Information Security Management System

Secure Software Development Life Cycle Policy

Document No. - JMBGRP/ISMS/SDLC

Version_v1.0

THIS DOCUMENT IS PRIVATE PROPERTY OF J.M BAXI GROUP (JMBGRP) MUST NOT BE REPRODUCED, DISCLOSED TO ANY THIRD PARTY, OR USED FOR ANY OTHER PURPOSE WITHOUT PERMISSION IN WRITTEN FROM J.M BAXI GROUP

Document Details

Classification	Internal	
Released date	28.08.2018	
Description	The documented policy and methodology for addressing and assessing the risks to JMBGRP's Information and processing facility.	
Custodian	Corporate IT dept.	
Approved by	Manish Jaiswal (Group CTO)	
Owner	Corporate IT Dept.	

Distribution List

Name
To JMBGRP Employees Only
Third Party and Auditors: On Need basis

Version History

Version no.	**Version date**	**Approved by**	**Comments**	**Signature**
v1.0	28.08.2018	Manish Jaiswal (Group CTO)	Reviewed & no changes	
v1.0	10.01.2019	Manish Jaiswal (Group CTO)	Reviewed & no changes	
v1.0	08.02.2020	Manish Jaiswal (Group CTO)	Reviewed & no changes	
v1.0	11.02.2021	Manish Jaiswal (Group CTO)	Reviewed & no changes	
v1.1	25.03.2022	Manish Jaiswal (Group CTO)	Logo changes done in this policy	
v1.2	18.08.2023	Manish Jaiswal (Group CTO)	Document reviewed. Modification done to "11. Reference to ISO 27001:2022"	
v1.3	16.10.2024	Manish Jaiswal (Group CTO)	Modifications done in Section: "6,7"	

Contents

1. Introduction

1.1. This document serves as the Secure Software Development Life Cycle (SDLC) Policy for JMBGRP. The policy outlines the framework for developing and maintaining secure software applications within the organization, aligning with the requirements and best practices established by the ISO/IEC 27001:2022 standard. The purpose of this policy is to mitigate security risks associated with software development and protect sensitive information assets. It provides guidelines and principles to ensure that all software development activities adhere to rigorous security standards throughout the entire software development life cycle.

2. Purpose and Scope.

2.1. The purpose of this Secure Software Development Life Cycle (SDLC) Policy is to establish a framework for developing and maintaining secure software applications within the organization. This policy aims to ensure that all software development activities are conducted in accordance with the ISO/IEC 27001:2022 standard requirements and best practices to mitigate security risks and protect sensitive information.

2.2. This policy applies to all software development projects and activities undertaken by the organization, including in-house development, outsourced development, and third-party software procurement. It covers the entire software development life cycle from initial planning to post-implementation maintenance and support

3. Normative Reference

ISO 27001:2022-

4. Terms and Definition

For this document, terms and definition are defined in Annexure 1.

5. Policy Statement

Required level of security shall exist to ensure the availability, integrity, and confidentiality of all software and the associated data. It shall be ensured that access to the software is authorized and, on a need, to know development and changes made to these are adequately controlled and audit trails are provided to log and detect any unauthorized activities. The Software Development life cycle provides direction for ensuring that:

5.1. Management and authorized personnel are responsible for following the defined and structured framework outlined in the SDLC policy to facilitate effective software development and testing.

5.2. Adherence to this policy is mandatory for all software development activities within the organization, including in-house development, outsourced development, and third-party software procurement.

5.3. The authorized personal is responsible for the SDLC process including planning, requirements gathering, design, coding, testing, deployment, and maintenance.

5.4. Security requirements for software development or modification to existing software at JMBGRP are defined, agreed, and documented as part of the overall business case for the system.

5.5. Acquisition, development, and deployment of software applications that are used in JMBGRP addresses confidentiality, integrity, and availability of the information.

5.6. Appropriate security is maintained in the development and support processes by controlling the development, live and support/test environment. Software code/source code shall be in a library with access to authorized personnel with a check-in and check-out process.

5.7. Checks shall be applied to ensure data input is validated and correctness is maintained. Similarly checks for validation of data output shall be present. Also, session inactivity time-outs and connection time outs shall be implemented to prevent unauthorized access.

5.8. Formal process shall be instituted for making changes to the packaged software after obtaining consent from the software vendor and after careful analysis of the business requirement and possible impact on the existing environment after modifications are live.

5.9. The test and production environment shall be separate along with separate VLAN for each of them.

5.10. Process shall be implemented to monitor, review, and address the known technical vulnerabilities in applications by either taking support for external agencies or performing the same through the identified group/ISG within the JMBGRP.

