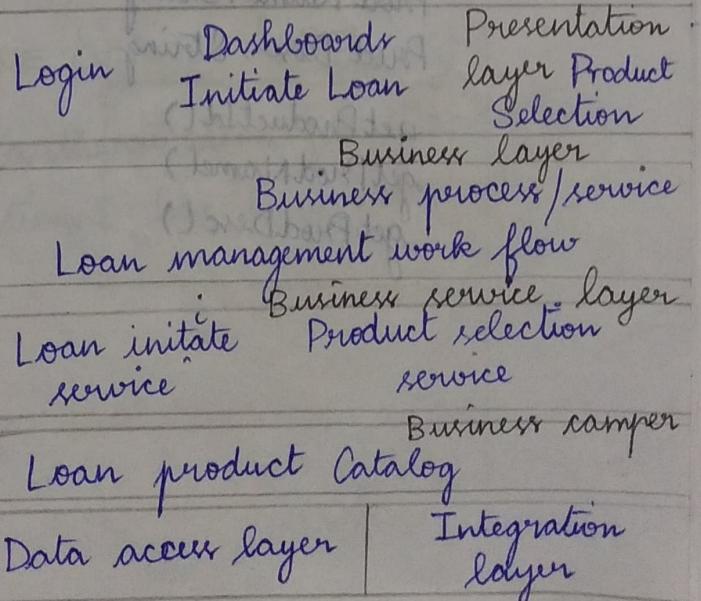
Enterprise Information

Layer

Infrastructure Service  
Layer

LOMS Logical Architecture

Logging  
Exception  
handling  
Auditing  
Caching  
Session  
Management  
Infrastructure Layer



# Enterprise information system layer

Infrastructure service layer

## TECHNICAL ARCHITECTURE:

- \* ) Logical Architecture + Usecase, Functional requirement + Non-functional requirement
- \* ) Translate logical architect into technical solution.
- \* ) Implemented Technical solution using OOP (Object Oriented Programming)
- \* ) Entity Class (Encapsulated Core Business Entity)
- \* ) Boundary Class (The class encapsulated all aspect of actors)
- \* ) Data Class (How E.A store and retrieve data from data store).

ENTITY CLASS DATA STORE

Loan Product

Prod Id : Int  
Prod Name : String  
Prod Desc : String

getProductId()  
getProdName()  
getProdDesc()

Initiateloan DAO

Serial Version : long

SaveInitialLoan()  
RetrieveLoanInfo()

Relationship among class (Relationship exist among object of class where they are aware of each other to fulfill Loan Product the functionality)

Customer

1...n

1...n

2. Aggregation:

(No. of object of the class participate in given association)

