

Mission To Provide People With Opportunities to Build a Better Future


Welcome to the
World of
Continuous
Improvement



We helped 6000+ Professionals to live in their BA Career

What is Business Analysis

Business analysis is the practice of **enabling change** in an enterprise by **defining needs** and **recommending solutions** that deliver **value to stakeholders**.



Business analysis enables an enterprise to **articulate needs** and the **rationale for change**, and to **design and describe solutions** that can **deliver value**.

What is Business Analysis

Business analysis is performed on a variety of **initiatives** within an enterprise.

Initiatives may be **strategic, tactical, or operational.**

Business analysis may be performed within the **boundaries of a project or throughout enterprise evolution and continuous improvement.**

It can be used to understand the **current state**, to define the **future state**, and to **determine the activities** required to move from the current to the future state.

Who is a Business Analyst

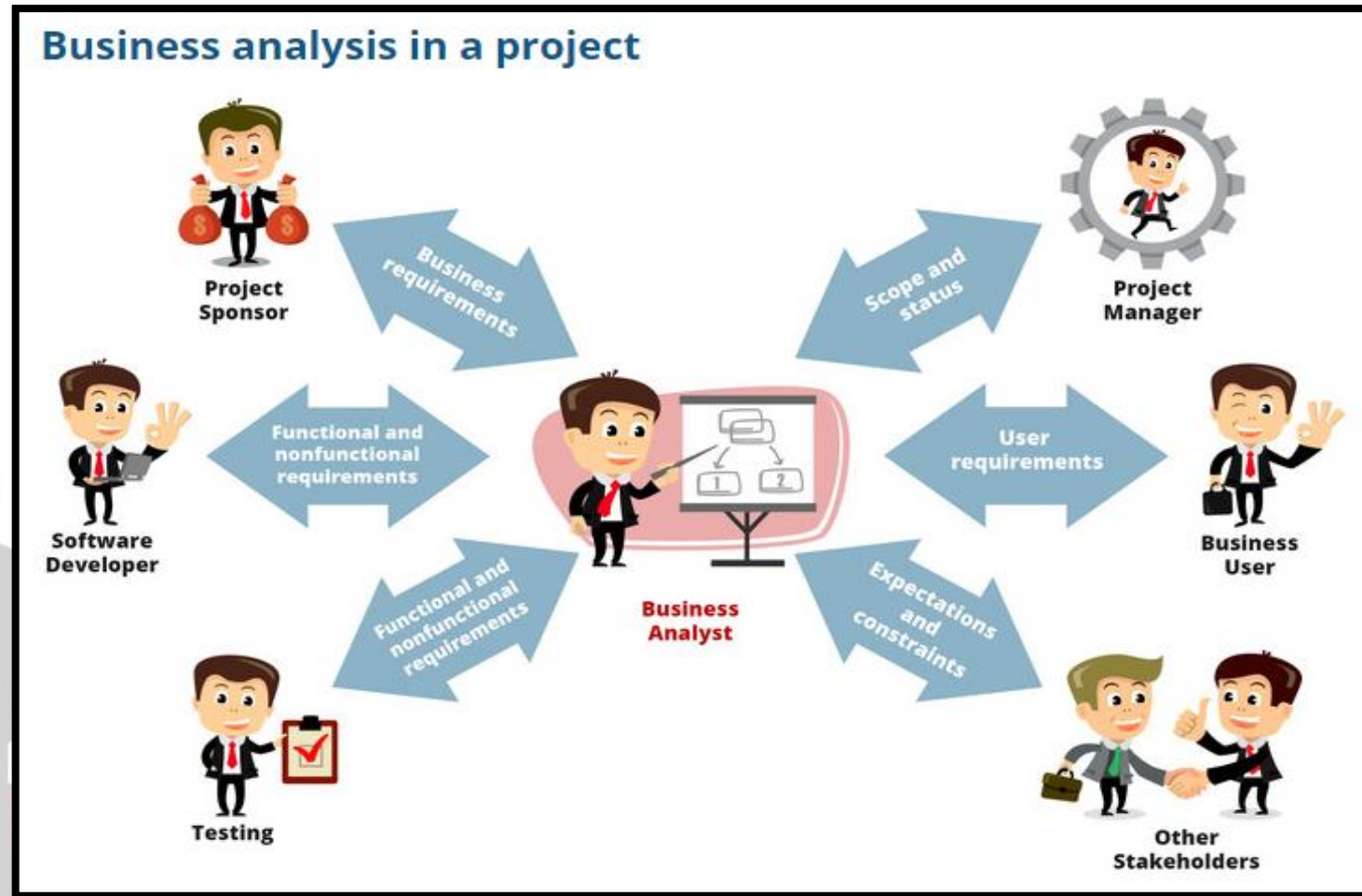
A business analyst is any person who performs business analysis tasks, *no matter their job title or organizational role.*

Business analysts are responsible for discovering, synthesizing, and analyzing information from a variety of sources within an enterprise, including tools, processes, documentation, and stakeholders.

The business analyst is responsible for eliciting the actual needs of stakeholders, which frequently involves investigating and clarifying their expressed desires in order to determine underlying issues and causes.

Business Analyst – other Job Titles

- Business Architect,
- Business Systems Analyst,
- Data Analyst,
- Enterprise Analyst,
- Management Consultant,
- Process Analyst,
- Product Manager,
- Product Owner,
- Requirements Engineer, and
- Systems Analyst.



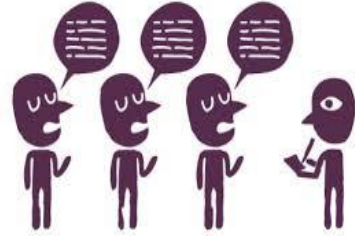
Responsibilities of a Business Analyst

Client Interaction

Ownership of Requirement

Process Re-engineering

Role of a Business Analyst in Project



Gathering
Requirement



Plan & Document
Requirement



Analyze & Model
Requirement



Communicate
Requirement



Handling Change
Request



User Acceptance
Testing

Role of a Business Analyst in Project

1

• REQUIREMENT GATHERING

2

• DOCUMENTING REQUIREMENT

3

• MODELLING REQUIREMENT

4

• COMMUNICATING REQUIREMENT

5

• TRACK REQUIREMENT

6

• HANDLE CHANGE REQUEST

7

• USER ACCEPTANCE TESTING

Requirement and Types

A requirement is basically the need of the client. This need of requirement will transform into solution while taking various shapes and forms as it progresses from each stage of SDLC.

Types of Requirement

- ❖ Business Requirement
- ❖ Stakeholder Requirement
- ❖ Solution Requirement
 - Functional Requirement
 - Non-Functional Requirement
- ❖ Transition Requirement

Requirement and Types

❖ Business Requirement

Business requirements are the high level statements of the goals, objectives, or need of the enterprise. Business Requirement describes needs of the organization as a whole and not groups or stakeholders within it.

❖ Stakeholder Requirement

Stakeholder requirements are statements of the needs of a particular stakeholder or class of stakeholders.



Requirement and Types

❖ **Solution Requirement**

Solution requirement describes the characteristics of a solution that meet business requirement and stakeholder requirement.

a. **Functional Requirement**

Functional requirement describe capabilities the system will be able to perform in terms of behavior or operations.

b. **Non- Functional Requirement**

Non-functional requirement don't directly relate to the behavior or functionality of the solution , but rather describes environmental conditions under which the solution must remain effective like capacity, speed, security etc.

❖ **Transition Requirement**

It describes the capabilities that the solution must have in order to facilitate transition from the current state of the enterprise to a desired future state.

Do's Don'ts and Challenges

- Never say NO to client
- Never imagine anything in terms of GUI
- There is NO word called as “By Default”
- Consult an SME for clarifications in Requirements

Challenges

- Obtaining sign-off on requirement
- Change Management- with respect to cost and timelines
- Coordination between developers & testers
- Conducting meetings
- Driving client for UAT completion
- People Management (coordinating with different people and different teams)

Stakeholders and Types

A stakeholder is any person or a group of persons or an organization that is directly or indirectly effected or impacted by the proposed IT solution.

Types of Stakeholders

1. Project Stakeholders
2. Business Stakeholders
3. 3rd Party Stakeholders
4. Negative Stakeholders



Stakeholders and Types

Project Stakeholders

- ❖ Business Analyst
- ❖ Project Manager
- ❖ Development Team
- ❖ Testing
- ❖ Operations

Business Stakeholders

- Project Manager
- Business Owner
- Business Sponsor
- Executive Sponsor
- Operation Team
- SPOC
- End user

3rd Party Stakeholders

- ❖ Auditors
- ❖ Focus Group
- ❖ Manufacturer
- ❖ Outsource
- ❖ Legal Specialist

Negative Stakeholders

- ❖ Competitor
- ❖ Hacker
- ❖ Development Team
- ❖ Political Party
- ❖ Public Opinion



Business Process Model

Has a GOALS
Has specific INPUTS
Use RESOURCES
Perform ACTIVITIES in some order
Has specific OUTPUT
Creates END VALUE to Customers



8 Skill Areas of a Business Analyst



IT Companies Overview

Types of IT Companies

Product Development IT Companies	Application Development IT Companies
Microsoft , Oracle, IBM, SAP, CRM	Infosys, TCS, Wipro, Satyam
IT Company will have the concept and they invest time and money to build the product	Client will have a requirement and will engage an IT Company to develop an IT application
IT Company initiates the development	Client initiates the development
IT Company is the owner of the developed Product	Client is the owner of the developed application
IT Company will sell the same product to multiple Clients or Customers	Client will be the only customer for this application
Customizations are done if required for each customer	
Installation will be done at the Clients place	Deployment will be done at the Clients place
Business Analyst Role	
Understand the product features	Understand the Client Industry
Understand the product Domain	Understand the Client requirements domain
Understand where this product fits in the Domain	
Understand the Client's requirement	
Understand what customizations are required for this product to fit in the Client's requirements	

How Projects get initiated

- RFP (Request for Proposal)
- RFI (Request for Information)
- Pre-Bidding Conference- PPT presentation and Q&A session
- RFQ (Request for Quotation)
- Verification: Technical & Financial
- Shortlist any 5 IT Companies
- Negotiation
- SOW is released by the client to IT Company



Basic Knowledge on Projects

BA Proportion in projects

- The total project time should be allotted to BA is 2 months time for 1 year project.
- 12% to 16% of team size should be BA's. (2 BA's in 12, 13 members team whereas 4 BA's in 24,25 members team

Project Sizes

- Small Projects: Up to 500 Man-Hours
- Medium Projects: Up to 500-1000 Man-Hours
- Large Projects: Up to Above 1000 Man-Hours



Basic Knowledge on Projects

Project Types

Billing Projects

Fixed Bid projects

IT Projects – Manhours

Milestones in Projects – Track using Weeks



Reasons for Project Failures

- Improper requirement gathering
- Continuous change in requirements
- Lack of user involvement
- Lack of executive support
- Unrealistic expectation
- Improper planning

Basic Knowledge on Projects

Working in a Project

- Important mails are followed up by a phone call
- All mails should be Acknowledged.
- All project mails should be copied to PM.
- Decision will be taken by PM & BA should follow them.

Quality Audits

- During the project progress, there will be external as well as internal Audits conducted.



Scope Creep

If any project cannot be completed within budget and time constraints.

Basic Knowledge on Projects

Gantt Chart

- A project manager generally plan their project by using Gantt Chart.
- Gantt Chart displays information visually as a type of bar chart in a clear to understand way.

Timesheets

Resources will have to fill in their timesheets everyday for 8 hrs work activity. These timesheets will be sent to client by the PM.



Server Information

Development Server- IT Company

UAT Server- Client

Production Server- Client

Go- Live

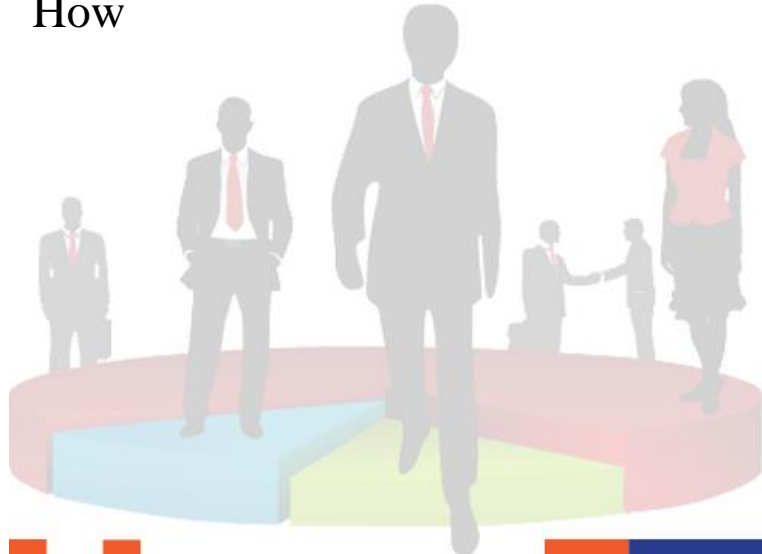
Basic Knowledge on Projects

Document Naming Standards

All documents will be named using some standards
like [ProjectID][Document Type]V[X]D[Y].ext

5W 1H- Tool of a BA

The tool is used for extracting requirement form client like Why, What, Where, Who, When & How



IT-Company- Standards

Some of the standards that IT companies adopt are:

- CMMI (Capability Maturity Model Integration)
- IEEE (Institute of Electrical & Electronics Engineering)
- ISO

Risk Analysis n Management

An uncertain event or condition which can have impact on either cost, time and scope. Risk Analysis is the process to identify the business, financial, technological & operational risk.

A *risk* is something that could affect the success or failure of a project. Analyze risks regularly as the project progresses. While you may not be able to avoid every risk, you can limit each risk's impact on the project by preparing for it beforehand.



Strategies include the following:

Avoid: Do something to eliminate the risk.

Mitigate: Do something to reduce damage if risk materializes.

Transfer: Pass the risk up or out to another entity.

Accept: Do nothing about the risk. Accept the consequences.

BA Competencies

Business Communication

3W Rule of Communication

- Who
- What
- When

Types of Business Communications

- Verbal
- Non-Verbal



BA Competencies

Supplier Management

Contract Types

Time & Material

Fixed Price Delivery

Risk & Reward



Leadership Qualities

- Team Player
- Mentor
- Motivator
- Coach

BA Competencies

Conflict Management - Thomas Kilmann Technique

X Axis- Co-operation, Y Axis- Assertiveness

5 Options of Conflict Management

- Competing
- Avoiding
- Accommodating
- Collaborating
- Compromising

5 Steps to Conflict Management

Identify Conflict

Discuss the details

Agree with root problem

Check for the every possible solution for the conflicts.

Negotiate the solution to avoid the future conflicts.



SDLC Methodologies

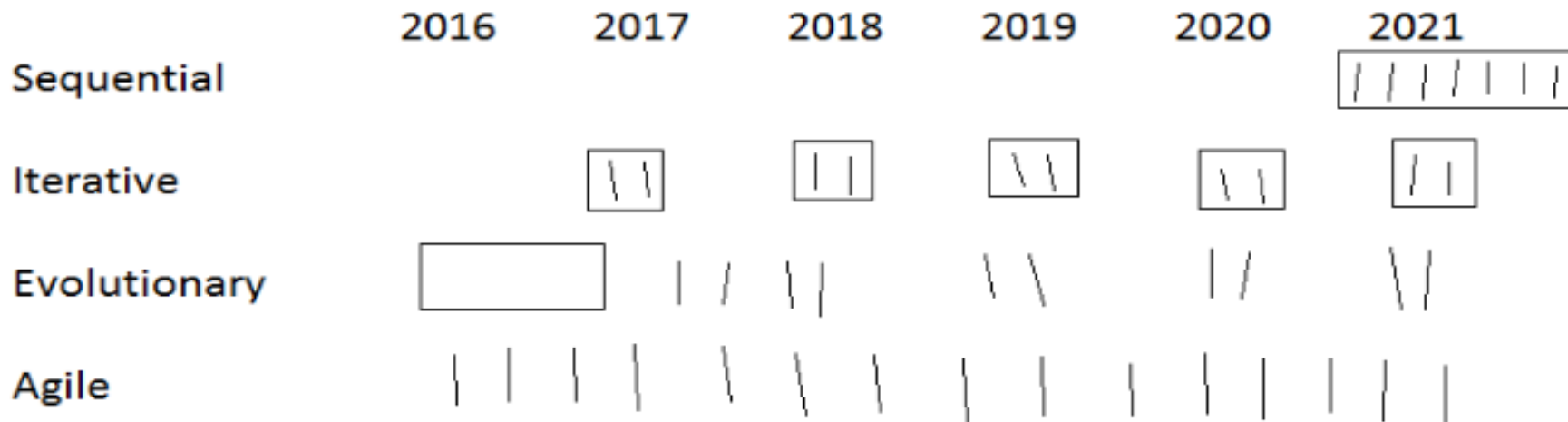
Process which an IT Company follows to develop the software application - SDLC

Methodologies - Models

Methodologies - set of guidelines - Constitution

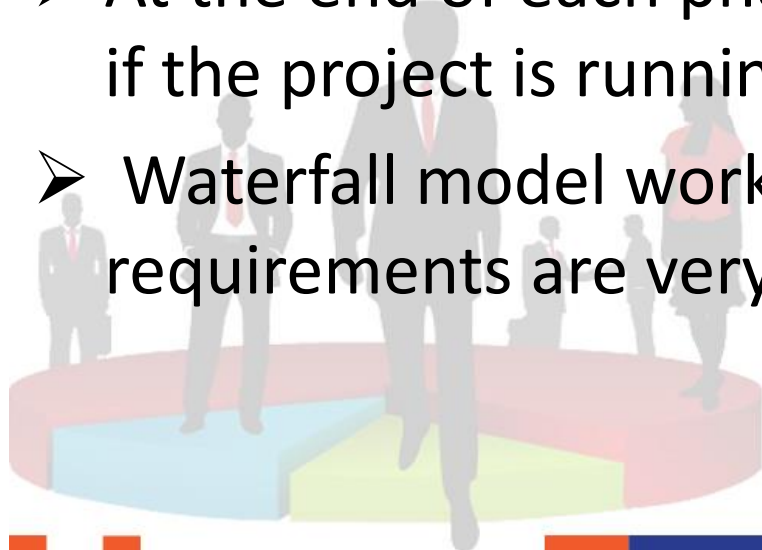
Models - follow the guidelines and achieve the development - Government

Methodologies - 5 yrs Project



Waterfall Model

- Waterfall model is a traditional model. Waterfall model follows a structured approach with each phase having specific deliverables.
- At the end of each phase a review takes place to determine if the project is running fine.
- Waterfall model works well for smaller projects where requirements are very well understood.