5.11. Any software copyright violation by employees shall lead to a disciplinary action, even leading to a legal action.

5.12. JMBGRP information technology infrastructure and applications shall only be licensed software's deployed on them for the development of internal software.

5.13. Suitable technology solutions shall be deployed which shall manage application monitoring to facilitate in better management of resources and track incidents arising out of software use

5.14. Follow secure design principles and best practices to design software that is resistant to common security vulnerabilities such as SQL injection, cross-site scripting (XSS), and buffer overflows. Use secure coding guidelines and standards to ensure that code is written securely and is resistant to attacks.

6. Procedure

The procedure covers the following areas:

6.1. Ownership / Responsibility of Software.

6.1.1. Each software shall have a designated owner who shall be responsible for the integrity, availability, and confidentiality of the application and the associated data. The Software / Information Owner shall, at a minimum, be responsible for the following:

6.1.1.1. Ensuring the security of the software and the associated data on advice from the Security In-charge in IT TEAM.

6.1.1.2. Prioritizing any changes to be made to the software and authorizing the changes.

6.1.1.3. Ensuring that adequate controls are built into the software by being actively involved in the Software design, development, and testing and change process.

6.1.1.4. Ensuring that the security of the software has been reviewed and getting sign-off from the identified Software Owner in IT TEAM before implementing in production. The security risk assessment shall be conducted before the UAT process.

6.1.1.5. Ensuring that the Software meets the functional needs of the users.

6.1.1.6. Taking decisions on any new software to be acquired / developed or any old software to be discarded. Keeping the ISEG team informed on the decision to purchase software and assessing the software based on the security policy requirements. All deviations to be signed by the Software Team Group Leaders before purchase.

6.1.1.7. Deciding on the archival/purging procedures for the data maintained by the software.

6.1.1.8. Working with business teams to classify the data according to the criteria.

6.1.1.9. Ensuring that Change Management process is followed for any changes to be done after assigning production status to the software and making the forms available for audit purposes.

6.1.1.10. Once a risk assessment is conducted the Software Owner shall report to the Application Owner of any deviations and take sign off on exceptions.

6.1.1.11. Ensuring that the software is not ported to any other servers from the original server unless the DC (Data Center) Manager has been informed to update the DC System Inventory.

6.1.1.12. Ensuring that the new software being purchased /developed meet the Information Security policies and procedures

6.1.1.13. Prepare Standard Operating Procedures (SOP) for the software users to minimize chances of software operating issues. This shall include software functioning process/procedures (which shall be included in the software user manuals) or administrative functions like daily activities, backup / restore process, user ID creation /modification / deletion processes.

6.1.1.14. Working out procedures with concerned department to monitor & review logs (administrator and operator) for any discrepancies.

6.1.1.15. Ensuring that the development, test, and production environment shall be properly segregated. Developers shall not have any production access unless explicitly approved (DGM and above).

6.2. Requirement Analysis:

6.2.1. The Department Head shall submit a business requirement to develop/change a system which shall include details such as:

6.2.1.1. User's Name

6.2.1.2. Department

6.2.1.3. Name of the program module to be developed/changed.

6.2.1.4. Requirement specifications.

6.2.1.5. Reasons for the changes, benefits, etc.

6.2.1.6. Security requirements such as:

6.2.1.6.1. Business value of the information asset involved

6.2.1.6.2. Potential business damage, which may result due to absence or failure of the security controls.

6.2.1.6.3. Framework of controls to fulfill the security requirements.

6.2.1.6.4. Role definitions.

6.2.1.6.5. Audit trails

6.2.2. The Software/Information/Process Owner shall ensure that the final product meets the business requirement. The following security requirements shall be considered wherever feasible:

6.2.2.1. Scalability of the system

6.2.2.2. Stability of the system.

6.2.2.3. Testing development environment shall be separate from the live environment.

6.2.2.4. Live critical and confidential data shall not be shared with the developers for testing purposes.

6.2.3. The Application Owner shall approve upgrade of software. Application owner / shall consider following points when upgrading software:

6.2.3.1. Software licenses are reviewed for validity, and compatibility with old system.

6.2.3.2. Vulnerability testing and regression testing are done before the upgrade

6.2.4. Function statement requirement:

6.2.4.1. The business owner will collaborate closely with the application developer to articulate comprehensive functional requirements.

6.2.4.2. Following detailed discussions, the application team member will translate these requirements into clear process flow diagrams, presenting the structure to the business owner for approval. Upon satisfaction, the business owner will provide sign-off, authorizing the application development team to proceed with development.