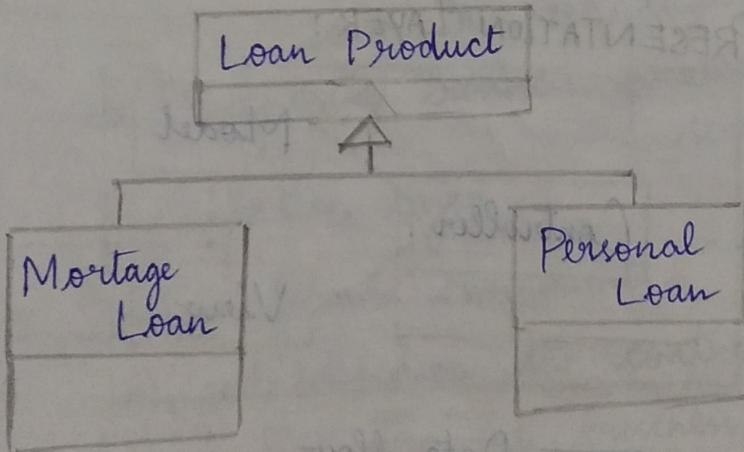
Loan detail

1...n

Customer

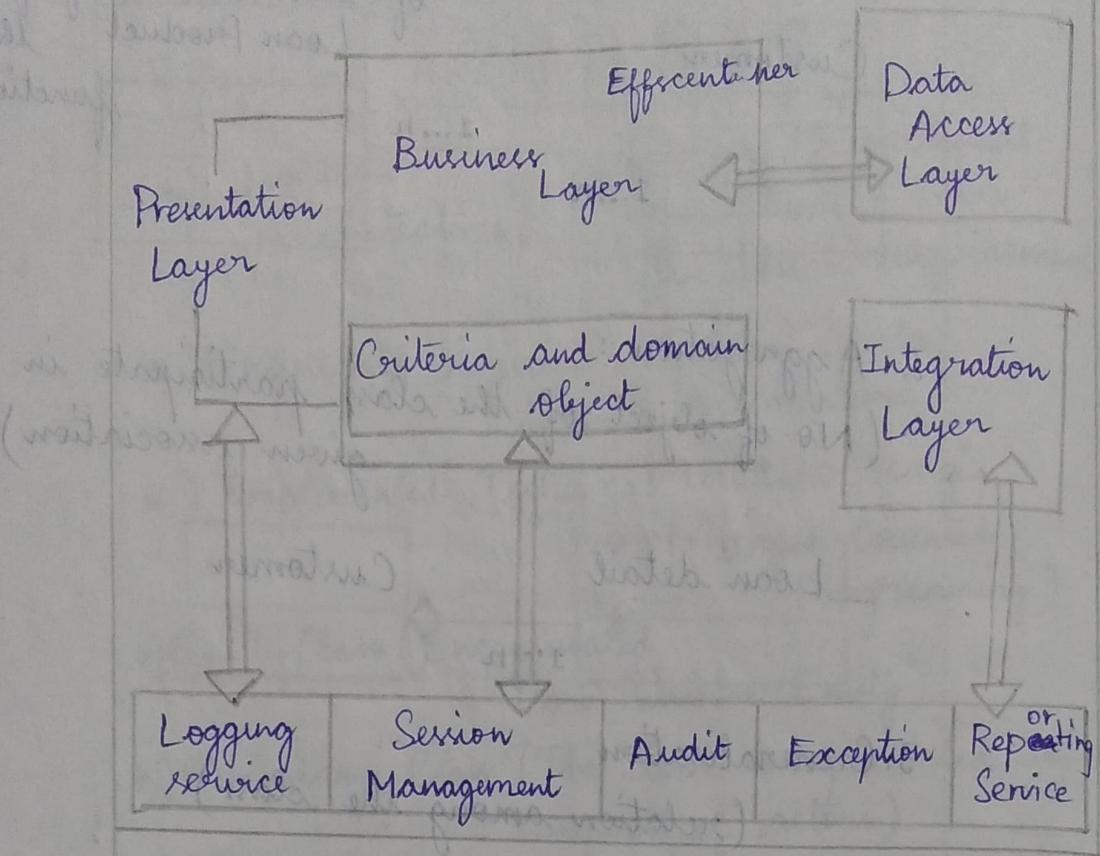
3. Generalisation

(relation among the class)

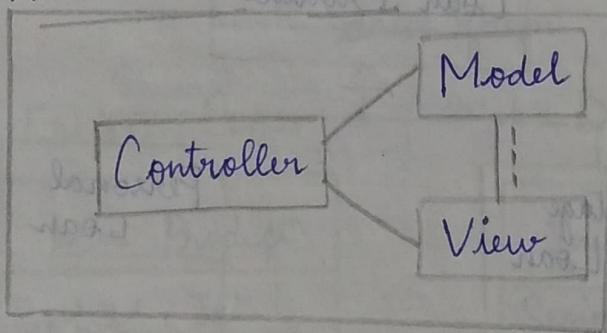


INFRASTRUCTURE SERVICE LAYER:

- Services authentication, authorisation, logging, exceptional handling, session management.