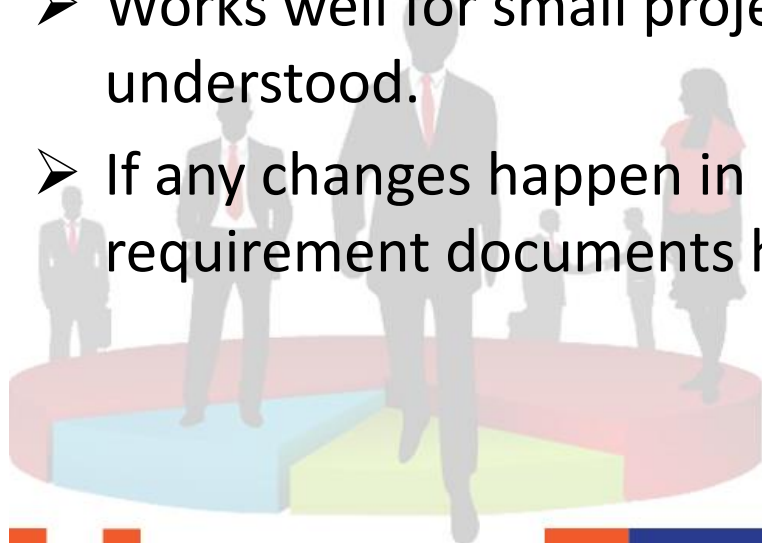


Stages – Waterfall Model

Stages of Waterfall Model	Resources	Artifacts
Requirements gathering	BA, PM	BRD
Requirements Analysis	BA, PM Tech Team – Sol Arch, NW Arch, DB Arch	FS/ FRS, SSD, SRS RTM
Design	Tech Team – Sol Arch, NW Arch, DB Arch, GUI Designer	HDD / ADD Solution Document
Development – coding	Programmers Developers	LDD /CDD Application
Testing	Testers	Test Documents Application with less Errors
Unit, Component, System, System Integration, UAT		
PROCESS - Configuration Management - PM		
Deployment & Implementation	Release Engineers	

V Model

- V- model means Verification and Validation model.
- Each phase must be completed before the next phase begins. Testing of the product is planned in parallel with a corresponding phase of development in V-model.
- Proactive defect tracking – that is defects are found at early stage.
- Works well for small projects where requirements are easily understood.
- If any changes happen in midway, then the test documents along with requirement documents has to be updated.



RUP – Rational Unified Process

- RUP stands for Rational Unified Process, Where phase/module wise (long term project) application is developed. Hence we can track the defects at early stages. This avoids the downward flow of the defects.
- Change request is welcomed in every phase of development.
- This model is called heavy weight process model.
- This model has multiple stages which requires more resources and more budget.

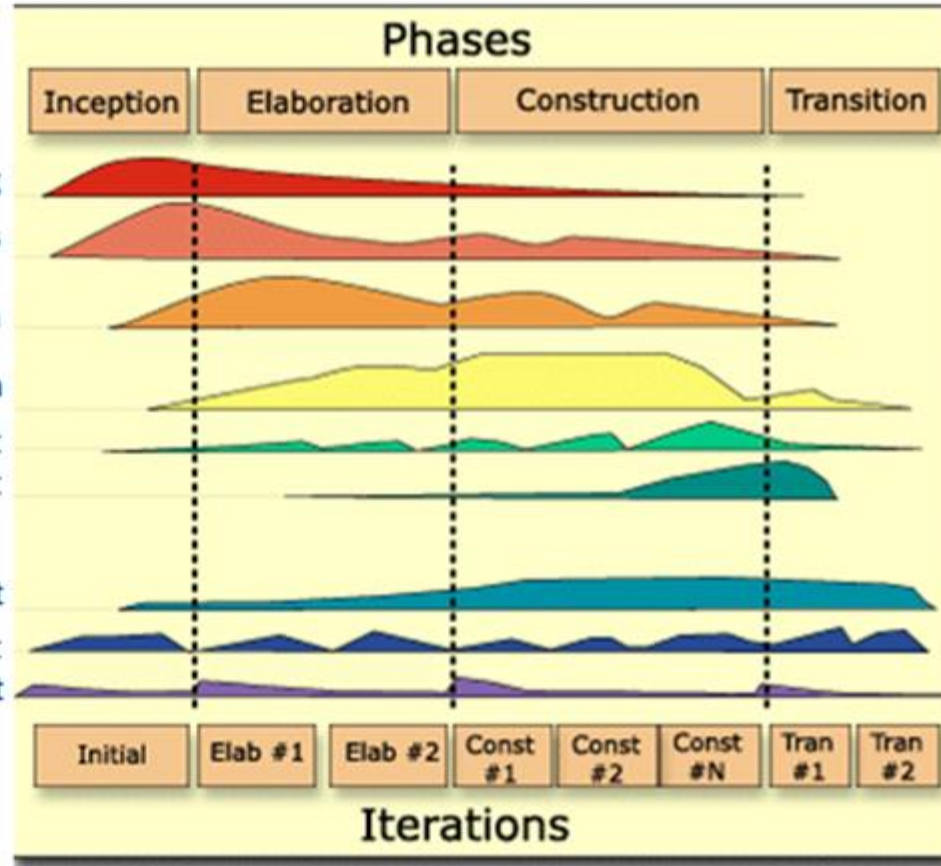
RUP – Rational Unified Process

Core Process Work Flow

Business Modeling
 Requirements
 Analysis & Design
 Implementation
 Test
 Deployment

Core Supporting Work Flow

Configuration & Change Management
 Project Management
 Environment

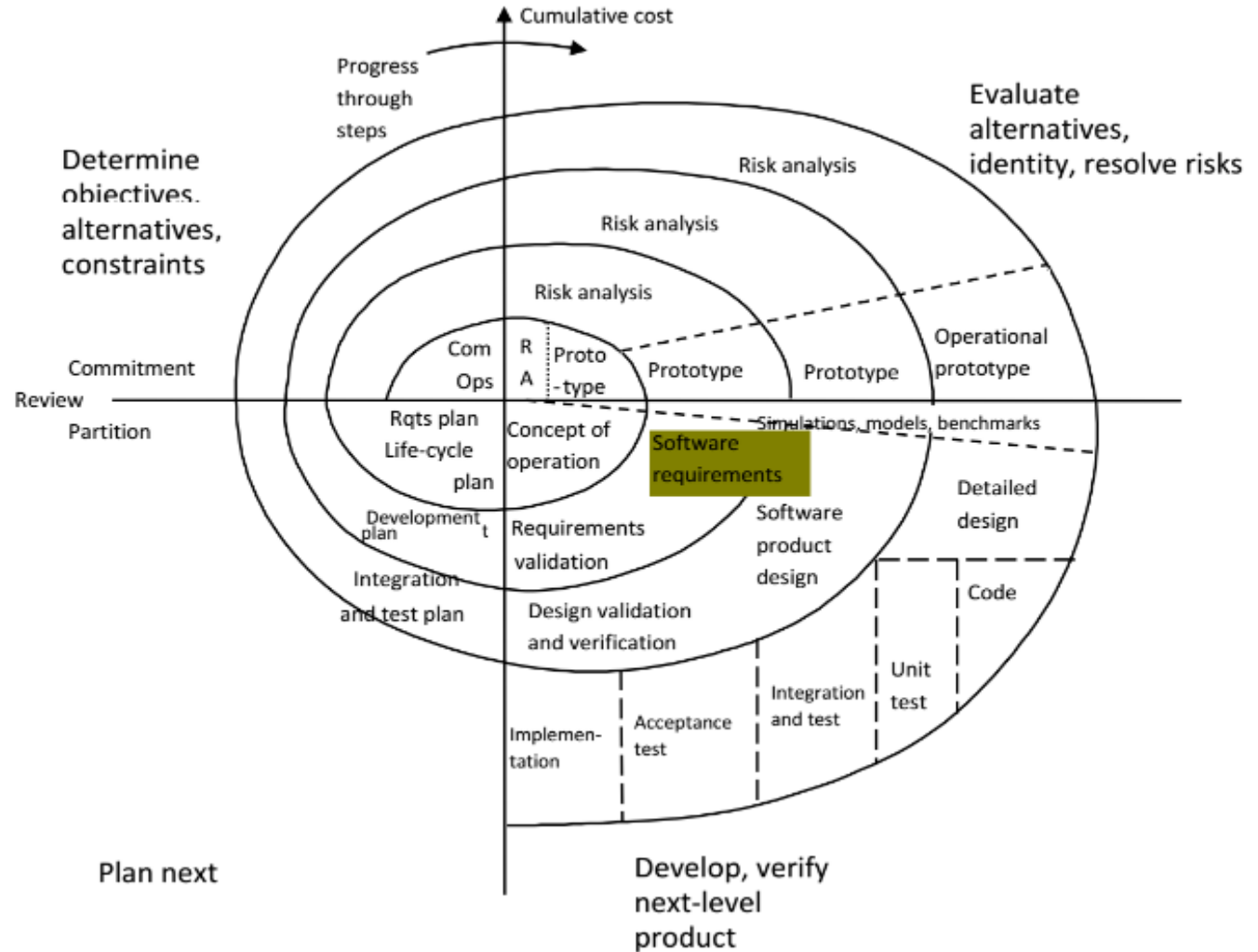


Spiral Model

- The spiral model is a risk-driven process model generator for software projects.
- The spiral model has four phases: Planning, Risk Analysis, Engineering and Evaluation.
- A software project repeatedly passes through these phases in iterations (called Spirals in this model).
- The baseline spiral, starting in the planning phase, requirements are gathered and risk is assessed. Each subsequent spirals builds on the baseline spiral.

Spiral Model

1.3. Evolutionary – Spiral



Agile Manifesto

Four main Values

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

Twelve Principles of Agile Software

1. Satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

Agile Manifesto

Twelve Principles of Agile Software

6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

7. Working software is the primary measure of progress.

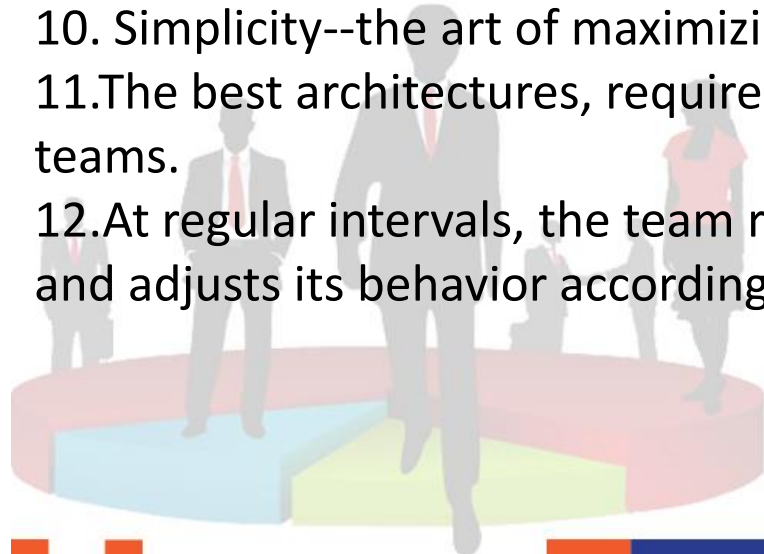
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

9. Continuous attention to technical excellence and good design enhances agility.

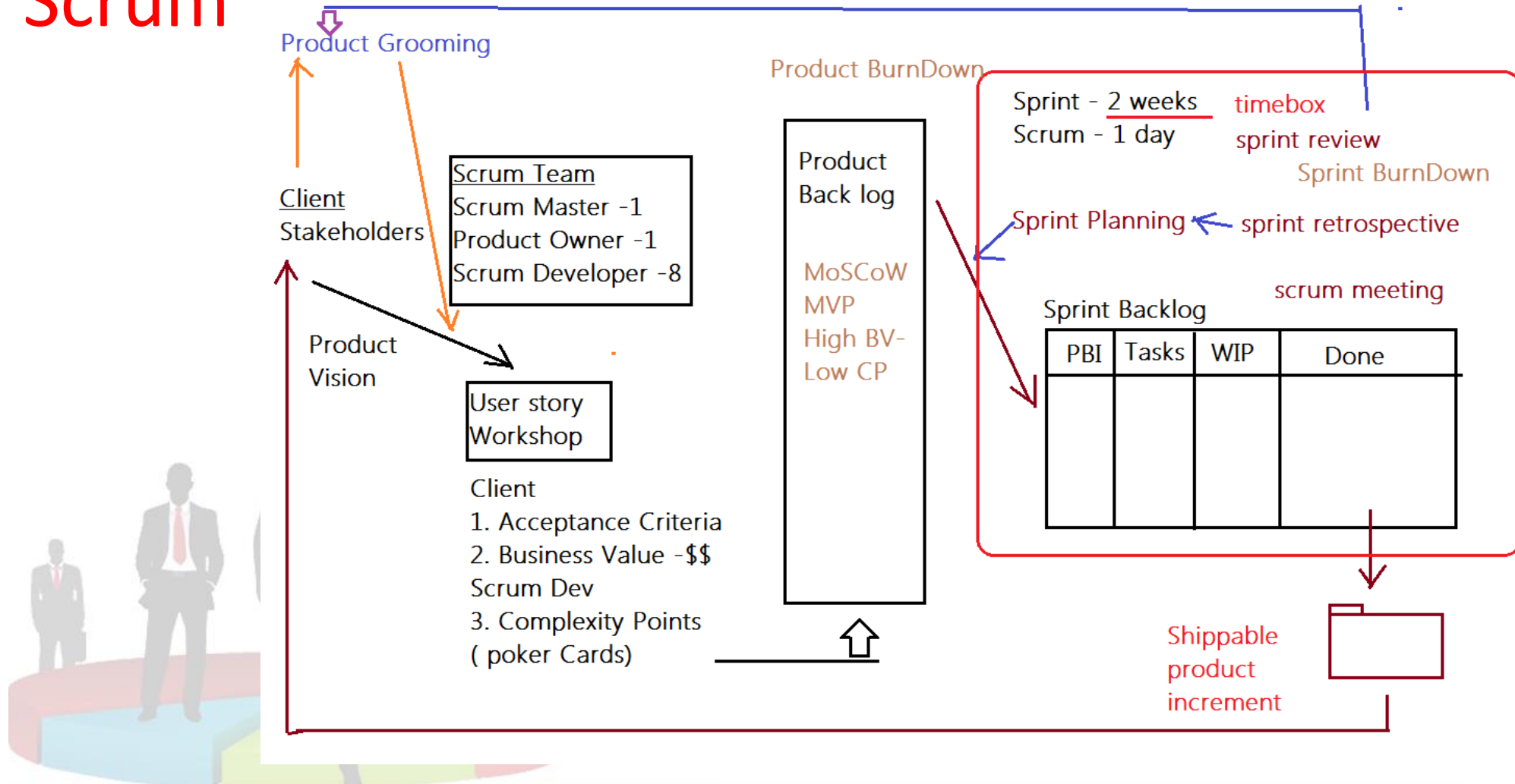
10. Simplicity--the art of maximizing the amount of work not done--is essential.

11. The best architectures, requirements, and designs emerge from self-organizing teams.

12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.



