**Introduction**

Enterprise Application

* Effectively managing a business and will continue to provide increasing value to companies that understand their benefits.
* An enterprise app is an application used by a business to assist the organization with solving its problems.
* These apps are designed to integrate or interface with other enterprise apps, and they are known for their reliability, stability, and predictability
* An enterprise application (EA) is a large software system platform
* To operate in a corporate environment such as business or government.
* Complex, scalable, component-based, distributed and mission critical
* A group of programs with shared business applications and organizational modeling utilities designed for unparalleled functionalities.
* Developed using enterprise architecture.
* Critical component of any computer-based information system.
* Enhances efficiency and productivity through business level support functionality.

Examples

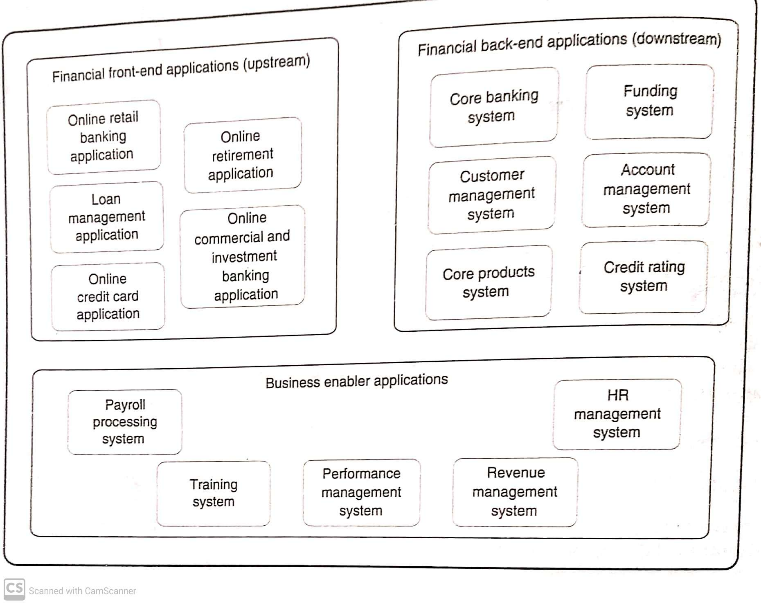
* Online shopping and payment processing, Interactive product catalogs, Computerized billing systems
* Security, content management, IT service management, Resource planning
* Business intelligence, Human resource management, Forms automation
* Sales force automation, Enterprise resource planning and business process management.
* IT compliance, Office productivity suites, Call Centers

What is an Enterprise Application

* A thin line of demarcation which separates whether an application can be termed as Enterprise Application or not.
* Typically an Application which satisfies the business functionalities (functional requirements) of an organization and also take care of factors such as efficiency, scalability, security etc (non functional requirements) is termed as Enterprise Application.
* An organization or enterprise may have several enterprise applications in its application landscape. They can be categorized from various perspectives.

Categorization of Enterprise Applications

Visibility to end user:

* + **Upstream:** Customers facing enterprise applications or front-end systems of an organization.
  + **Downstream:** Back-end enterprise applications working behind the scenes in an organization to fulfill the customers’ or end users’ needs.
  + **Business Enabler:** Applications which fulfill the general organizational needs.
* 
* Industry domain specific application
  + Billing – telecom – differ
  + Billing - retail
* **Type of processing supported:** Enterprise applications may fall under categories such as batch processing, online transaction processing (OLTP) or online analytical processing (OLAP) applications, or decision support systems (DSS).
* Custom built or readymade application.
* Host centric or Distributed

Building Enterprise Application

BEA

* Markets Flexible, unpredictable
* Vary Customer demands, Change in technology
* Diff themselves from their competitors

Challenges in Raising Enterprise Applications

With changing market dynamics, customer needs and technology, Enterprise applications face many challenges to stand out from the crowd. Below are the key challenges:

* Business Process Automation
  + Flow through – EA spread across the organization to achieve the business process simplification - Advantages
    - Quicker time to market
    - Improved productivity
    - Resource Optimization
* Data Harmonization (brining together)
  + Lack of single version of truth (mulitiple versions avail)
  + Unification and standardization of data difficult
* Application Integration
  + Different platforms
  + Different technologies
* Application Security
  + Access by users and other applications
  + Maintain the Enterprise brand value
* Internationalization
  + Across geographies- but serve localized manner
  + Localized stds – like Across time, date and currency
* Transaction Management
  + EA software – access by partners, suppliers, customers
  + Business level transactions – multiple system level transactions **– must be maintained by EA**
* Rich User Experience – End User Expect desktop like software over internet
* Quality of Service (QoS)
  + Scalability, Availability
  + Maintainability quality attributes (illities)
  + Reliability, Security
  + No QOS – lost productivity, revenue and goodwill
* Technology Selection
  + Platform, Framework, Tools
* Governance and Team Productivity
  + Right skills, effective reuse of existing software
  + Team diversity
  + Time-to-production-short