6.3. Planning and Designing:

6.3.1. Software Team Work flow and Responsibility.

6.3.1.1. Software is tested before implementation. In addition to Unit testing, integration testing, load testing, penetration testing shall be carried out wherever possible.

6.3.1.2. Software is tested for any covert channels, trap doors and Trojan code.

6.3.1.3. No debugging information is included in the compiled program.

6.3.1.4. Implementation of change shall be carried out to the live environment by the software administrator, only after successful user acceptance testing. The software administrator shall ensure that:

6.3.1.4.1. An approval from Head Software Development Team and the relevant Business Head / Process owner shall be taken for implementing the changes to the live environment.

6.3.1.4.2. Expected period of disruption (if any) in service shall be communicated to the users in advance.

6.3.1.4.3. A complete backup of existing data and source code is taken before implementing the changes to the live environment.

6.3.1.4.4. All the changes shall be documented as per the Change Request so as to facilitate a smooth roll back when required.

6.3.1.4.5. The Department Head / Process Owner / Information Owner shall give a sign off after implementing the changes.

6.3.2. Model Decision:

6.3.2.1. Head Software Developer Team shall Gather and document the specific requirements and objectives of the software project, including functional, technical, and business requirements.

6.3.2.2. Research should conduct to identify and evaluate various software development models suitable for the project, such as Waterfall, Agile, Scrum, Kanban, or a hybrid approach for both Software and Application by the authorized person

6.3.2.3. Evaluate the nature of the project, including its complexity, size, scope, and timeline, to determine the most suitable software/application development model that aligns with the project's needs and constraints.

6.3.2.4. Assess the skills, expertise, and experience of the software development team members to identify the model that best leverages their strengths and supports effective collaboration and communication

6.3.2.4. Consult with project stakeholders, including business owners, end-users, and project sponsors, to understand their preferences, expectations, and priorities regarding the software development process and outcomes.

6.3.2.5. Identify potential risks, challenges, and constraints associated with each software development model, such as resource availability, budget limitations, technology dependencies, and regulatory compliance requirements.

6.3.2.6. Perform a cost-benefit analysis to compare the advantages and disadvantages of each software development model in terms of project cost, time-to-market, flexibility, adaptability, and overall project success.

6.3.2.7. Based on the gathered information, analysis, and stakeholder input, authored personal should make an informed decision to select the most appropriate software development model that best fits the project's requirements, constraints, and objectives.

6.3.2.8. Document the rationale behind the chosen software development model, including key factors considered, decision criteria, and the expected benefits and outcomes of adopting the selected model.

6.3.2.9. Communicate the selected software development model to all project stakeholders, ensuring alignment and understanding of the chosen approach, roles, responsibilities, and expectations throughout the project lifecycle.

6.3.2. Detailed interface Design:

6.3.2.1. Head Developer Team should collect and document detailed requirements for the interface design, including user interface (UI) elements, functionality, navigation, user workflows, and integration points with other systems or components and review the overall system architecture and technical specifications to understand how the interface design will integrate with other system components and external dependencies.

6.3.2.3. Define the desired user experience (UX) by considering factors such as user personas, user journeys, usability principles, accessibility requirements, and design best practices.

6.3.2.4. Design data models and interaction flows that outline how data will be presented, collected, processed, and stored within the interface. Define data validation rules, error handling mechanisms, and feedback mechanisms.

6.3.2.5. Incorporate branding elements, visual styling, color schemes, typography, and imagery consistent with the organization's brand guidelines and design standards to ensure a cohesive and branded interface.

6.3.2.6. Ensure that the interface design is responsive and compatible with various devices, screen sizes, resolutions, and platforms, including desktops, tablets, smartphones, and web browsers.

6.3.2.7. Conduct design reviews and usability testing sessions to gather feedback from stakeholders, end-users, and usability experts. Validate the interface design against usability principles, accessibility standards, and user requirements

6.3.2.8. Configure separate development, test, and production environments to

support the interface design implementation and testing phases. Ensure that each environment is appropriately provisioned, secured, and isolated from production data.

6.3.2.9. Document detailed design specifications, including UI layouts, component specifications, interaction behaviors, data schemas, API endpoints, and integration requirements, to guide development and implementation and collaborate with developers to translate the detailed interface design specifications into functional code and implement the interface design components within the designated development environment.

6.3.3. Frontend-customer facing Application:

6.3.3.1. Encryption for clients accessing front-end with certificates from appropriate signing authorities shall be provided**.**

6.3.3.2. Shall be on https. Two or Multifactor authentication for user access (physical token or OTP) shall be deployed**.**

6.3.3.3. Software / Application audit shall be conducted to check for the following vulnerabilities as listed below. Audit shall be conducted every 12 months or every time a change occurs within the content of the website whichever occurs first.