Scrum



Scrum

User Stories will have below Information:

As a

<User>

I Want to

<What is the Purpose>

So that

<What you gain by this story>

eg:

As a

Customer

I want to

Login to the net banking site

So that

I can perform banking activities



- ✓ Product/ Sprint Burndown
- ✓ Product Backlog
- ✓ Sprint Backlog
- ✓ DOR- Definition of Ready
- ✓ DOD- Definition of Done
- ✓ Task
- ✓ WIP – Work in Progress
- ✓ Sprint Meetings
 1. Sprint Planning Meeting
 2. Daily Scrum Meeting
 3. Sprint Review Meeting
 4. Sprint Retrospective Meeting

OOA- Object Oriented Approach

Object :

An object is an instance
of a class



Identity – Regd Number

State – Attributes - Attribute values

Color - Red

Passenger Capacity – 4

Fuel – Diesel

Wheels – 4

Behavior– Actions – Methods – Camel casing

turnRightAndThenSecondLeft();

slowDownAndStop();

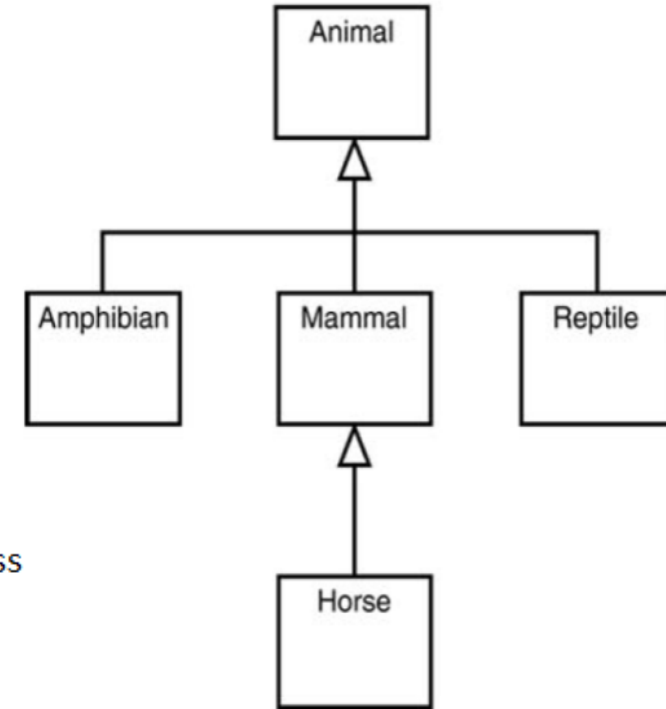
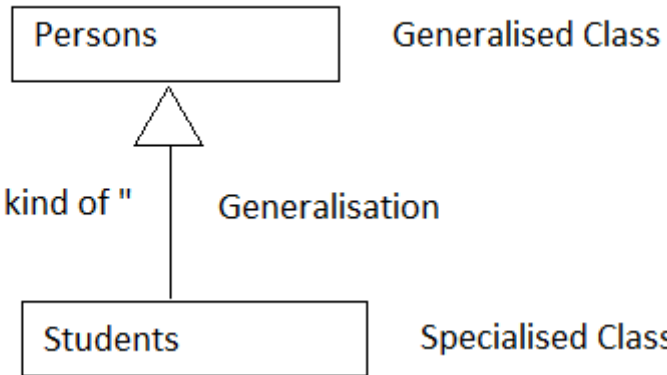
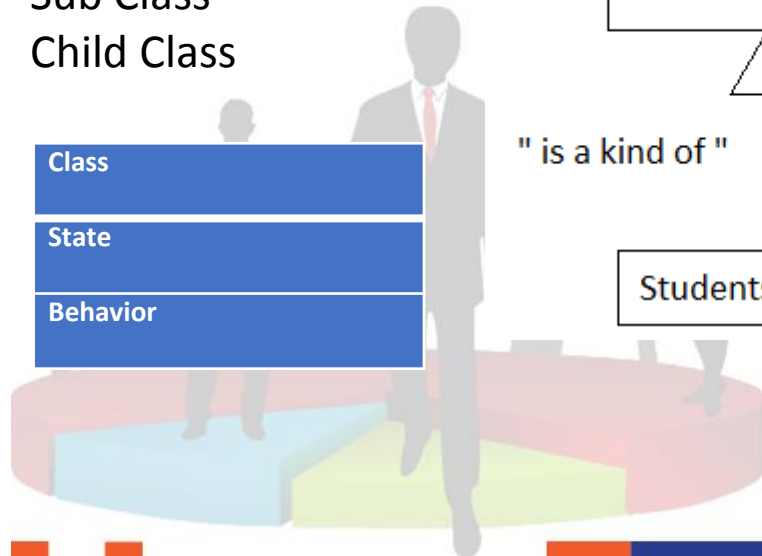
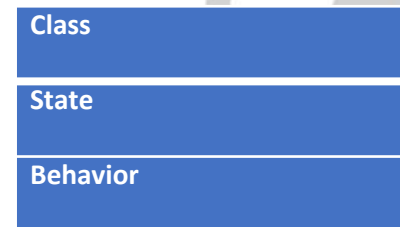
accelerateAndIncreaseSpeed();

Class

Collection of similar Objects is a Class
Object can be an instance of the Class.

Types of Classes

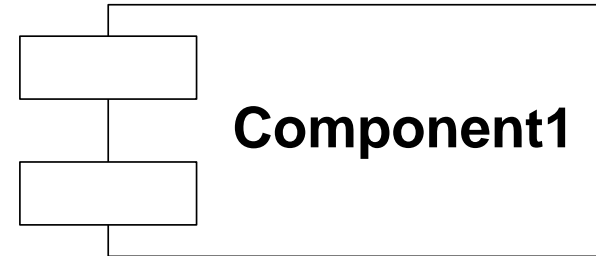
- Super Class
- Parent Class
- Generalized Class
- Specialized Class
- Sub Class
- Child Class



Component –Package- Subsystems

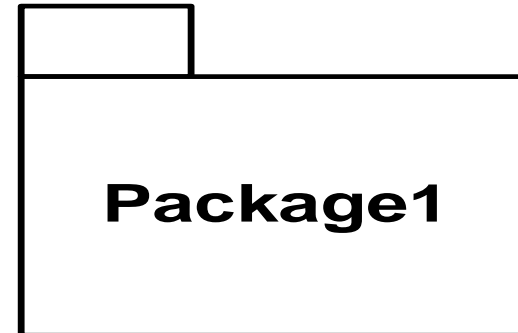
Component

Collection of Classes is Component.



Package

Collection of Components which are not reusable in nature



Subsystems

Collection of Components which are reusable in nature

Note:

Product Development Companies work on Subsystems and Application Development Companies work on Packages

Implementing OOA

Abstraction

Considering what is required, ignoring what is not required.

Encapsulation

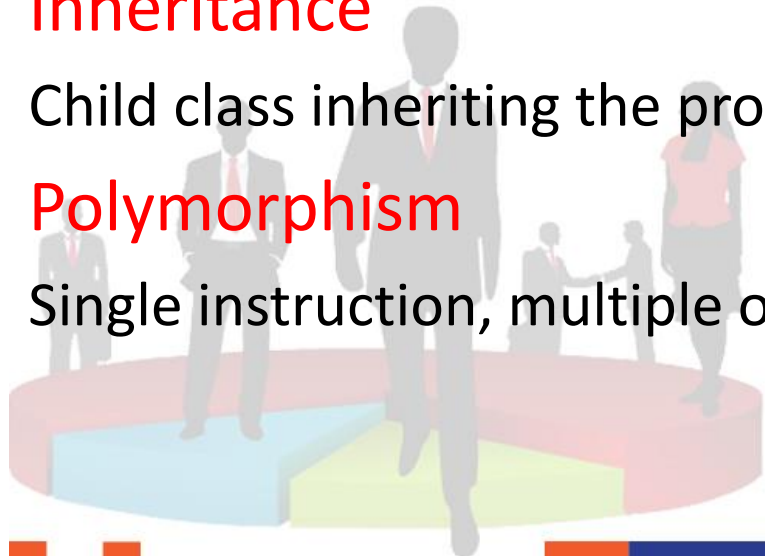
Information & Complexity hiding technique.

Inheritance

Child class inheriting the properties of Parent class.

Polymorphism

Single instruction, multiple operations.



Relationships

Relationships exists between classes or between objects, but not between class and an object.

There are 4 types of relationships.

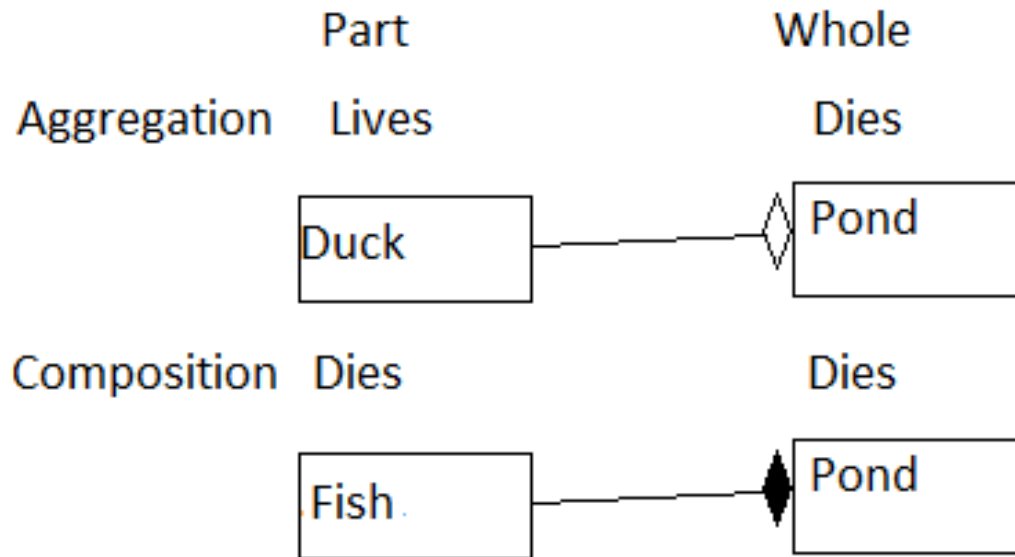
- **Association-** has a relationship
 - Unary- One Way
 - Binary- Two Way
 - Multiplicity- 1 to Many, Many to 1 or Many to Many
 - Reflexive- Single class with multiple role & one role is directed to itself.

- **Generalization**
- **Aggregation**
- **Composition**



Relationships

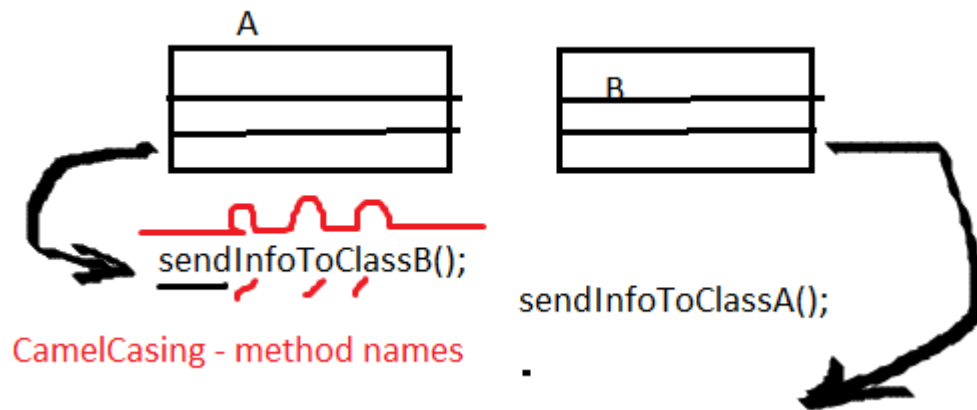
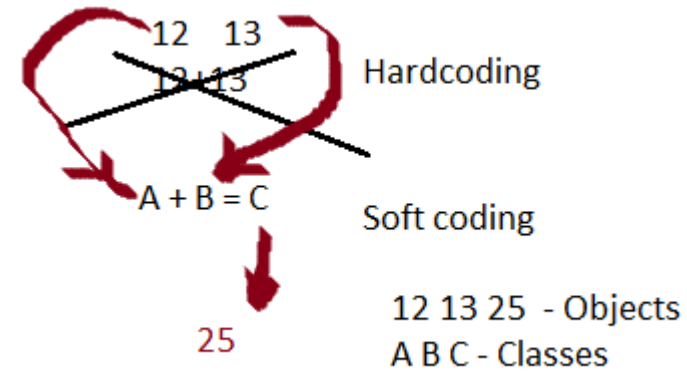
- Aggregation
- Composition



Classes are basic blocks of Code

Hard coding should be avoided, and soft coding should be encouraged

we will write code using Classes - try passing objects through Classes and achieve Functionality



UML - (Unified Modeling Language)

- UML diagram is known as language of diagrams.
- The base of UML diagram is object oriented approach (OOA)



- UML has 9 diagrams:
 - 5 Static
 - 4 Dynamic

UML - (Unified Modeling Language)

5 Static

- Use-case
- Class
- Component
- Packages
- Deployment

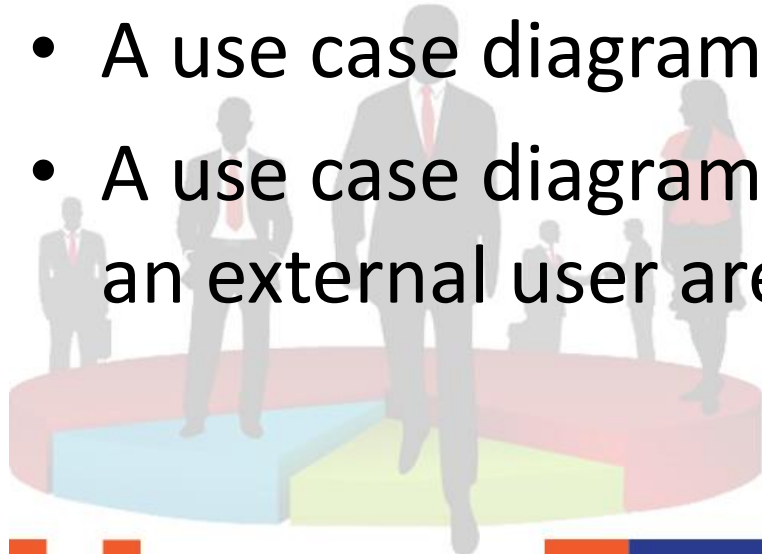
4 Dynamic

- Sequence
- Activity
- State Chart
- Collaboration



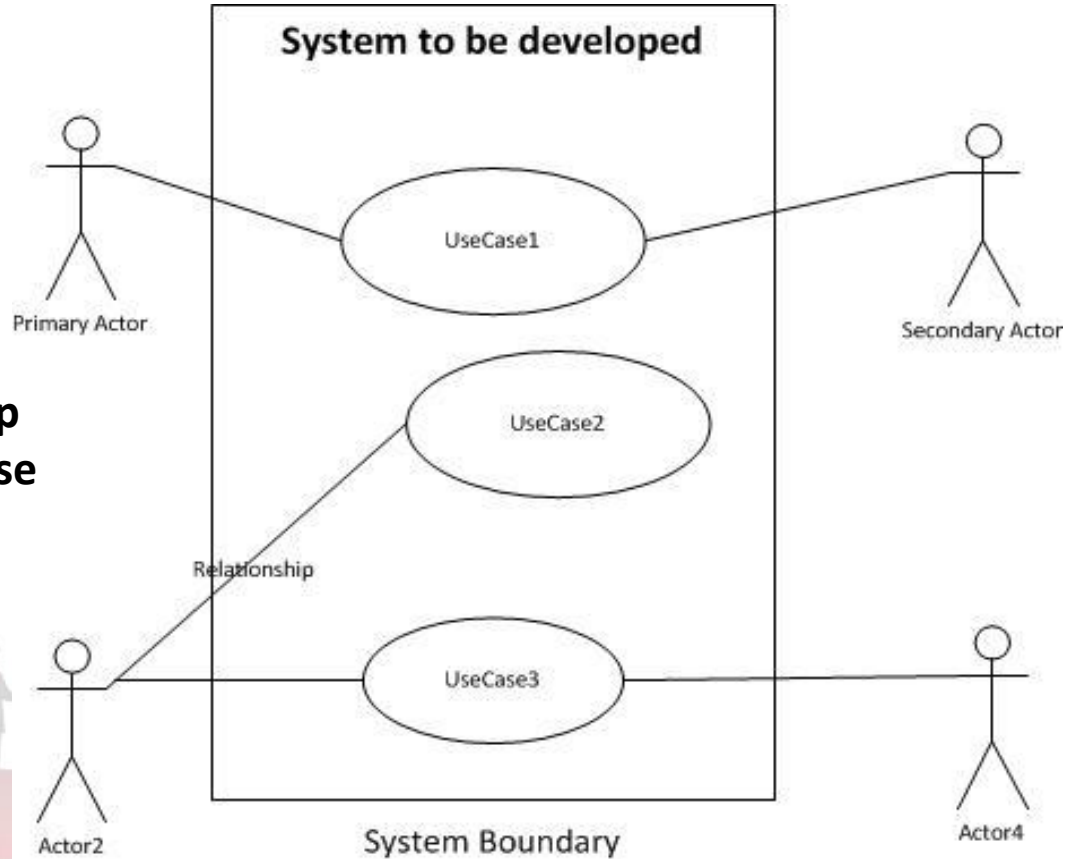
Use Case Diagram

- A Use case is a high level diagram.
- The main purpose of the diagram is to identify the requirement.
- A use case diagram is an actor specific.
- A use case diagrams are designed to explain how an external user are interacting with the system.