Software Engineering Methodologies

* Comprises of methodologies
  + to develop, manage and maintain the software with focus on
    - software quality
    - software requirements
    - software design
    - software development
    - Testing
    - configuration management

Software Engineering Methodologies

* **Waterfall methodology**
  + Traditional approach of software development that typically comprises of a sequence of phases
  + requirements, analysis, design, build and testing —
  + each phase output acts as input to the next phase (dis adv.)
  + **More time to production**
  + **Business value realized – end of s/w dev. phases\**
  + **Premature design – rework later**
  + **Req – may change – over a period of time – may be @ design time**
* **Iterative methodologies**
  + **Cyclic – build components in incremental manner**
  + **Choose subset req – to be implemented – each iteration**
  + **Development – learning from previous iteration - improve software component**
  + IBM RUP: RUP(Rational Unified Process) has assembled the iterations in four phases: inception, elaboration, construction and transition
  + Each iteration – unit of work – 9 discipline

6 Engineering Dicipline – business modelling, analysis and design, Implementation, test and development

3 – configuration and change management , project management and environment

Software Engineering Methodologies

* + Agile software development: an extension to the iterative approach to build applications in a nimble fashion with a light weight process.
  + Short feedback loop to accommodate changes quickly
  + Use 2 tech : code refactoring and test-driven

1. Code refactoring : to improve the quality of the design through code modifications based on hindsight.
2. A close involvement of the customer in the development process to provide continuous feedback

Eg. Extreme Programming, Scrum

Software Engineering Methodologies

**Extreme Programming (XP)**

* **Uses User stories for gathering requirements**
* **CRC cards for just in time design**
* **Pair Programming for coding**

**Scrum**

* **Shorter iterations with 30 day release cycles**
* **Daily standup meeting**
* **Concept of product backlog**
* **Hybrid approach – agile + RUP**

Life Cycle of Raising an Enterprise Application

EA – Life Cycle – **Incepting, Architecting and designing, constructing and testing**

Stages:

* Inception
  + Typically starts as a result of enterprise analysis and business modeling activities.
  + Requirements engineering is the key activity.
  + Req Analysis **– uses use cases, prototypes or user stories**
  + **To Validate- Req. Analysis – use the factors - budget, tech., business req. etc**
  + Concludes with casting the plan and project estimation.
* Architecting and Designing
  + Takes key inputs from the enterprise architecture initiatives of an organization.
  + Output:

- design patterns

- frameworks

- technology and tools

* + Designing
    - use several iteration to reach final one
* Construction
  + starts with building the application framework components
  + Code review – static and dynamic
  + Static : type checking, style checking, security review, software quality checking
  + Dynamic : code coverage, code profiling
  + Code coverage is an assessment of how much of your code has been run. This is used to see how well your tests have exercised your code.
  + Profiling is **used to see how various parts of your code perform**
* Testing
  + includes integration testing, system testing and user acceptance testing;
  + System testing :
    - Performance testing
    - Interface testing
    - Globalization testing
    - Compatibility testing
    - Usability Testing
    - Penetration testing
    - successful user acceptance test leads to application rollout.

Key Determinants of Successful EA

* Business case readiness
  + - Organizational objective, Vision and strategy, Support financial perceptive
    - EA derive well defined parameters from business case – used to measure its success
* Strategy to execute
  + Comprehensive plan
    - Manifest budget, Resources, Timeline, Availability of subject Experts
* Excellence in Execution
  + Time to completed , Stack holders management, Change management, Continuous Assessment

Skill Areas Requirements to Raise an Enterprise Application

Following are the key skill sets required to develop an enterprise application:

* **Knowledge of organizational dynamics:** understand the organizational business and business needs of end users.
  + Input to EA team - user interfaces req, Components – reused
* **Domain knowledge: Input to EA team : prescriptive business processes, information and data models**
  + comes handy especially during inception, architecting and design and testing phase of applications.
* **Business analysis skills:** collection of domain knowledge, technical knowledge, use of business analysis related tools and practice of soft skills.
* **Program management skills:** includes planning, estimation, budgeting, talent management, change management, positive communication and many more
* **Architecting and designing skills:** includes the knowledge of architecture views and view points, architectural patterns, design patterns, design paradigms like object orientation, aspect orientation and service orientation, usage of design tools, architectural and design best practices, technical frameworks, knowledge of modeling languages like Unified Modeling Language, etc.
* **Programming skills:** includes knowledge of a programming language ,knowledge of the underlying platform, knowledge of an Integrated Development Environment (IDE) tool, programming best practices, code review skills, knowledge of unit testing tools, configuration management and build tools, static code analysis tools and dynamic code analysis tools etc.
* **Testing skills:** includes skills for performing integration testing, performance testing, load testing, stress testing, application security testing, interface testing and user acceptance testing.
* Knowledge of tools