### PRESENTATION LAYER :



— Data flow

-- Control flow

### DESIGN ASPECT OF PRESENTATION LAYER :

1. Scalability  
Usability
  2. Input validation
- Logical placement of UI component
- Proper usage of screen

### 3. Internationalisation

4. Data transformation

### 5. Navigation

6. Session management

and cache management

### 7. Deployment

Web server - Static page

Application server - Dynamic page

### 8. Reusability

### 9. Security

### 10. Portal Integration

## BUSINESS LAYER:

\*) Mirror the business process of organisation

1. Business delegate component.

2. \*) Act as bridge btw presentation and business layer.

\*) Exposing business process and service to presentation layer.

2. Session facade component.

\*) Centralisation the control of underlying fine grained business entities.

\*) Co-ordinate them to realise business functionality.

3. Business Service component

(Implement Business functionality on an application).

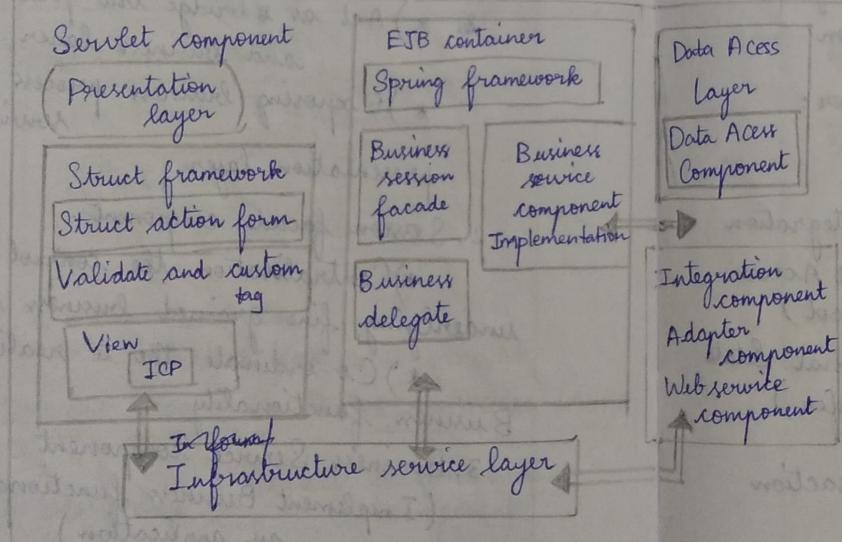
4. Business Model component

(Encapsulate real data and method to manage data).

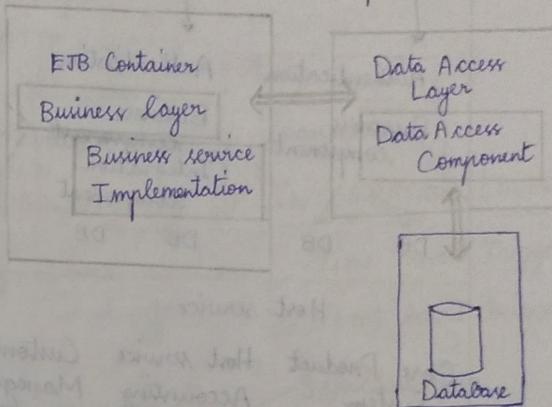
## DESIGN ASPECT OF BUSINESS LAYER:

1. Transaction management
  - ACID properties
2. Remote (or) Local access of Business logic
  - Distributed
  - Client / Server
  - Single
3. Use of third party component
4. External functionality Business function integration.

## ARCHITECTURE DIAGRAM OF BUSINESS LAYER:



DATA ACCESS LAYER: (Physically store enterprise data)



DATA ACCESS LAYER: (responsible for accessing element underlying data layer (relational database))

- DAO (Data Access Object) - convert underlying data format into object format.