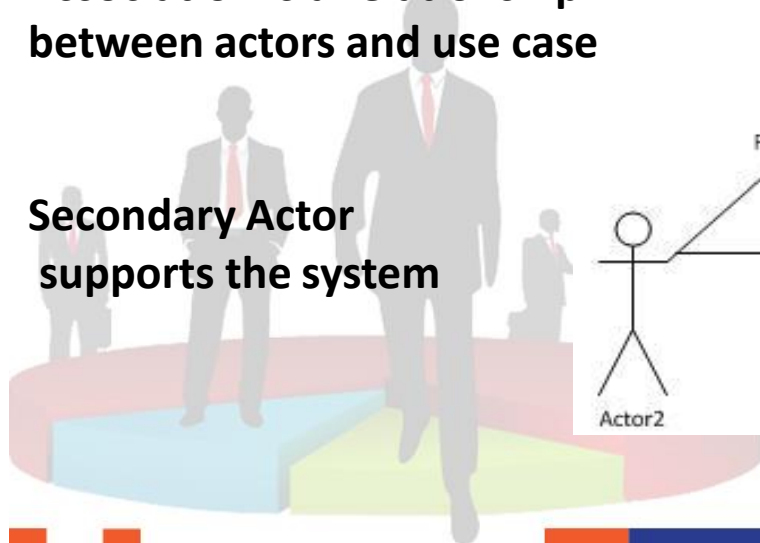
Use Case Diagram

**Primary Actor directly
interacts with the system**



**Association is a relationship
between actors and use case**

**Secondary Actor
supports the system**



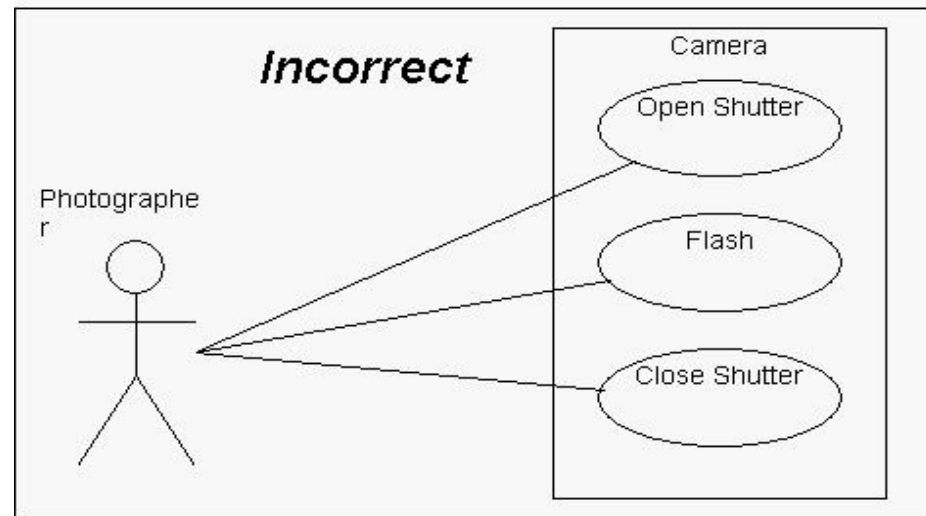
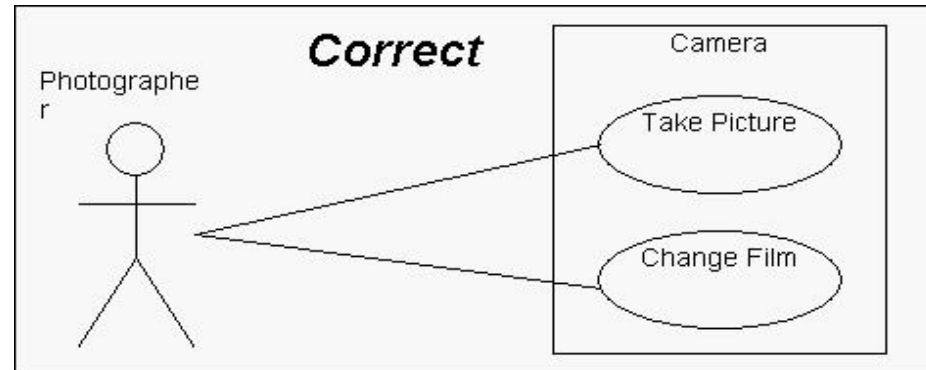
Use Cases – Essential/Supporting

Use cases are Verbs and are unique

Actors are Nouns

Essential Use Cases–
makes sense and completeness
to the end user

Supporting Use Cases –
makes sense and
supports Essential Use case



Use Case Diagram – Example - Basic

An ATM Application should have the following features:

- Customer should be able to withdraw Cash using the Card issued by the bank .
- After the Card is inserted into the Slot , 4 digit PIN Should be entered.
- If the PIN is Correct , then ATM will allow to do a Transaction.
- Three times incorrect PIN tried, The ATM will retain the Card.
- Customer should be able to deposit cash in the ATM
- Customer should be able to transfer funds within his Accounts
- Operator will refill the cash in the money Trays.
- All Transactions will be communicated by the ATM to the Bank and acquires **no objection** from Bank to perform that Transaction.

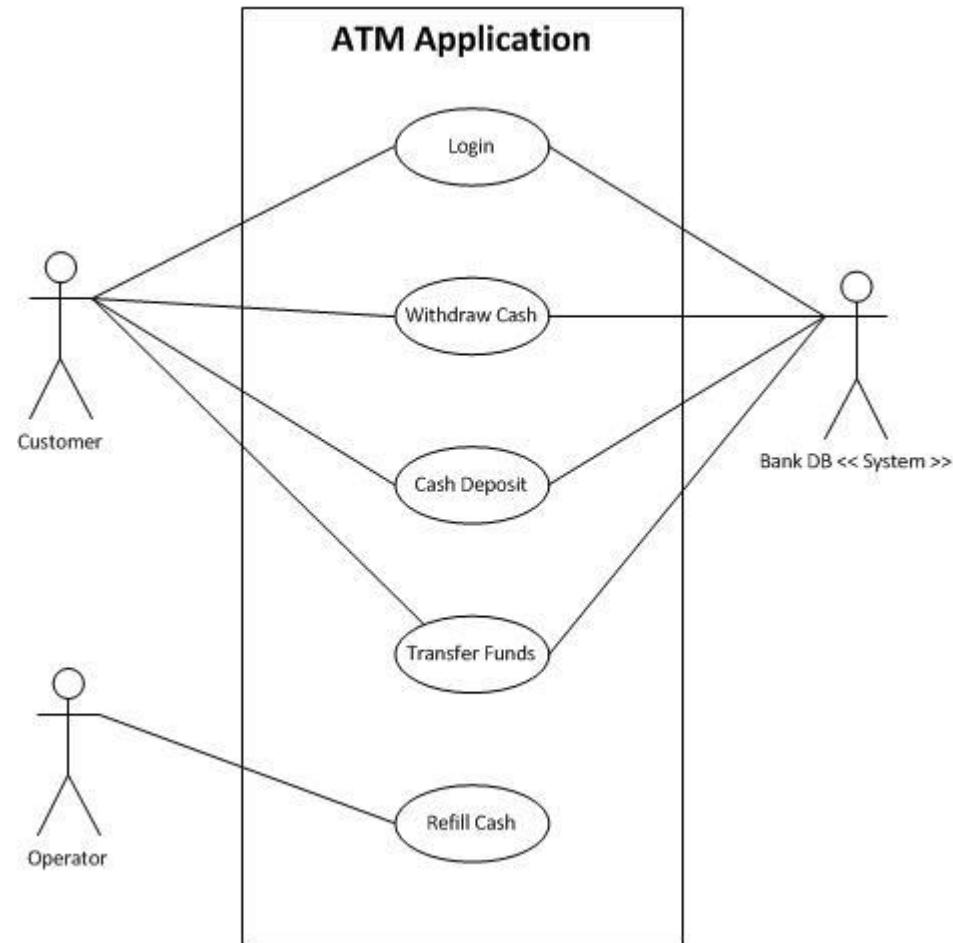
Use Case Diagram – Example - Basic

Use cases are Verbs and are unique

Actors are Nouns

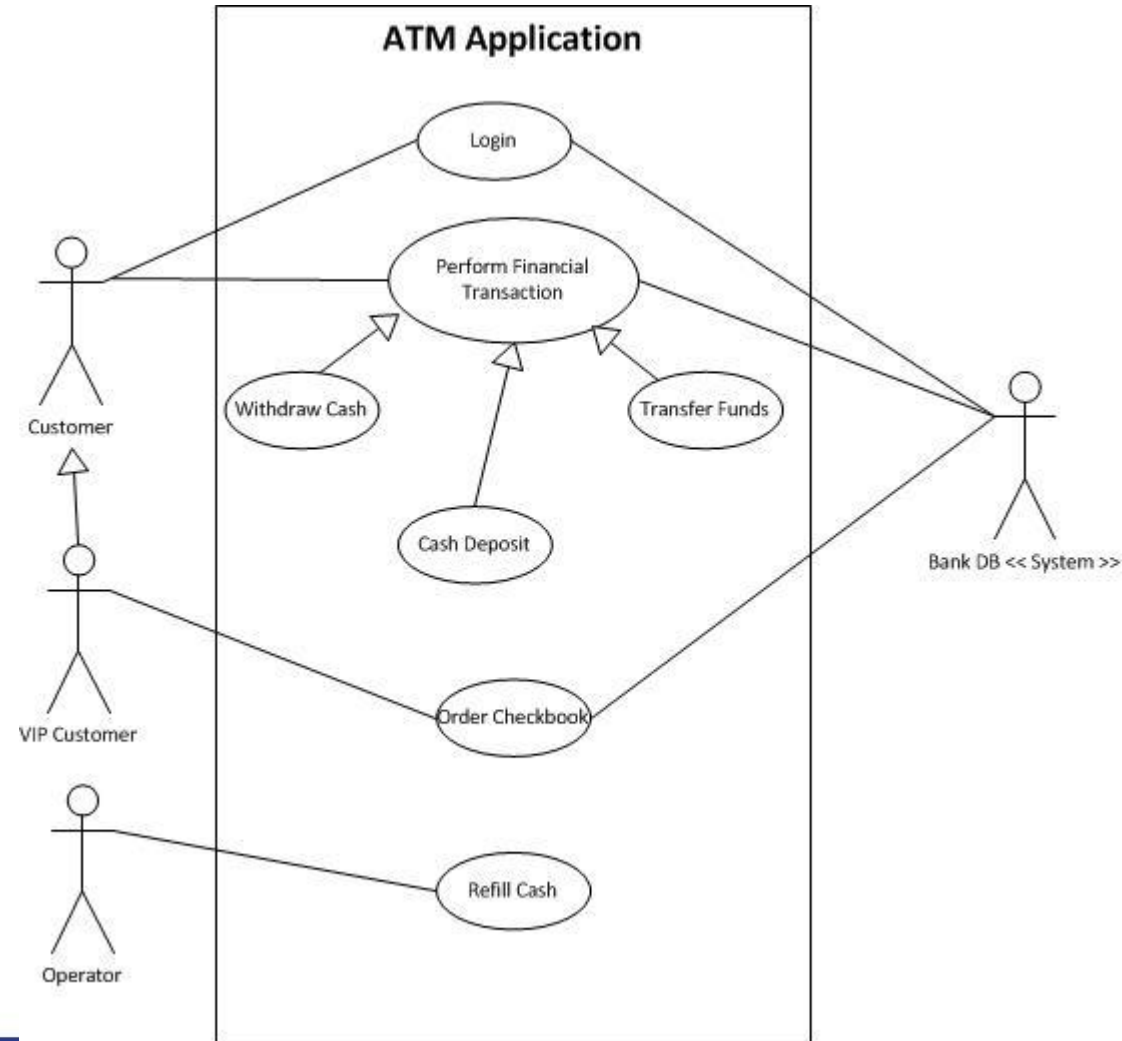
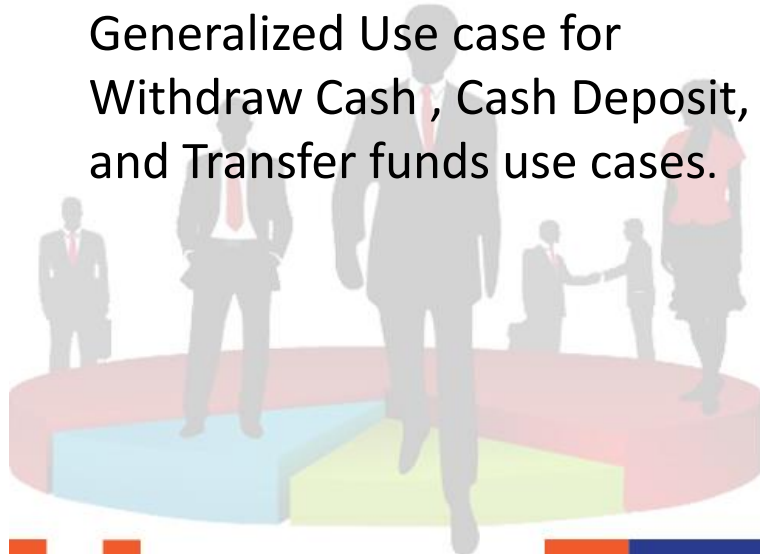
Essential Use Cases—
makes sense and completeness
to the end user

Supporting Use Cases –
makes sense and
supports Essential Use case



Use Case Diagram – Generalization

- A VIP Customer has an extra privilege to order Checkbook through the ATM.
- Customer is Generalized Actor and VIP Customer is a Specialized Actor.
- Financial Transactions is Generalized Use case for Withdraw Cash, Cash Deposit, and Transfer funds use cases.



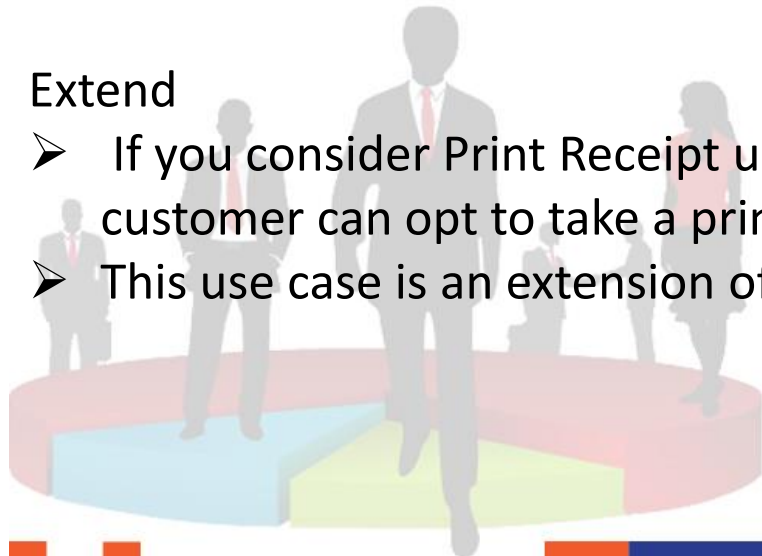
Use Case Diagram – Include / Extend

Include

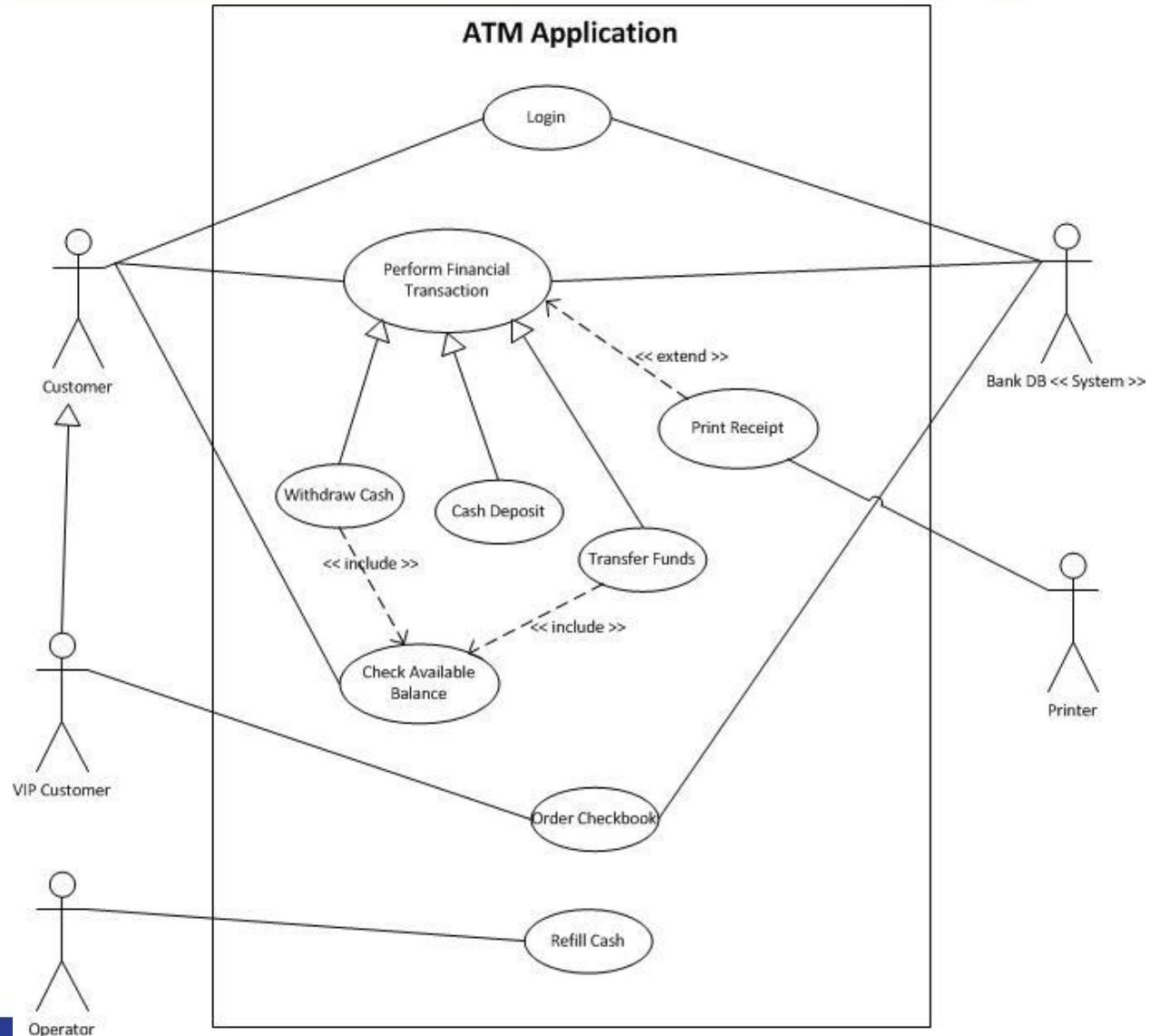
- Check Available Balance is a use case. This can be initiated by the Customer to check his / her Balance.
- Also when Withdraw Cash , Transfer Funds use cases are initiated, then Check Available Balance use case will be initiated and performed inherently.
- This is include relationship that exists between Withdraw Cash and Check available Balance. And also between Transfer Funds and Check Available Balance use case.