Measuring the Success of EA

* Once we are done with raising an enterprise application , we need to measure its success based on different parameters.
* Measuring the business process automation gain and the gain in terms of the ease with which end user can carry out the same process.
* Quality of application in terms of non functional requirements.
* Time to production
  + faster the time to market , better for the organization.
* Adherence to budget and timeliness.
* Cost effectiveness of application.
* Productivity of development teams.
* Effectiveness of the solution
  + Measuring the productivity gain acquired by the end users of EA in performing their job function
* Quality of Enterprise Application
  + Measure functional req., defect free code, use software metrics to measure the software code
* Time to Produce
  + Total time taken to roll out the enterprise application since its inception
* Cost Effectiveness of EA
  + Expenditure versus the benefits of the enterprise application
  + RoI(Return on Investment) is used to determine its cost effectiveness
* Budget and Schedule adherence
* Overruns of budgets and exceeding the timelines have a direct impact on the time to production and cost-effectiveness parameters.
* EA’s Program managements team has the bottom line responsibility for fulfilling this parameter
* Productivity
* Is a key measure from the perspective of development team
* Reuse, Use best practices, frameworks, tools,
* Automation of software processes

**Incepting Enterprise Applications**

**Introduction to Inception**

Incepting an enterprise application is the first life cycle phase of raising enterprise applications that primarily consists of the following activities:

* **Enterprise analysis**
* **Business modeling**
* **Requirements elicitation & analysis**
* **Requirements validation**
* **Planning and estimation**

**Enterprise Analysis**

Enterprise analysis is a holistic **analysis of business enterprise** that is typically done ***by business analysts***, and comprise of following key activities:

* Identification of business opportunities and business changes.
* Identification of various stakeholders across business units.
* Collection of business requirements and prioritizing them.
* Define business roadmap with scope and exclusions.
* Determines the high level investment needed for the enterprise
* Conduct feasibility study for any changes proposed
* Conduct risk analysis and competitive analysis.
* Decides on build-or-buy strategy
* Create RoIe business cases with proper justification
* Get the necessary approvals from the sponsors.

**Business Modeling**

* Enterprise analysis could lead to two forms of programs/projects:
  + Creating something new/**development project**
  + Extension**/change to something** which already exists/re-engineering project
  + **ReEngineering**
    - **Create new system using Java (old System in COBOL)**
      * **Get entire business process, Validations**
      * **Requirements derived from old system**
* Business modeling helps one to understand the business information and the **business processes** which an **enterprise uses** to fulfill **its business goals.**
* Business modeling is done by **business analysts**.
* To improve a business process, you first have to know its current state (As Is) and then you need to model the improved future state (To Be).
* To understand the business problem at hand it’s essential to have AS-IS and TO-BE modeling of the business processes
  + **AS-IS** business process model reflects the existing business process.
  + **TO-BE** business process model reflects the desired business process.

**How the As Is - To Be Model Works**

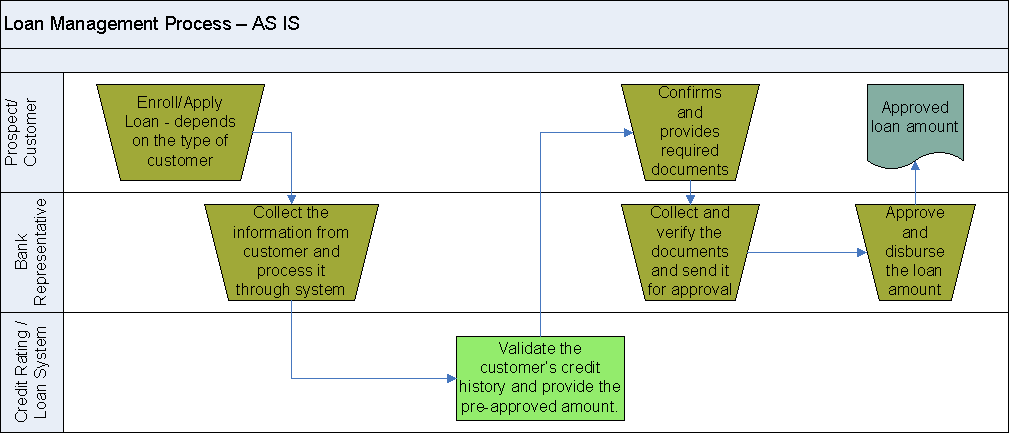
* **As-Is** process maps to:
  + Capture an accurate visual picture of your organisation's processes
  + Visualise process breakdowns and areas of risk
* Proposed **To Be**way of doing things, thereby:

Modelling - any future process changes before you make them

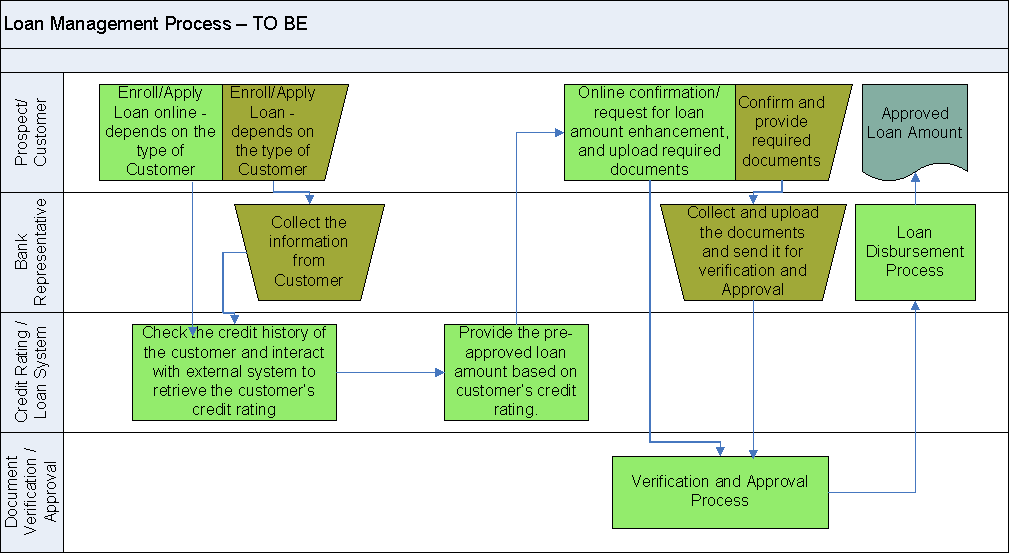
**Introduction to Case Study-EM Bank**

* Easy Money Bank**(EM Bank**) is a leading bank with its headquarters in Europe and offices and presence in another 50 countries.
* services offered by the EM bank are core banking , investment banking , mortgage and wealth management.
* Bank to expand the loan offerings and to reach out to more number of customers (both existing and new).
* Business Analyst has performed the task of business modeling and has come out with the AS-IS and TO-BE processes depicted in the two diagrams shown next.

**EM Bank- ‘AS-IS’ Process**



**EM Bank- ‘TO-BE’ Process**



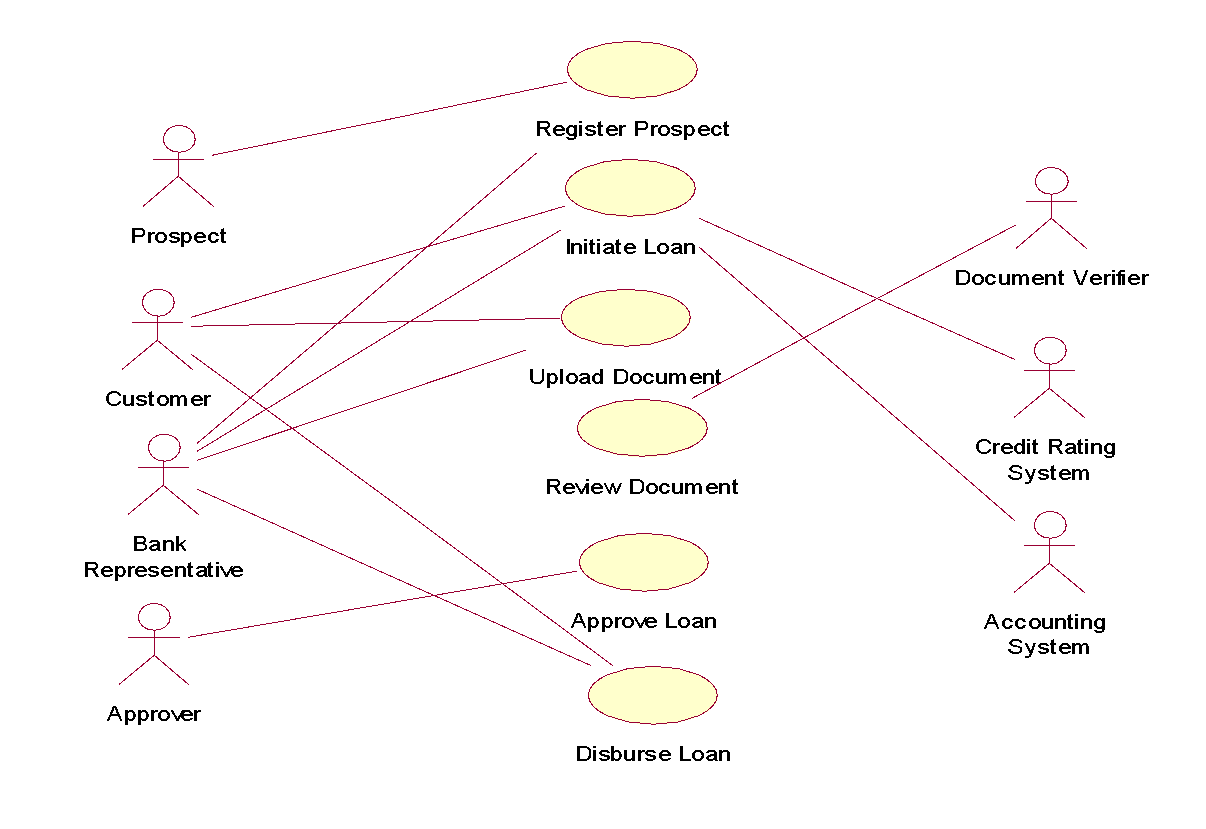
**Requirement Elicitation and Analysis**