- Design Aspect Data Access Layer

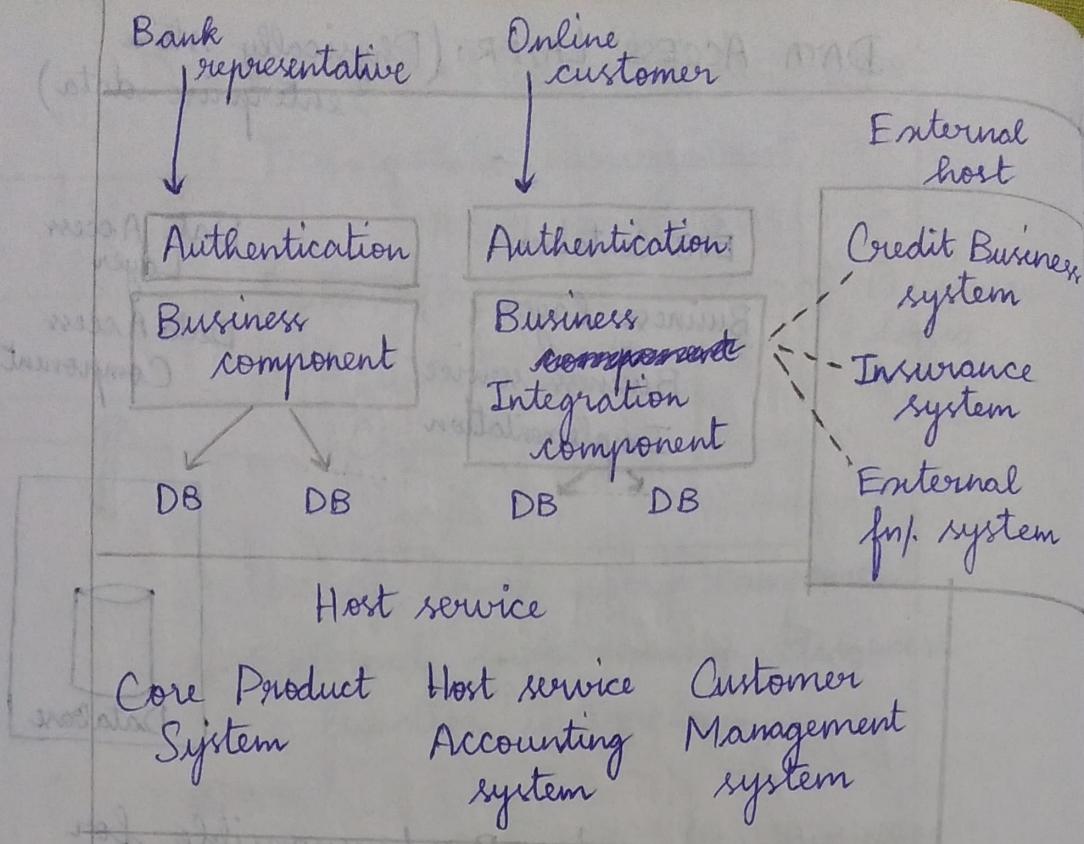
- Performance consideration.

- Security integration.

- Object relational mapping.

- Handling XML data.

INTEGRATION LAYER: (capability to connect external system through their exposed interface)



Integration types

- UI integration
- Functional integration
- Data integration

### Evaluation of Integration Layer:

1. Third-party EAI solution

2. Java Based Application Integration

3. Web service

SOAP (Simple Object Access Protocol)

REST (Representational State Transfer)

#### SOAP:

- Complex stateful and transaction oriented protocol.

- HTTP, SMTP, SNMP

- Message format

- Security, transaction, messaging.

## REST :

- Simple
- HTTP
- Light weight.
- Security, messaging transaction.

## DESIGN ASPECT OF INTEGRATION LAYER :

1. Communication
  - Time based trigger
  - Event based trigger
2. Communication
  - Synchronous method
  - Asynchronous
3. Interface contract - Provide stable view of service.
4. Error Handling mechanism - Robust error handling mechanism
5. Integration Topologies
  - Point to Point
  - Message Broker
  - Message bus

(most frequently used)

Integrating at various levels of  
- lower level  
- higher level