Extend

- If you consider Print Receipt use case , this is an optional use case, where the customer can opt to take a print out or not to take a print out.
- This use case is an extension of the Financial Transaction use case.

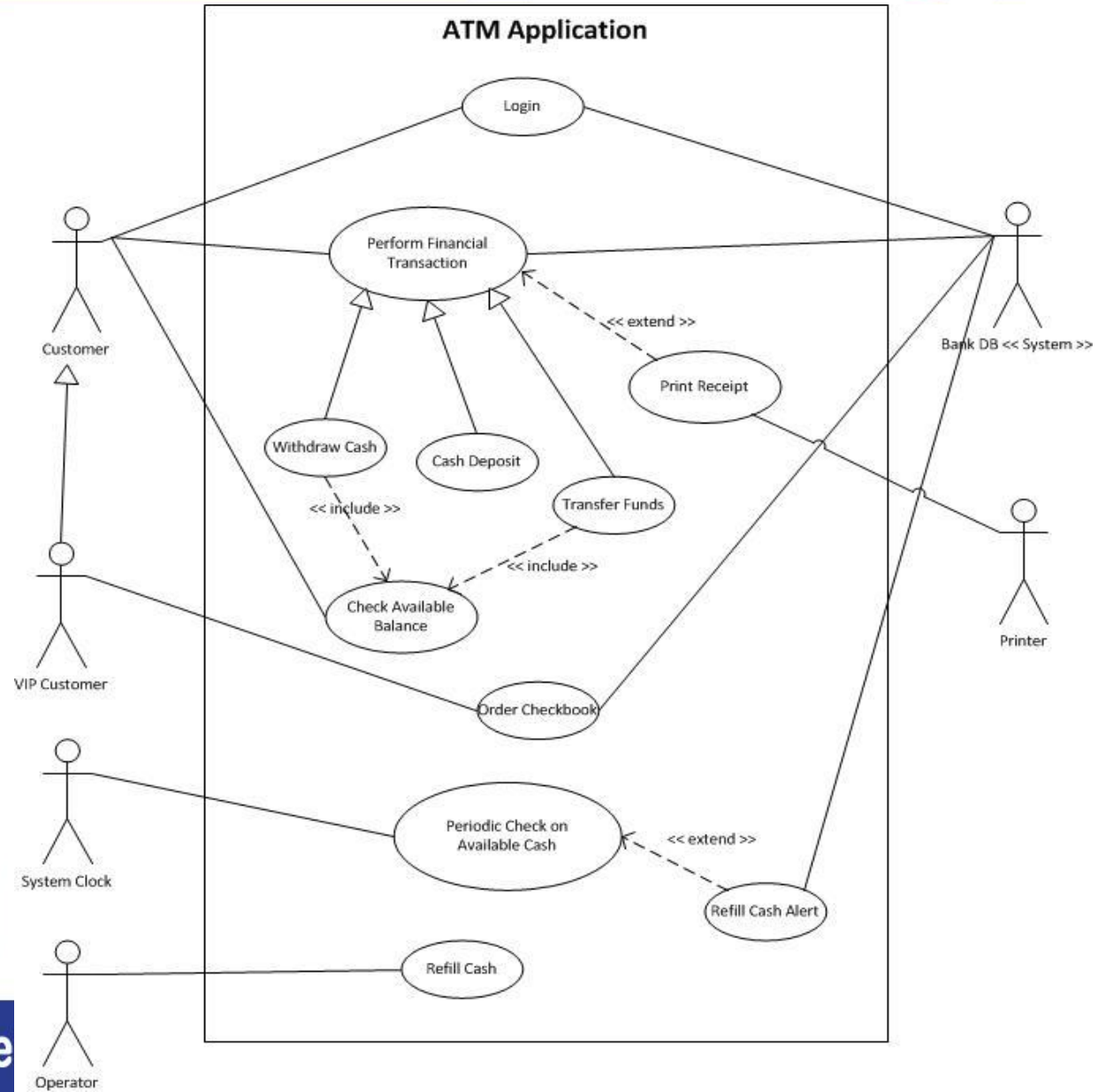


Use Case Diagram Include / Extend



Use Case Diagram System Clock Automation

When the Money tray in ATM reaches a minimum level of cash, It will automatically send an alert to the Bank . This can be explained as , periodically **the System clock** will be checking for available cash levels and if found less than the minimum level, then a alert is sent to Bank.



Use Case Diagram from a Case Study

1. Information which we **do NOT** model in Use Case diagrams are
 - Names of the systems (laptops, Desktops, Workstations),
 - Architectures (2 Tier, 3 Tier, n Tier, Client Server),
 - Databases Names (DB2, SQL Server, My SQL)
 - Networks (LAN, WAN, Internet),
 - Brand Names (HP, Lenova, Wipro, Sony),
 - Technology Names (Java, .Net, Mainframes)
2. Differentiate information against Actions
3. Write all sequence of Actions
4. Try to find out which actor is performing the above action
5. Try to identify Essential Use cases and Supporting Use Cases
6. Try to identify some modules with respect to functionality or usage.
7. Try to draw the relationships appropriately between the identified Actors and Use cases

Use Cases – Important Points

Generalization - is a kind of

[Parent Class exist through one (or more) of Child Classes]

Direction of Arrow – based on dependency[Not information flow]

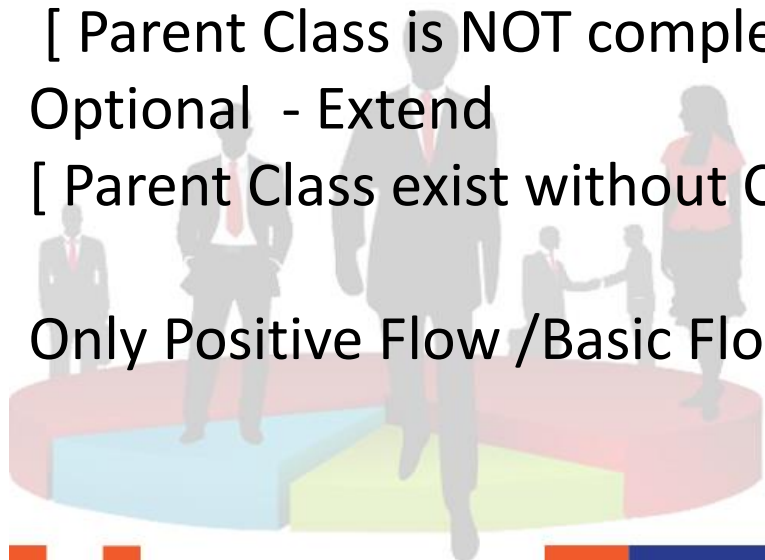
Compulsory – Include

[Parent Class is NOT complete without Child Classes]

Optional - Extend

[Parent Class exist without Child Classes]

Only Positive Flow /Basic Flow is modeled in a Use case Diagram



Use Case Description Document

Every Use case will have it's own use case Description Document or **Use case Specification**

1. Use case Name
2. Use case Description
3. Actors

Primary Actors

Secondary actors

4. Basic Flow
5. ALTERNATE FLOW
6. Exceptional flows

7. Pre- Conditions
8. Post- conditions
9. Assumptions
10. Constraints
11. Dependencies
12. Inputs and Outputs
13. Business Rules
14. Miscellaneous Information

Test Case Document

Test Case Id	PQ786TS003	Test Case Name	Search Query
Project ID	PQ786	Project Name	Online Flight Reservation
PM ID	4869	PM Name	Raman V
Test Strategy ID	PQ786TS001	Tester Id	
Test Plan ID	PQ786TP001	Tester Name	
Test Schedule ID	PQ786TS001	Date of Test	

Scenario : Explanation about Project and that particular Case

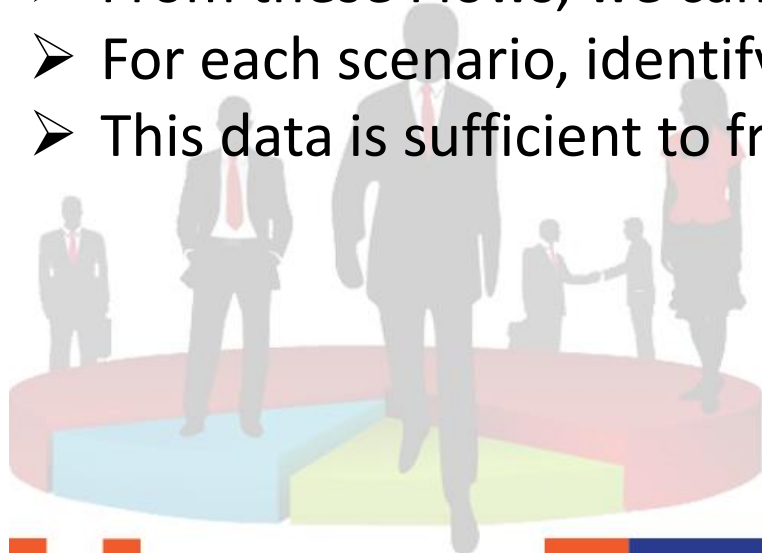
Example: Online Flight Reservation System In that Search for Flight... we have to 8 inputs... 6 compulsory and 2 optional.... Then press Search Button.... And after that results are displayed in a tabular way ...

Link to that page:

	Set 1	Set 2	Set 3	Set 4	Set 5
Input Data	1 Pune 2 Chennai 3 10 Dec YY 4 1 way 5 2 passengers 6 Economy 7 window 8 Veg	1 Hyderabad 2 Chennai 3 13 MAR YY 4 2 way 5 1 passengers 6 Business 7 Asle 8 Non-Veg	1 Nagpur 2 Mumbai 3 17 May YY 4 1 way 5 2 passengers 6 Economy 7 window 8 Veg	1 Delhi 2 Chennai 3 20 July YY 4 2 way 5 1 passengers 6 Economy 7 Asle 8 Veg	1 Pune 2 Delhi 3 25 Sep YY 4 1 way 5 2 passengers 6 Economy 7 window 8 Non-Veg
Expected Behavior	Eg: 10 flights should come for above data				
Actual Behavior					
Comments					
Result Pass/Fail					

Deriving Test Case out of Use Case

- Use- case Diagram
- Use- Cases
- Use-case Description Document
- In Use-case Description Document, we have Basic Flow, Alternate Flow and Exception Flows.
- From these Flows, we can identify the scenarios.
- For each scenario, identify at least 3 to 5 sets of valid test data.
- This data is sufficient to frame a Test Case.



3-Tier Architecture

Screens , Pages, Validations on Page,
Company Specific logic, Functionality

Application Layer

All Re-usable components, Frequently changing components,
Governing Body Rules & Regulations, Compliances ...
Example : Printer, Payment gateways, Mail Servers
RBI rules for Banks, IRDA rules for insurance

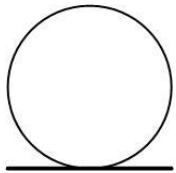
Business Logic Layer

Database components connecting to Databases

Data Layer



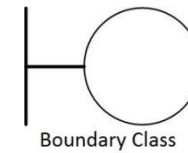
MVC Architecture



Entity Class
Data base classes ,
Persistent class
(Back end designers)



Controller Class
Transient Class
(Given to Front end
designers)



Boundary Class
(or) FORM Class



Model – View – Controller (MVC)

MVC Architecture Rules

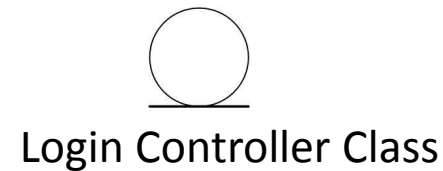
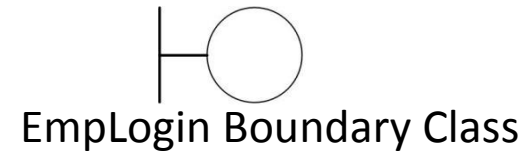
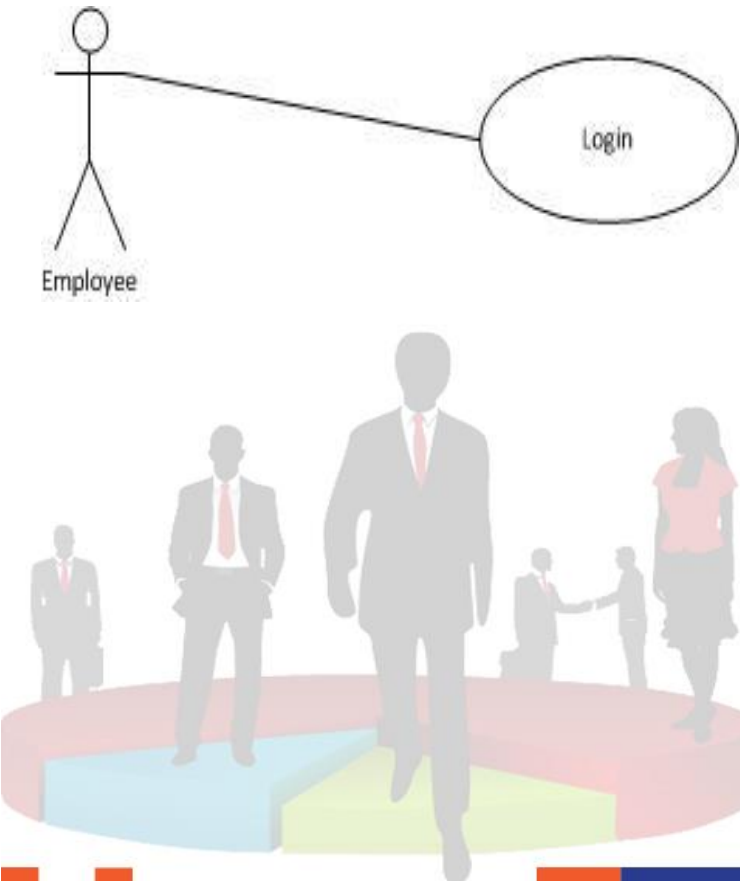
1. Combination of One Actor and an use case results in one Boundary class
2. Combination of Two Actors and an use case results in two Boundary classes
3. Combination of Three Actors and an use case results in Three Boundary classes and so on....

Note: only one primary actor is to be considered with a use case.

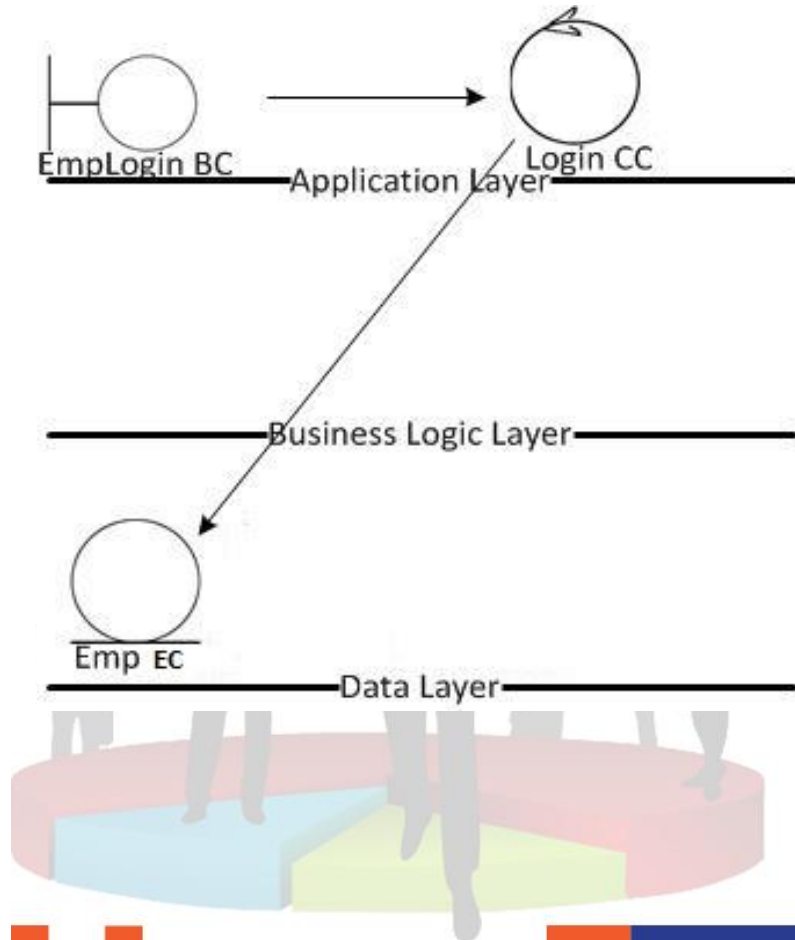
4. Use case will result in a controller class
5. Each Actor will result in one entity class