* Requirements elicitation and analysis is a **systematic approach** of **capturing client requirements**, analyzing them and **documenting** the problem domain.
* There are various kinds of requirements which need to be elicited and are broadly divided into two categories:
  + **Functional Requirements:** capture what the system is expected to do - mechanisms such as **Use cases and prototypes** are used to depict functional requirements.
  + **Non Functional Requirements(NFR):** NFRs capture how the system does what it is expected to do with respect to its constraints and expected qualities of service (QoS) such as reliability, scalability, portability, usability, availability, security and performance.

**Functional Requirements**

* Use case diagram
  + Entities – Actors – represents a role
    - Human users, external systems, external devices
    - To find out Actors
      * Who is going to use the system?
      * Eg: Customer, Reviewer and Approver
      * Eg: Accounting system, Credit rating system
  + Functional reasons – Usecases
    - Why does an actor need to interact with the system under consideration?
    - Why does the credit rating system interact with the system?
  + Relationship
    - Generalization, Extends, Uses/include

**LoMS Use Case Diagram**

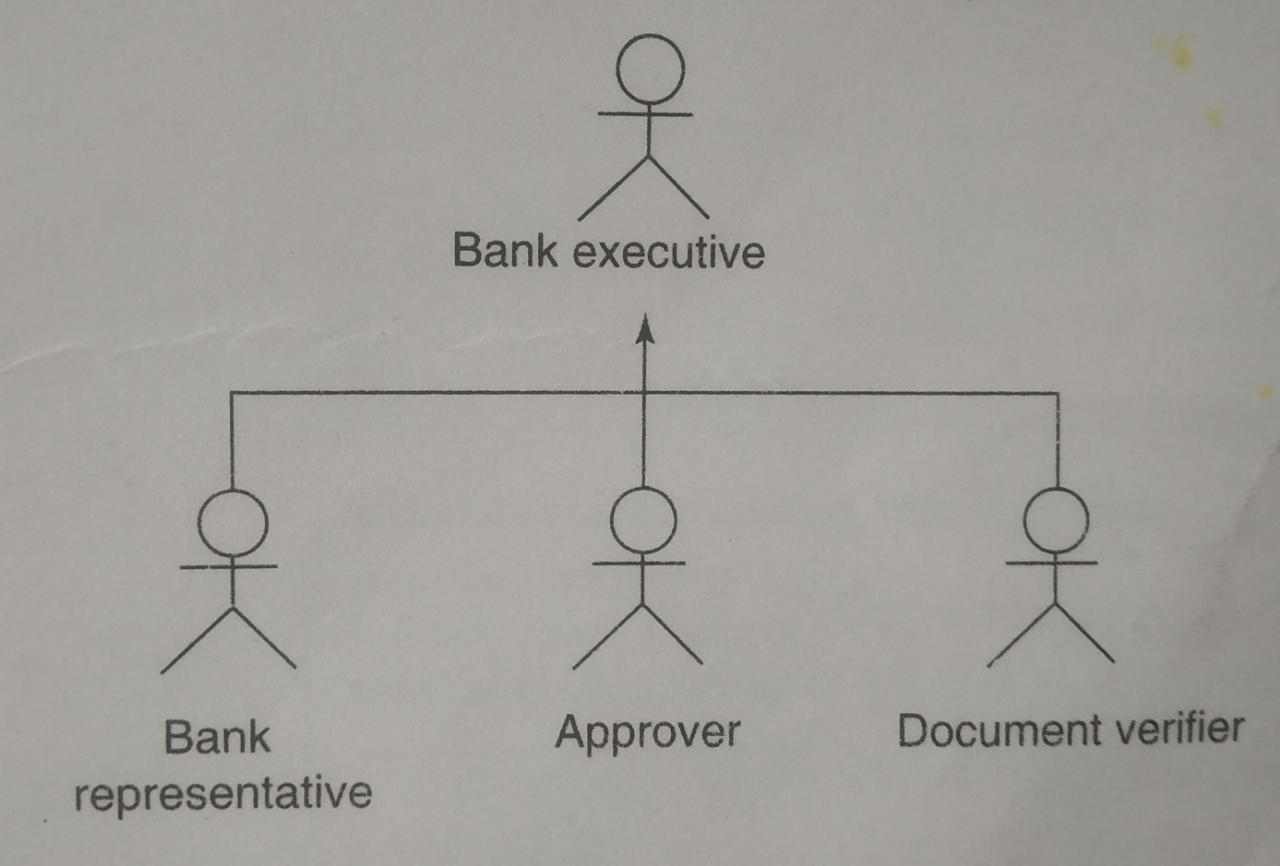


**Relationship**

* **Generalization relationship** among **Actors**

Bank Executive

Bank representative, Approver, Document Verifier

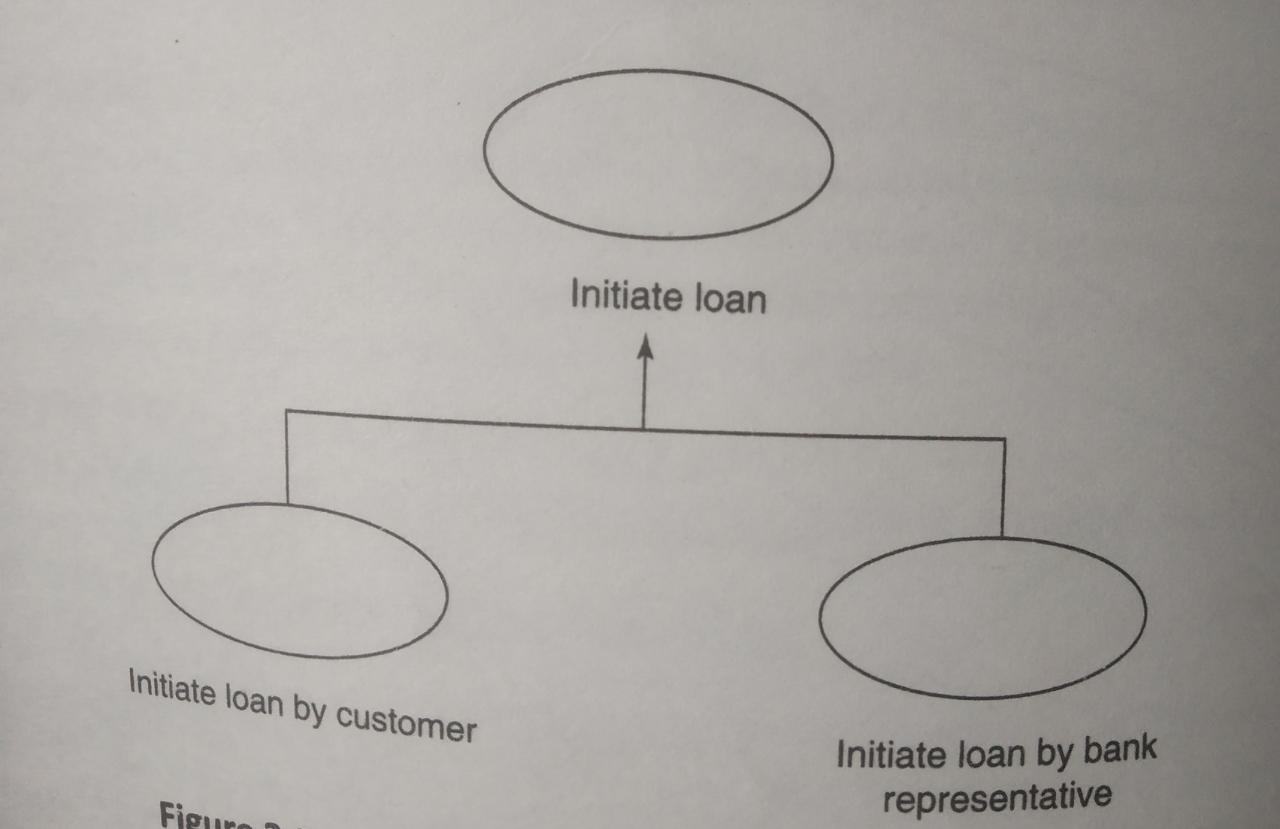