MVC Architecture - Example

Applying MVC Rules we get the Classes as



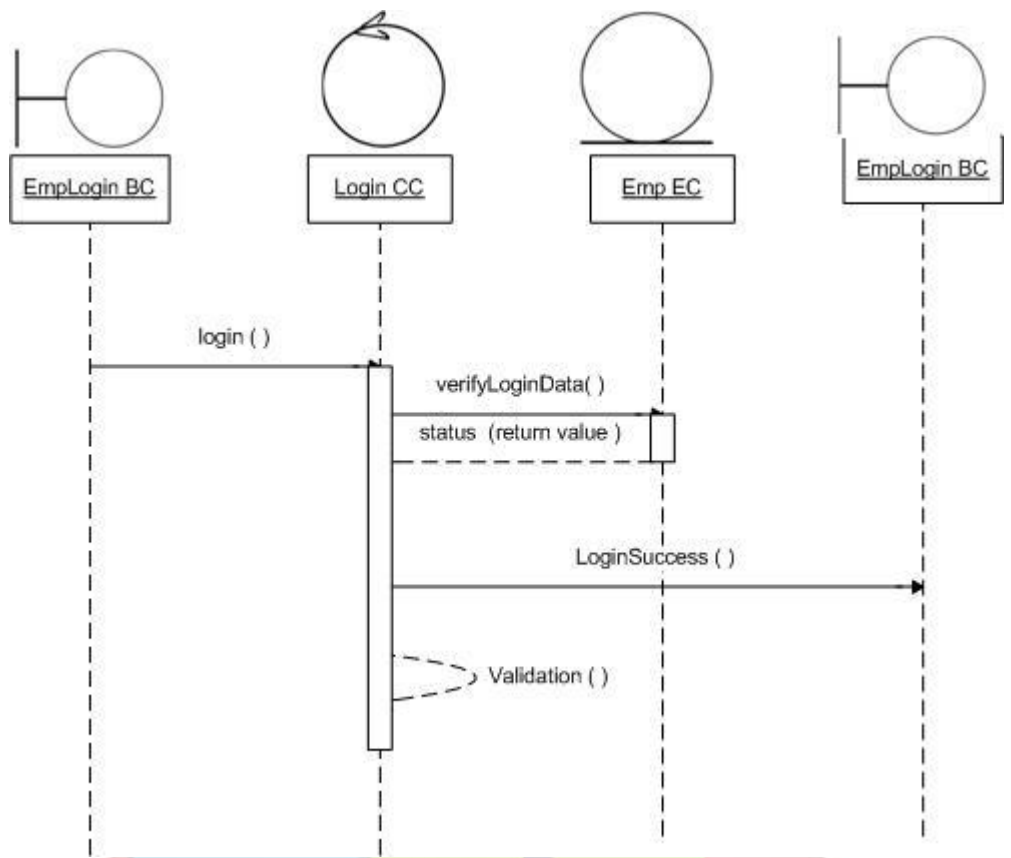
MVC Classes in 3 Tier -Guidelines



Guidelines to place identified MVC Classes in a 3 Tier Architecture

- Place all Entity Classes in DB Layer
- Place Primary Actor associated Boundary Class in Application Layer
- Place Controller Class in Application Layer
- If governing Body influence or Reusability is there with any of remaining Boundary Classes , place them in Business Logic Layer else place them in Application Layer

Sequence Diagram



Lifeline: Lifeline indicates the life of an class (here the Class)

Camel Casing: entire first word will be in lowercase and subsequent words first letter should be in Upper Case . There will be no gap in between words. Example: getEmpId(); turnLeftAndSlowDown();

Return message: This will always flow towards the Controller; this is just a message and NOT a method.

Note: Method always flows in timeline. Never Method will flow backward in timeline.

Focus of Control: It shows the life of method.


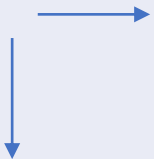


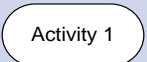
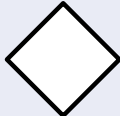
Activity Diagram

An Activity diagram is drawn to model how the system should function in order to achieve Business Logic, Business Functionality and Business Objectives.

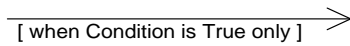
Activity diagram is basically a flow chart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. So the control flow is drawn from one operation to another. Activity diagrams are not only used for visualizing dynamic nature of a system but they are also used to construct the executable system by using forward and reverse engineering techniques.

Flow Chart	Activity Diagram
<ol style="list-style-type: none"> 1. Photographer visits the property 2. Photographer clicks pics of the property 3. Photographer uploads pics in the portal 	<ol style="list-style-type: none"> 1. Accept Property Pics from the Photographer

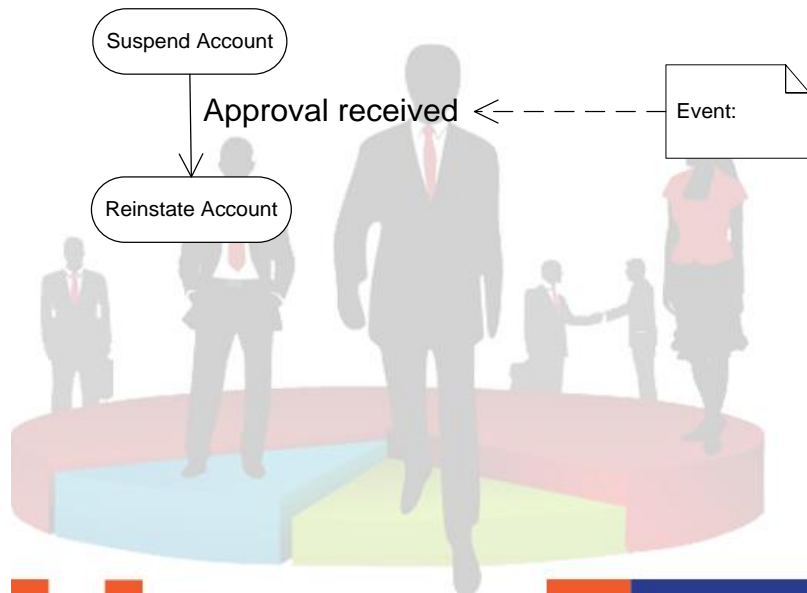
Activity Diagram – Drawing Elements

Sl No	Symbol	Name	Description & Example
1.		Initial Node Start Node	Indicates where the workflow begins.
2.		Control flow	An arrow showing the direction of the workflow. Generally, control flows from top to bottom and from left to right.
3.		Final node End Node	Indicate that the workflow is completed
4.		Stop Node	When a deadlock happens and no more we can progress in that workflow. Example three times PIN tried wrong in an ATM
5.		Activity	Indicates a step in the process. It is a unit of work done by the system or a consistent state achieved.
6.		Decision Box	A diamond symbol, indicating a choice. Workflow will proceed along one of a number of possible paths, according to the guard conditions

Activity Diagram – Drawing Elements

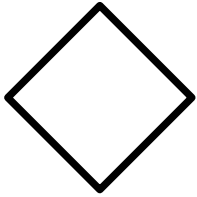


Guard Condition - A condition attached to a control flow. When the guard condition is true, workflow may flow along the control flow. Guard conditions are usually attached to control flows that come out of a decision symbol. (However, they can also be used without the decision symbol.) A guard is shown within square brackets.

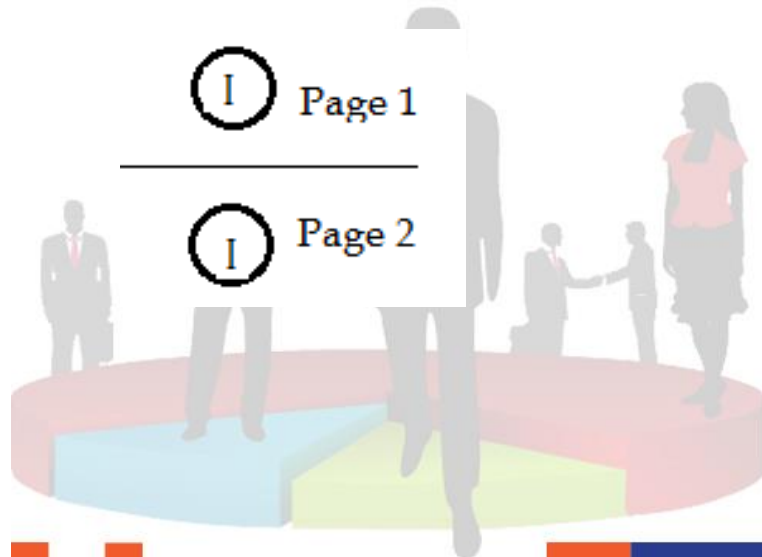


Event - A trigger attached to a control flow. An event must occur for the flow to move along the control flow. Declaring something as an event has a stronger implication than a calling it a guard. Event is externally triggered or it can be a consistent point of a defined time frame.

Activity Diagram – Drawing Elements

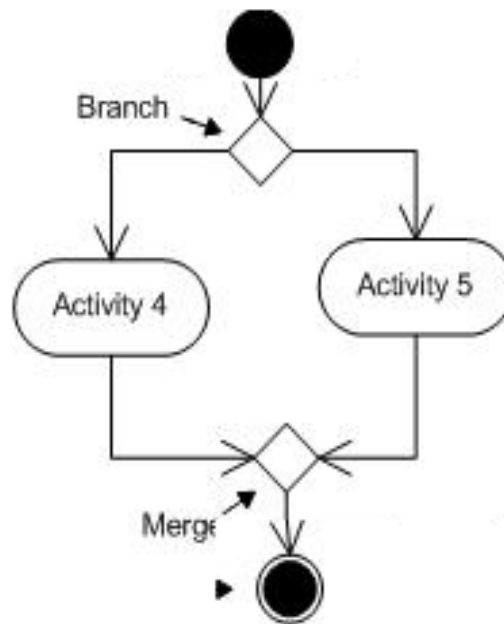


Connector - A connector has multiple inputs and multiple outputs. Practically we use connectors for 1- many and many-1. .



Continuation Node - If Activity is big and is spanning pages . We give continuity from one page through another page by using Continuation Nodes. It can also be used to connect extreme left to right sides in a page or for reusability of a particular flow

Activity Diagram – Drawing Elements

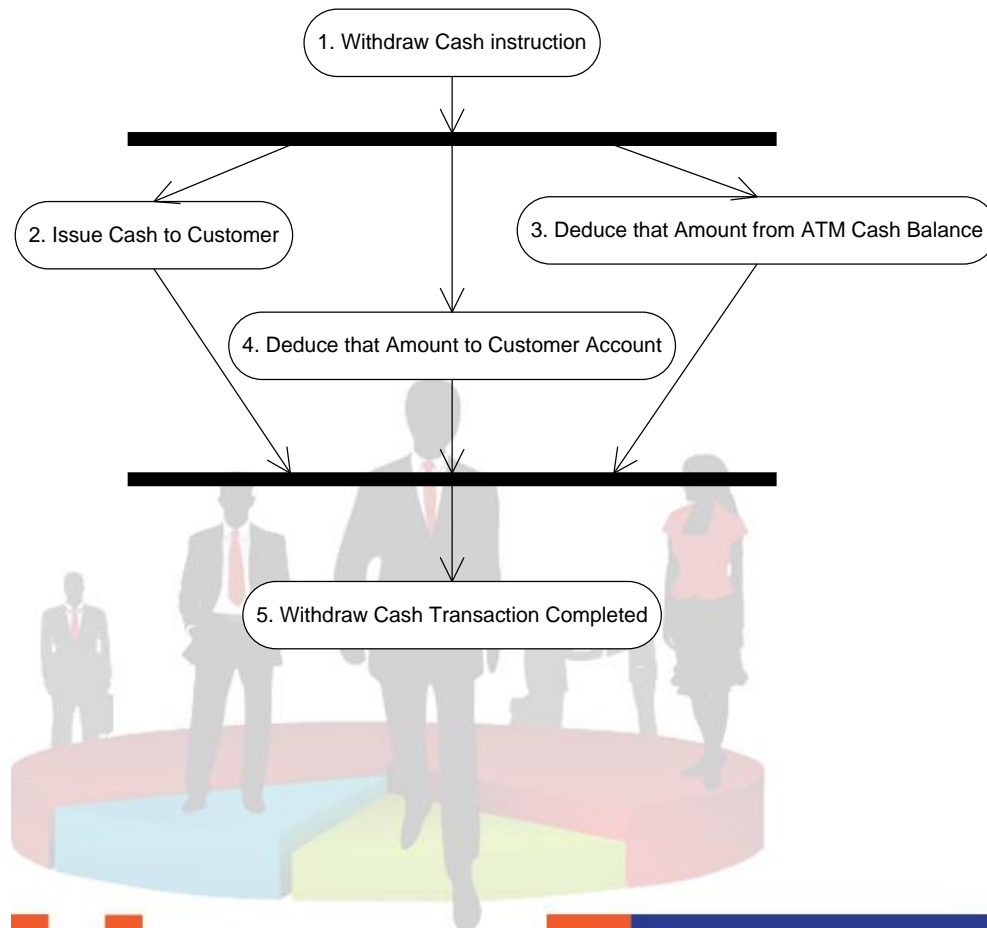


Branch And Merge

Between a Branch n Merge, we can have n number of paths. For the control to pass from branch to merge at least 1 path should execute. The condition between the paths will be “OR”

Example : Payment by cash, card, net banking coupons

Activity Diagram – Drawing Elements

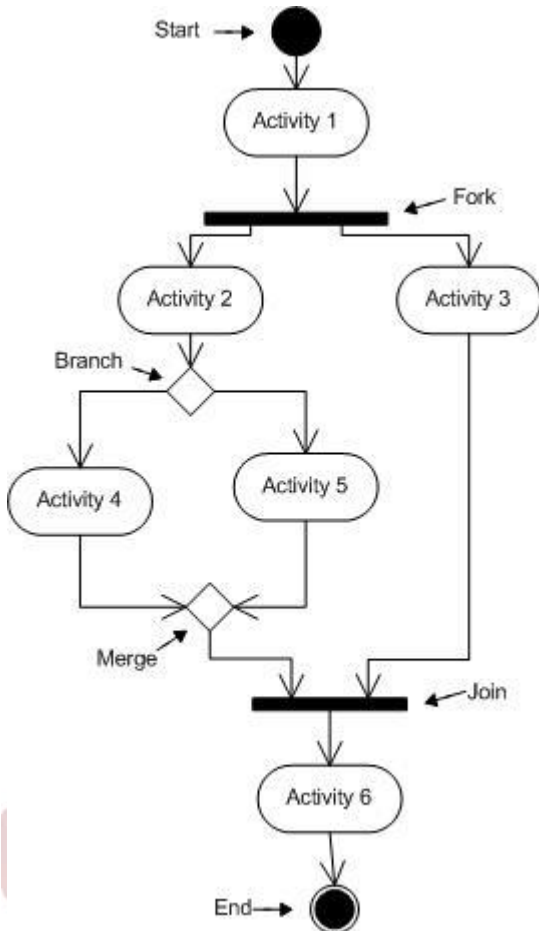


Fork and Join

Between a Fork n Join, we can have n number of paths. For the control to pass from Fork to Join all the paths should execute. The condition between the paths will be “AND”

Example : Cash Withdrawal from ATM

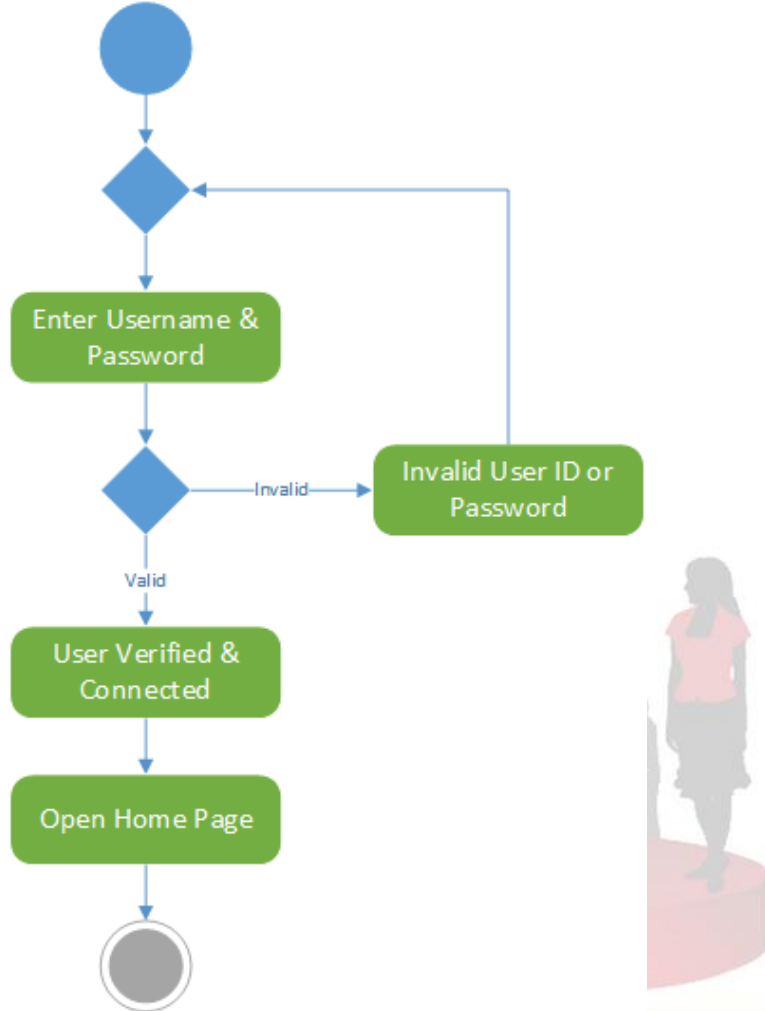
Activity Diagram – Drawing Elements



Nested Loop Activity

This is an activity wherein this activity may contain a series of activities internally.

Activity Diagram – User Login Page



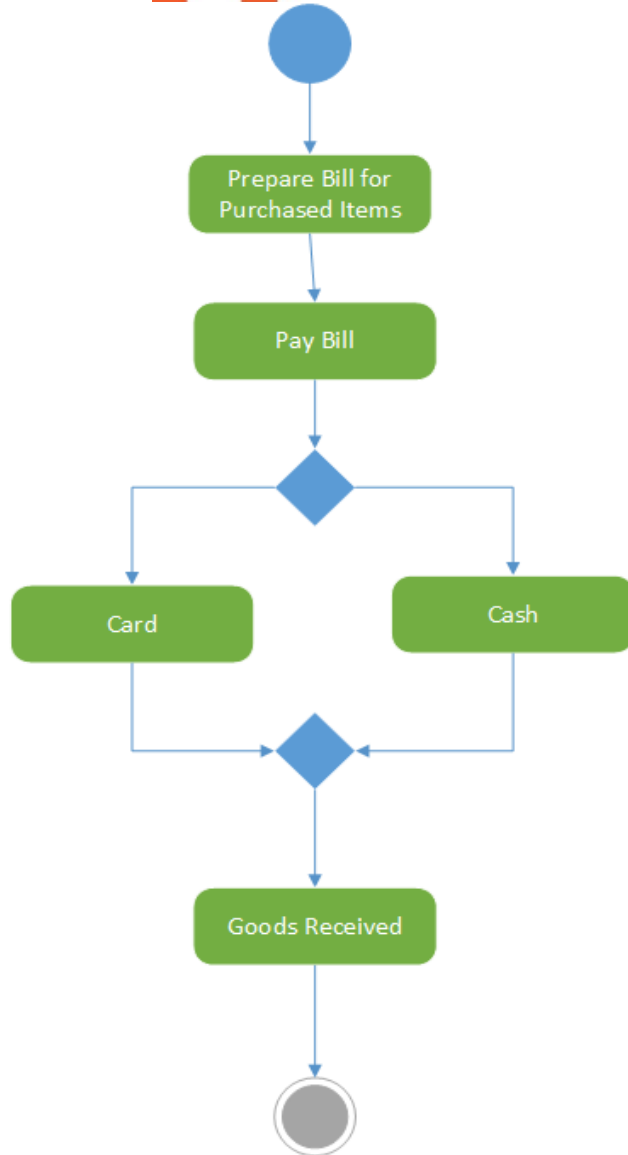
Explanation

First Username and password is entered

If credentials are correct
It opens the home page

Else Invalid Username and Password is
prompted

Activity Diagram – Example



Explanation – Retail Store – Bill payment by cash or Card

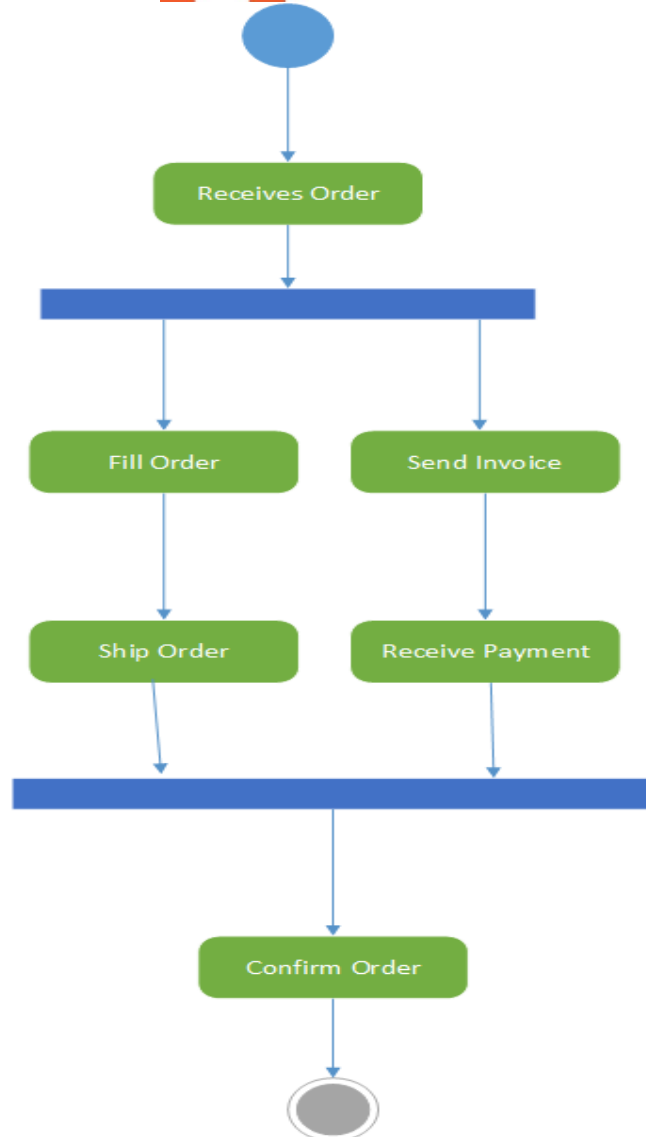
The customer buys the items from the retailer.

The retailer prepares bill for the goods purchased by the customer,

he receives amount by credit or by cash from customer.



Activity Diagram – Example



Explanation – receives order and confirms the order

An Ecommerce Company receives an order from customer. The inventory department will fill the order and ship the order whereas the finance department will send invoice and receive invoice. At the end, order confirmation notification will be send to customer.



Activity Diagram – Important Points

1. Activity Diagrams are drawn to **simplify Complexity**

2. Activity Diagrams can be **drawn at any level**

* Project wise, module wise, department wise, functionality wise, use case wise

3. Activity Diagrams can be **drawn based on frequency**

* Setup, 1 Time, Annual, Half-yearly, Quarterly, Monthly, Weekly, Daily, Predefined Frequency and Adhoc basis - specific to an initiative

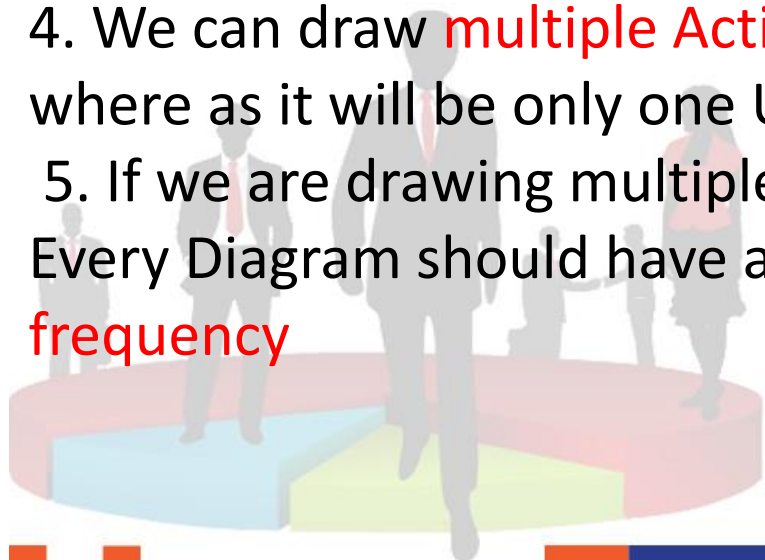
4. We can draw **multiple Activity diagrams for a single Case study**

where as it will be only one Use case diagram.

5. If we are drawing multiple Activity Diagrams for a single case study

Every Diagram should have a **heading based on level and time**

frequency



Activity Diagram – Swimlanes

- Swimlanes divide activity diagrams into sections. Each swimlane is separated from adjacent swimlanes by vertical, solid lines on both sides.
- Each action is assigned to one swimlane.
- Activity flows can cross lanes.
- We cannot model an Activity in-between Swimlanes
- Swimlanes do not change ownership hierarchy.
- The relative ordering of swimlanes has no semantic significance.
- There is no significance to the routing of an activity flow path.
- Parts representing internal behavior can be specified on swimlanes.
- Have Less Than eight Swimlanes.

Requirements Engineering

Requirements Engineering starts from

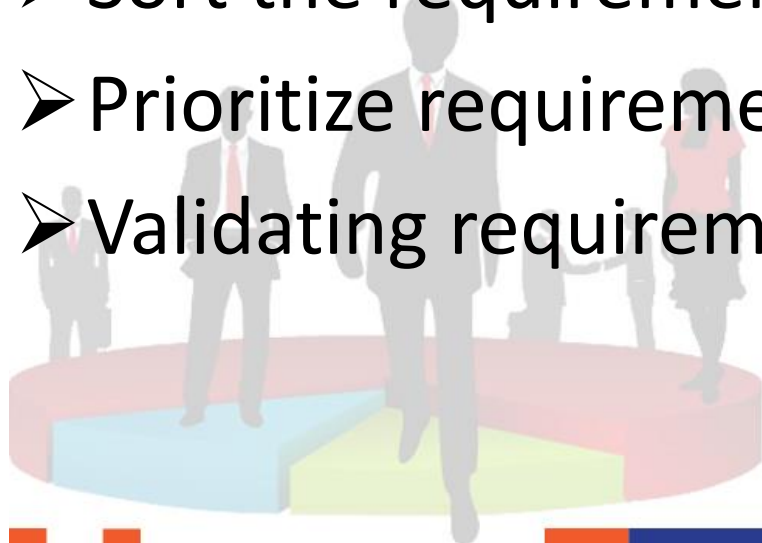
- identify the sources of Requirements i.e. stakeholders,
- Elicit Requirements, Document,
- Model and Confirm the Requirements,
- Prioritize and Validate the requirements,
- Communicate and Manage the Requirements and
- facilitate Solution Assessment and Implementation

And is best understood by 3 Steps

1. Business Requirements Initiation (Gathering Stage)
2. Business Requirements Management
3. Business Solutions Evaluation and Implementation

Business Requirements Initiation (Gathering Stage)

- Stakeholder Analysis
- Apply requirement elicitation techniques
- Sort the requirement
- Prioritize requirements
- Validating requirements



Stakeholder Analysis

Stakeholder analysis is based on ILS & RASCI Matrix

- ILS (Identify, List & Summary)
- RASCI Matrix (Responsible, Accountable, Supporting, Consulted and Informed)



Requirement Elicitation Techniques

- Document Analysis
- Reverse Engineering
- Focus Groups
- Observations
- Workshops
- JAD
- Interview
- Prototyping
- Survey/ Questionnaire
- Brainstorming
- Usecase Specs



Document Analysis

Document analysis is done through reading a document and understanding the product, process and project.



Reverse Engineering

- Reverse engineering, also called back engineering, is the processes of extracting knowledge or design information from anything man-made and re-producing it or re-producing anything based on the extracted information.
- The process often involves disassembling something and analyzing its components and workings in detail. Majorly used in migration projects.

Focus Groups

A focus group is a means to elicit ideas and attitudes about a specific product, service or opportunity in an interactive group environment.



Observations

Observing, shadowing users or doing a part of their job, can provide information of existing processes, inputs and outputs.



Workshop

A requirement workshop is a structured approach to capture requirements. A workshop may be used to scope, discover, define, prioritize and reach closure on requirements for the target system.



JAD (Joint Application Development)

Application developed through JAD has higher customer satisfaction and less number of errors as user is directly involved in the development process.



Interview

Interview of a user and stakeholders are important in creating software.

An interview is a systematic approach where interviewee is going to ask relevant questions related to software and documenting the responses.



Prototyping

Prototyping is an attractive idea for complicated and large systems for which there is no manual process or existing system to help determining the requirements.



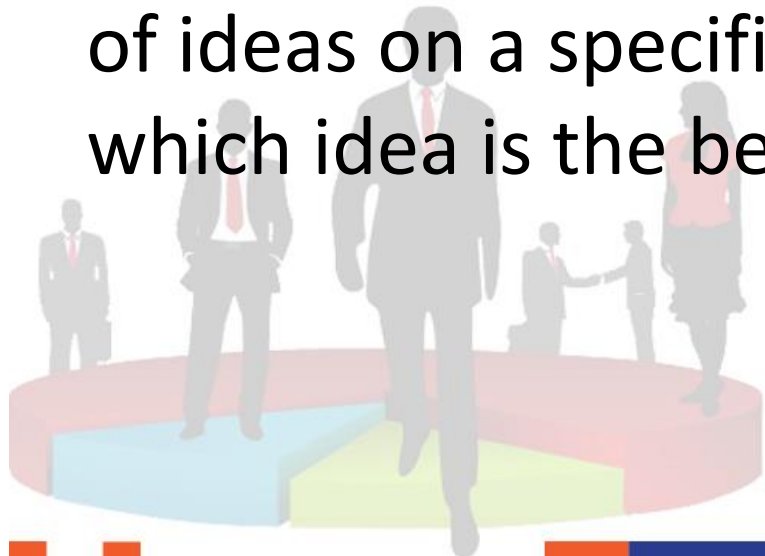
Survey/ Questionnaire

Questionnaire can be useful for obtaining limited system requirements details from the users/ stakeholders, who have minor input or are geographically remote.



Brainstorming

- Brainstorming can be done either individually or in groups. The ideas collected during the brainstorming session are reviewed or analyzed.
- Brainstorming is an effective way to generate lots of ideas on a specific issue and then determine which idea is the best solution.



Prioritize the Requirement

“MOSCOW” Technique:

MOSCOW is a prioritizing technique which is used in business analysis and software development to reach a common understanding with stakeholders on the importance of each requirement.



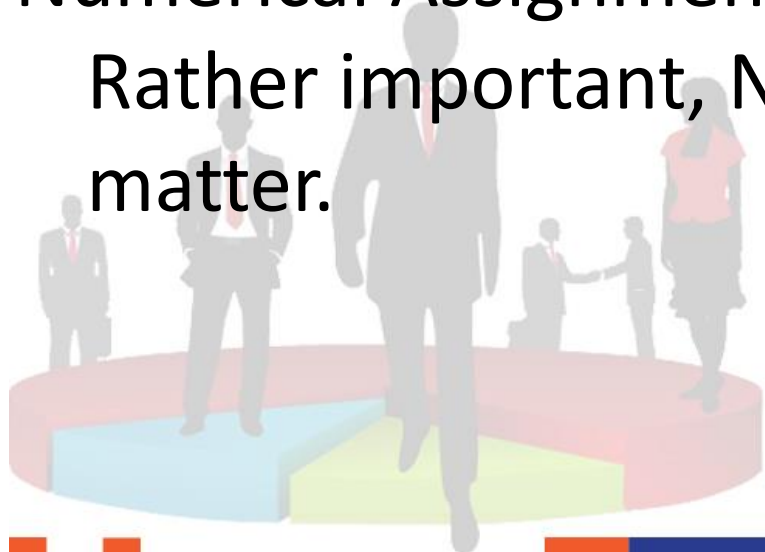
Prioritize the Requirement

Other Techniques are:

100 Dollars Test

Top 10 requirements

Numerical Assignment- Mandatory, Very important,
Rather important, Not Important, does not
matter.



Validating Requirement

“FURPS” Technique

This technique is used to validate must requirement.

F- Functionality

U- Usability

R- Reliability

P- Performance

S- Supportability (Extendable, Testable & Enhance-able)

Other Validating Techniques are

CUCV

- Clarity
- Understandable
- Consistent
- Verifiable

CAE

- Complete
- Accurate
- Executable

APVU

- Authorized
- Prioritized
- Verifiable
- Unique

SMART

- Specific
- Measurable
- Attainable
- Realistic
- Traceable/ Time Bound

Business Requirement Management

Requirements Management is all about processing the elicited requirements to implement the most suited solution.

Requirements tracing, a process of documenting the links between the requirements and the work products developed to implement and verify those requirements.



Business Requirement Management

1. Requirements Communication

- 3R Concept

2. Requirements Management

3. Requirements Organization

- Requirements Definition
- Requirements Modeling
- Requirements Verification

Business Solutions Evaluation and Implementation

1. Business Solutions

2. Solution Assessment

- Assess proposed Solutions
- Requirements Allocation
- Organizational readiness Assessment

3. Solution Validation

- Verification Vs Validation

4. Solution Evaluation

5. Solution Implementation

Guidelines to probe into Requirements

While Asking Questions, We should probe into

1. 5W 1H of that concept (Why, What, Who, Where, When and How) and
2. confirm the Requirement is SMART before accepting it for development.
3. Stakeholder analysis – RACI Matrix
4. Refer to 3 Tier Architecture.

Application Layer –
Business Logic layer
Database layer

5. UML Diagrams
 - Use case
 - Use case Spec
 - Activity Diagram

Guidelines to probe into Requirements 2

6. Models – (Design in BABok V3)

Domain Model

Conceptual Model

Data Model

DFD

ER Diagram

7. Screens / Pages are consequence of Matured Functional Requirement.

8. Sign offs (Confirmations) should be taken on all Documents, Diagrams, SDLC Stages from responsible Stakeholders.

9. Any Information which you gather should fit into any one of the above 6 sections described, otherwise it is just an information or a non-functional requirement.

Guidelines to probe into Requirements 3

As a Business Analyst, kindly understand that we are trained to extract requirements from the Stakeholders

and the Stakeholders are
NOT Trained to share requirements with the Business Analyst.

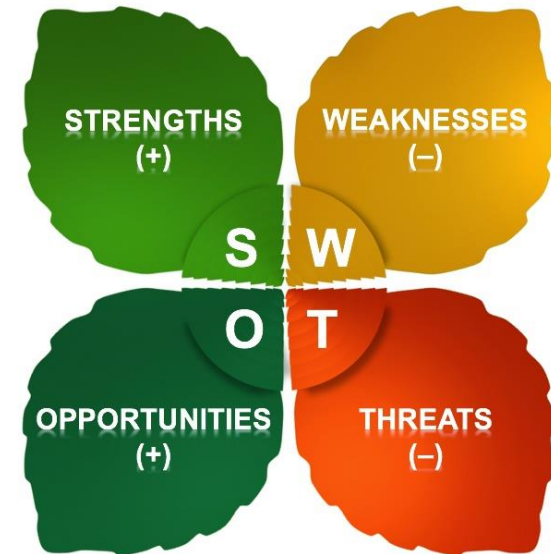


Enterprise Analysis

- Enterprise analysis (also known as strategic enterprise analysis or company analysis) is defined as focusing “on understanding the needs of the business as a whole, its strategic direction, and identifying initiatives that will allow a business to meet those strategic goals.”
- Enterprise analysis involves a thorough examination of not only the business problem (need) and its proposed business solution (if one already exists), but also an in-depth look into whether the proposed solution is truly the best solution, a detailed analysis of what the solution entails, its risks, and its feasibility in the existing organizational climate.