**Relationship**

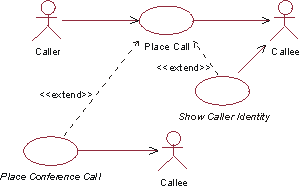
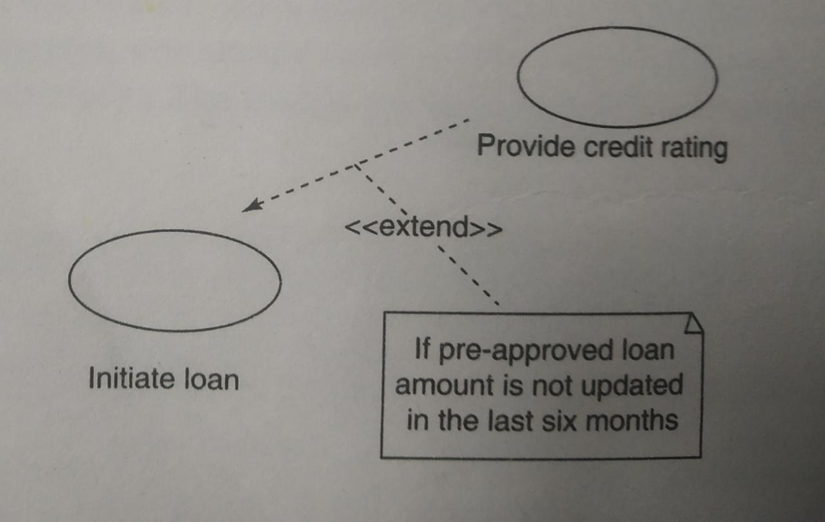
* Generalization relationship among Usecases

Initiate Loan

Initiate loan by customer, Initiate loan by bank representative

****

**Relationship**

* Extends relationship
* 
* Provide credit rating extends Initiate loan
* 

**Relationship**

* Uses/Include relationship
* An include relationship points from the CheckOrderStatus use case to the LogIn use case to indicate that the CheckOrderStatus use case always includes the behaviors in the LogIn use case.
* Textual Description of use case functionality is documented (**Use Case Specification)**

**Elements of Use Case Specification**

|  |  |
| --- | --- |
| **UCS Elements** | **Description** |
| Actors | List of participating actors in the use case |
| Description | Brief description of the use case |
| Preconditions | Prerequisites to start the use case |
| Post conditions | Outcome of the usecase |
| Priority | Business criticality |
| Trigger | The event that initiate the use case |
| Primary Scenario | Primary flow of use case |
| Alternate scenario number | List of alternate flows |
| Field Definitions | Size, type, values |
| Exceptions | Exceptional flows |
| Assumptions | Assumptions if any |
| User Interface | Reference to prototype |
| Related Use cases | List of related use cases |
| Non functional requirements | Capture NFR |

**User Prototypes**

* + The human actors identified need to interact with the system
  + This leads to creation of UI or user prototypes or wireframes
  + Purpose
    - To allow business users to validate the user interface
    - Look and feel perspective
    - To find out usability requirements like navigability
    - To understand the exceptions and variations in use cases

Points to take care with user prototypes:

* + Tools(Mspower point) used – for prototyping
  + Sample data used in prototyping(real data)
  + Protypes consider – primary and alternate scenarios
  + Use wireframe tools – convert to skeletal code

**Non Functional Requirements(NFR)**

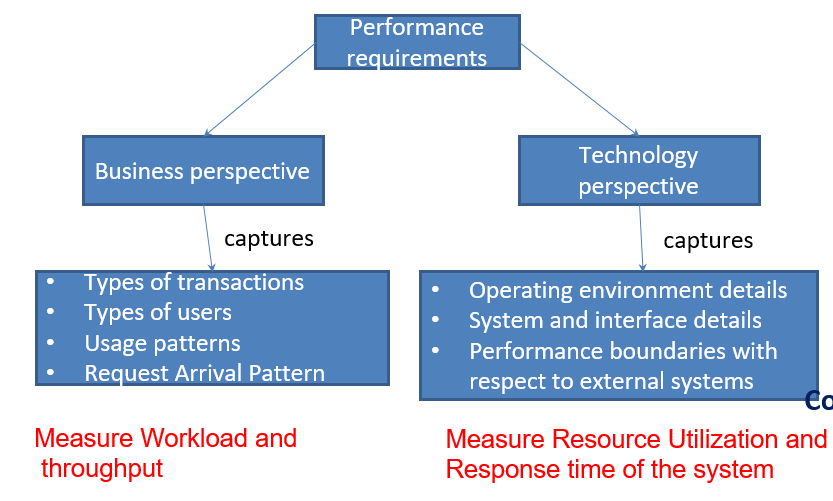
* NFRs play a pivotal role in the validation of the architecture and design of enterprise applications.

|  |  |
| --- | --- |
| **Key NFR** | **Description** |
| Performance requirements | Capture values such as peak load, throughput, response time |
| Usability requirements | Effective use of screen space, internationalization, easy navigability |
| Scalability Requirements | Data storage, planning for growth, resource utilization |

|  |  |
| --- | --- |
| **Key NFR** | **Description** |
| Interface requirements | Steps for integration of the application with other systems |
| Operating requirements | Security, Maintainability and Reliability requirements |
| Lifecycle requirements | Testing , portability, reusability |
| Regulatory requirements | Legal and other compliances to be followed |

**Non Functional Requirements (Contd…)**

* Performance Requirements are related to the measure of performance constraints.
* Workload of the system, server s/w, Hardware