SWOT Analysis

- SWOT, which stands for strengths, weaknesses, opportunities and threats, is an analytical framework that can help your company face its greatest challenges and find its most promising new markets.
- A SWOT analysis is an organized list of your business's greatest strengths, weaknesses, opportunities, and threats.
- Strengths and weaknesses are internal to the company (think: reputation, patents, location).
- Opportunities and threats are external (think: suppliers, competitors, prices).



Feasibility Study

The feasibility study focuses on helping answer the essential question of “should we proceed with the proposed idea”.

Possibility of doing a project within some constraints like technology, budget & time.

Technology includes Software, Hardware and Trained Resources

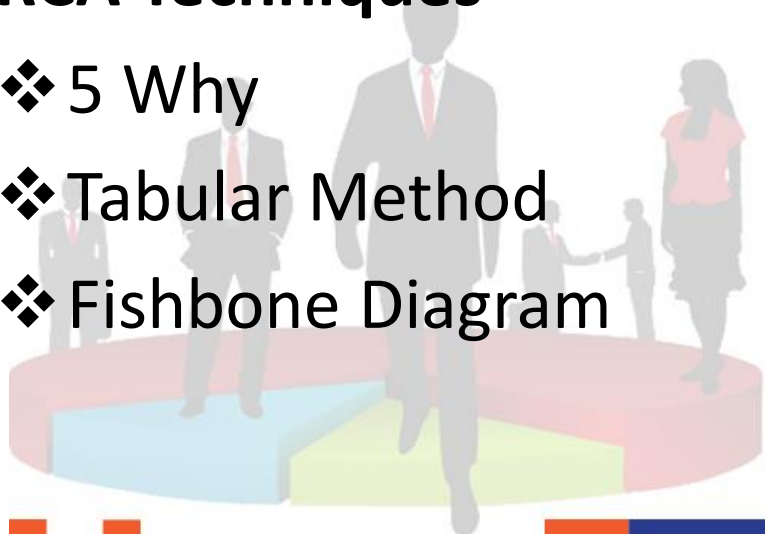


Root Cause Analysis

An event which hinders the smooth flow of the process is termed as an issue and the reoccurrence of the same event over a period of time is termed as problem.

RCA Techniques

- ❖ 5 Why
- ❖ Tabular Method
- ❖ Fishbone Diagram



Root Cause Analysis – Fishbone Diagram

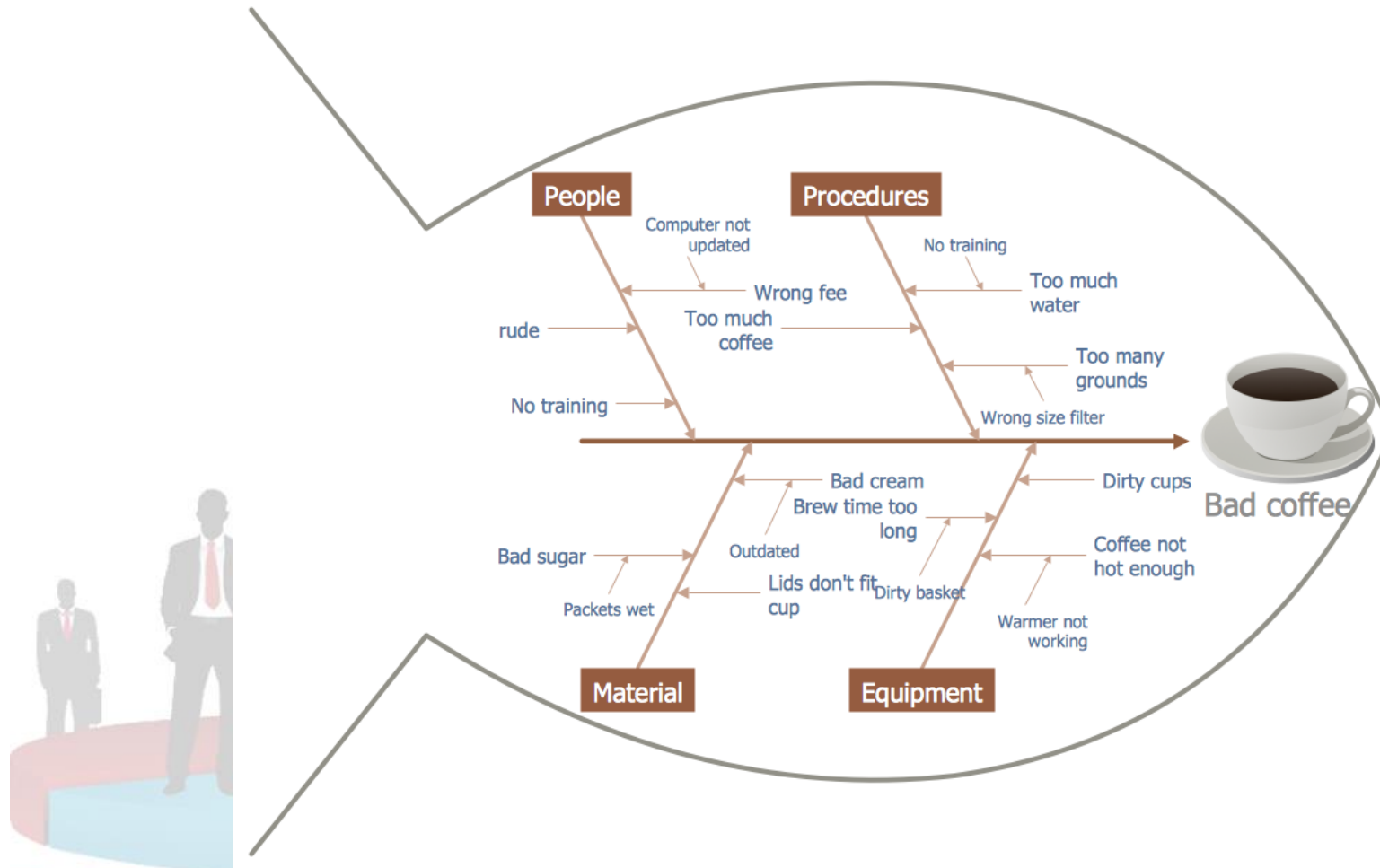
The Root Cause Analysis Process is also known as the “Ishikawa Diagram,” the “Fishbone Diagram,” and the “Cause-and-Effect Diagram.”

These tools make it possible to identify all of the roots (basic causes) in a retrospective approach, or, all of the potential effects (possible outcomes) in a prospective approach.

Ishikawa identified five (5) key areas which occur repeatedly in either type of analysis:

People – Processes – Machines – Materials - Environment

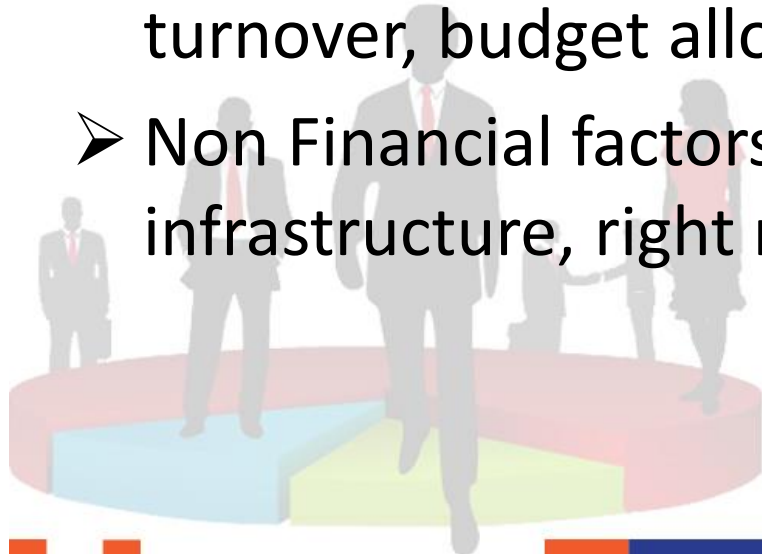
Root Cause Analysis – Fishbone Diagram



Decision Analysis

Decisions and their resulting outcomes can be either financial or non-financial in nature.

- Financial factor like Company turnover, quarter wise turnover, budget allocation for new project etc.
- Non Financial factors like required skill set to do project, infrastructure, right resource to do the project etc.



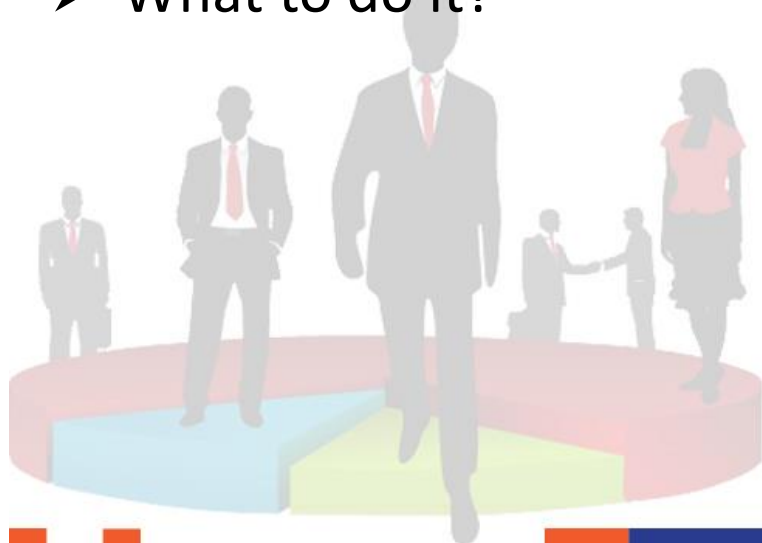
Strategy Analysis

Strategy is the direction and scope of an organization for the long-term, which helps in getting profits for the business or units through its implementations.

- How to do it?
- What to do it?

Effecting Factors

- External Environment Factor
 - PESTLE Technique
 - Porter's Five Force Model
- Internal Environment Factor
 - MOST Analysis Technique



Enterprise Architecture Frameworks

➤ Zachman Framework

is used for bigger organizations.

➤ POLDAT Framework

is used for smaller organization.



➤ TOGAF Framework is an open group architecture framework.

Solution Scope

A Business Analyst should model and define scope in a way that it provides enough details to address the business need and capabilities. This will help stakeholders to visualize the solution and understand how the solution will deliver the required capabilities.

In-Scope: These statements define what components, capabilities, interfaces, organizational units and processes are included in solution. If a solution is implemented in phases or iterations, the In-Scope statements should be described with respect to each phase or iteration.

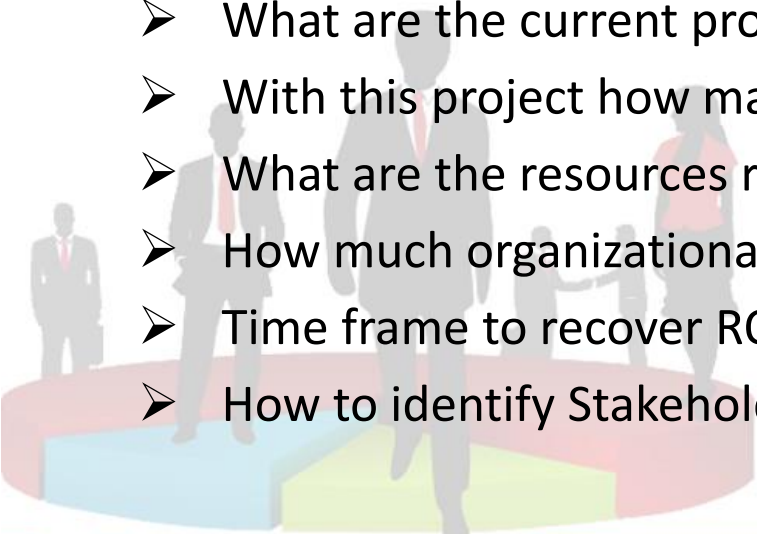
Out-Of-Scope: These Out-Of-Scope items can be identified by visualizing the bigger picture and taking off items which are somehow related to the solution but are not covered in the scope. These can also be potential requirements which can become a part of the scope in future.

Business Case

A business case is a package of information, analysis & recommendations.

A business case is prepared by Sr. BA, Business Architect & Pre-sales team.

A business case also helps in identify key stakeholders who are affected by the problem.

- 
- Why is this project initiated?
 - What are the current problems?
 - With this project how many problems could be solved?
 - What are the resources required?
 - How much organizational change is required to adopt this technology?
 - Time frame to recover ROI?
 - How to identify Stakeholders?

Handling Change Request

Initially BA should check the change request whether it is a defect from previous installations or not. If it is a defect then it is assigned to the technical team else

BA should be do the following

1. Impact Analysis
2. Feasibility Study
3. Effort Estimation



CCB – Change Control Board

Role of BA in Projects

Stages	Activities	Artifacts & Resources
Pre project	Enterprise Analysis – SWOT Analysis, GAP Analysis, Market Research, Feasibility Study, Root Cause Analysis, Decision Analysis, Strategy Analysis, Enterprise Architectural Frameworks, Project Scope and Business case writing, Risk analysis	Business Case SOW (Statement of Work) PO (Purchase Order) Sr. BA, Business Architects Pre sales Consultants
Planning & Estimations & Assessment	1. Understand Assumptions and Constraints along with Business Rules and Business Goals 2. Plan Packages for Big Projects 3. Understands the project plan from PM	
Project Kick Off (Big Picture Plan)	4. BA conducts stakeholders Analysis 5. Plan BA approach strategy (Req. gathering techniques, communication, Req. mgmt, Documents to follow, Tools to use, Change Request Handling methodology)for this Project	PM Sr. BA

Role of BA in Projects 2

Stages	Activities	Artifacts & Resources
Requirements Gathering	<ol style="list-style-type: none"> 1. Stakeholders identify and document 2. Client gives BRD or BA prepares BRD by interacting with Client – Brainstorming , Document Analysis, Reverse engineering, Interviews, workshops, Focus Groups, Observation, Questionnaires . 3. Prototyping can be used by BA to make the Client to give more specific requirements 4. Sort the gathered Requirements (avoiding duplicate Reqs , grouping into similar functionality or into modules) 5. Prioritize requirements – MoSCoW 6. Validate Requirements - FURPS 	BRD (Business Requirements Document)

Role of BA in Projects 3

Stages	Activities	Artifacts & Resources
Requirements Analysis	<ol style="list-style-type: none"> 1. Draws UML Diagrams (Usecase and Activity Diagrams) 2. Prepares Functional Requirements from Business Requirements 3. All Architects comes up with Technical Requirements (SSD) 4. SRS will have Functional Requirements and Technical Requirements 5. Takes Signoff on SRS from Client. SRS is the first legal binding Doc between the Business and the technical Team 6. BA prepared RTM from SRS before Design phase starts. (BA is the owner of RTM). 7. BA traces how requirements are dealt in each phase of development life cycle from Design till UAT 	Functional Requirements Specification SSD(Supplementary Support Document) SRS (Software Requirements Specification) RTM (Requirements Traceability Matrix)

Role of BA in Projects 4

Stages	Activities	Artifacts & Resources
Design	<ol style="list-style-type: none"> 1. From Usecase Diagram , Test Manager or BA will prepare Test Cases 2. Communicates with Client on the design and Solution documents (updates Status to Client and make them understand how the solution would look like to prepare them to drive UAT) 3. BA will initiate the preparation of End user manuals 4. updates RTM 5. From Use case Diagram Solution-Architect recommends Architecture of the IT solution 6. DB Architect uses Persistence Classes (Entity Classes) and comes up with ER Diagrams or DB Schema. 7. GUI Designer will look into Transient Classes (Boundary Classes) and designs all possible Screens for the IT Solution 	<p>Solution Document Design Document – HDD – ADD</p>

Role of BA in Projects 5

Stages	Activities	Artifacts & Resources
Coding	<ol style="list-style-type: none"> 1. BA organizes JAD Sessions 2. BA clarifies queries of Technical Team during Coding 3. Developers refer Diagrams and Transient (Controller Classes) of BA and code their unit 4. Update End user manuals 5. Update RTM 6. Conducts regular Status meetings with technical team and the Client and tuning Client for participation in UAT 	LDD – CDD Application



Role of BA in Projects 6

Stages	Activities	Artifacts & Resources
Testing	<ol style="list-style-type: none"> 1. BA- Prepares Test Cases from Use Cases or assists Test Manager to do so 2. BA performs high level testing 3. BA prepares Client for UAT 4. Test Data is requested by BA from Client 5. Updates End User Manuals 6. Updates RTM 7. Take signoff from Client on Client Project Acceptance form 	<p>Test Concerning Documents Application with less errors</p>



Role of BA in Projects 7

Stages	Activities	Artifacts & Resources
Deployment and Implementation	<ol style="list-style-type: none"> 1. Forwards RTM to Client or the PM which should be attached to the Project Closure Document 2. Coordinates to complete and share End User Manuals 3. Plans and Organizes Training Sessions for End Users 4. Prepares Lessons learned from this project (to take precautions for coming projects) 	



Your Questions



Interesting...?

Connect Help Desk



Toll Free :1800 208 7869

Email : helpdesk@coepd.com