**Non Functional Requirements (Contd…)**

* Usability Requirements dictate the user experience
  + - Thumb rules in smooth acceptance of EA
      * Logical Grouping
      * Easy to use and navigability
  + Auto complete of data
  + Interactive with application’s one preferred language
  + Support for popular browser
* Scalability requirements
  + Ability of the system to sustain its performance with growing number of users and data of the application
  + Example:
    - Ave. annual loan growth 20%, 1000/day 🡺 data processing and storage
    - Storage req. 1 gb / per day
    - Vertical Scaling =>resources are topped up on existing system
    - Horizontal Scaling => Separate phy. Sys. added
* Interface requirements
  + Interoperability and integration req. of EA
  + **Enterprise follow the mechanism for integration**
    - AT APPLICATION’S User Interface level
    - At business data level
    - At business process level
  + **To gather the interface related req.**
    - **Collect trigger events for commu. Between applications**
    - **Structure of data**
    - **Methods used to commu. Between appli.**
    - **Mech. To handle error**
    - **Interface is crossing the technology domain**
* Operating requirements are the *constraints* on the enterprise application which lay down the requirements related to security, maintainability and reliability of enterprise applications.
  + Two type of Security req.
    - Network Security req.
      * Related to network firewalls, routers and intrusion detection systems
    - Application security Req.
      * Authentication, Authorization and auditing
* Lifecycle requirements relate to needs that an **application** should **fulfill** at one or more **stages of its lifecycle**, and can be captured in terms of
  + testability req.: quality of an application
  + Reusability req.: check to develop with reusable components
  + portability req.: Check H/W and OS platforms – capable to deploy
  + installability req.: installation manual, installation-automated
* Regulatory requirements capture the restrictions and the legal requirements related to certain categories of sensitive data, and ways in which they can be processed.

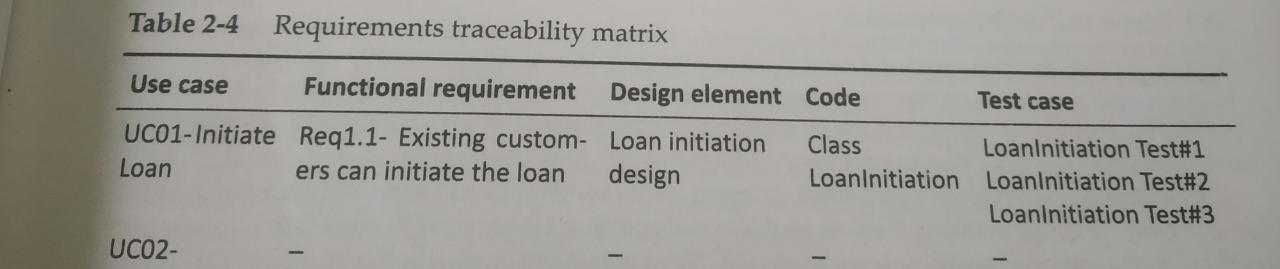
**Software Requirement Specification**

|  |  |
| --- | --- |
| **SRS Elements** | **Description** |
| Business Overview | Brief overview of what an enterprise does |
| System Overview | Overview of the system under consideration |
| Functionality list | List of business functionalities |
| Use cases | List of identified use cases |
| NFR | List of non functional requirements |
| Prototypes | Wireframes of the primary functionalities |

**Requirements Validation**

* Requirements validation is an exercise which is typically facilitated by business analysts to ensure that the requirements stated during **requirement elicitation and analysis** are **meeting the business objectives.**
* Requirements validation typically comprise of the following three activities:
  + Ensuring the coverage of all business needs identified during enterprise analysis and requirements elicitation phase.
  + Ensuring the requirements documentation sanctity by subject matter experts and end users.
  + Ensuring the feasibility of requirements to the extent possible.
* There are many ways to perform requirement validation such as requirement traceability matrix (RTM), user acceptance test (UAT) cases and proof of concepts(PoC).

**Requirements Validation**

* Use Req. Traceability Matrix: validated from the viewpoint of completeness – target solutions addresses all the business needs
* 

**Planning and Estimation**

* The business case acts as the primary input for preparing the highest level planning document called Project Charter.
* Prepared by project manager, business analyst and architects
* Once the Project Charter is realized, detailed planning is done followed by the process of “estimation”, taking care of cost, schedule, size and effort required to build the enterprise application.
* A few of the popular estimation techniques are :
  + **Ballpark Estimation:** involves decisions based on heuristics. Data available in the organization from similar projects, similar domains or industry standards are used for estimation.
  + **Use Case Point Estimation:** The target functionality which is depicted in the form of Use case models is the basis for doing Use Case Point estimation.
    - It is popular
  + **Function Point Estimation:** A function point or FP is the unit to measure the functional size of the software under measurement. To perform FP estimation, a fully documented functional specification of the target software solution is needed.