6.3.3.3.1. Cross Site Scripting (XSS)

6.3.3.3.2. SQL Injection

6.3.3.3.3. Cookie injection and manipulation

6.3.3.4. Non business oriented administrative ports shall be closed (ftp, telnet, Terminal Services).

6.3.3.5. IDS/IPS/HIDS shall be installed on system to detect and prevent possible attacks.

6.3.3.6. User activity shall be monitored on a periodic basis and logs shall be reviewed for detection of any fraudulent activity or assist any detection of a new vulnerability.

6.3.4. Backend:

6.3.4.1. Strict Source routing shall be deployed if Front-end and backend are on different servers.

6.3.4.2. Backend Databases shall be encrypted be encrypted.

6.3.4.3. Communication between front end and backend shall be encrypted.

6.3.4.4. Non business oriented administrative ports shall be closed (ftp, telnet, Terminal Services).

6.3.4.5. IDS/IPS/HIDS shall be installed on system to detect and prevent possible attacks.

6.3.4.6. Access logs shall be monitored and reviewed.

6.3.4.7. Remote access to databases shall be prohibited to the concerned support personnel shall go on site to rectify the problem.

6.3.5. Implementation of Developed/Procured Application Software:

6.3.5.1. During the cut over period during implementation of developed / procured new software application (old system has been stopped and new system has not gone on live) it is the responsibility of the Process Owner/Department Head.

6.3.5.2. To provide accurate information/data for uploading into new system as per the necessary format.

6.3.5.3. Validation of accuracy and completeness of uploaded data/ information is the responsibility of the Process/ Application/Information owner/Department Head and sign off shall be given on the correctness and completeness of the data/information.

6.3.6. The Design also ensure that:

6.3.6.1. Appropriate logs (administrator) and audit trails are enabled. Every application affecting sensitive, critical, or valuable information shall provide for logging the transaction id, date, time, originator id and authorizer id. Additionally, the Information Owner shall also evaluate the need for maintaining audit trails for transactions of a specific nature, value, or any other applicable parameter. Applications shall also provide

for logging unsuccessful logon attempts and access to sensitive options in the application, e.g., master record changes, granting of access rights, etc. Arrangements shall be made for proper storage of these logs (and other audit data) in a manner that they are retrievable for a period fixed by the application owner.

6.3.6.2. Login credentials are not hard coded within the program code.

6.3.6.3. Server parameters (e.g., Network address or name) are not hard coded within the program code.

6.3.6.4. A user account is locked after specified unsuccessful log-on attempts.

6.3.6.5. Exception and error messages do not contain system related information. A generic error message is displayed to the Application users.

6.3.6.6 Segregation of duties: As a part of a good internal control system as well as to minimize the risk of negligent or deliberate system misuse, every application shall provide for adequate segregation of duties. It is the responsibility of the Information / Application Owner and/or the Department Head, to ensure adequate segregation of duties. Critical areas where segregation of duties is required include updation of master files, parameter files, interest rates, entry and authorization of transactions, system administration, security audit, as well as systems development and maintenance.

6.3.6.7. Employees shall not have access to the database prompt of the applications. Access to the database prompt shall be restricted only to the database administrator to generate reports using SQL statements. Super-user / Privilege rights shall be given to the database administrator. Administrator Rights shall only be available with 2 personnel in the Application Team.

6.4. Implementation:

6.4.1. Source Code:

6.4.1.1. The developer shall ensure that stored procedures, modules, including files and functions shall be referenced correctly and the source code shall

address the GUI guidelines

6.4.1.2. The source code so developed shall be compatible with the hosted hardware and software environment. Source code shall have the ability to handle exceptional conditions. The business logic shall have its own storage location.

6.4.1.3. The source code shall demonstrate a comprehensive coverage of information, ensuring clarity and completeness in its structure and implementation. The source code is devoid of vague sections or insufficient detail, thereby enhancing its usability and maintainability. Additionally, the code shall appropriately comment, facilitating easy understanding for the target audience and enabling seamless comprehension of its functionality and logic.

6.4.1.4. IT TEAM / Manager – Software/Applications shall ensure that Program source code and the executable code are maintained in separate libraries. The access to source code library server shall have documented and restricted access. User activity shall be logged. Only Head Application Development Team shall have administrative access and rest of the users shall have access as per their work profiles. The collection of all source code these libraries shall be known as software library.

6.4.1.5. For all application software that is used for carrying out critical business functions in the JMBGRP, the source code shall be received from the vendor. Alternatively, the JMBGRP shall have a software vendor agreement with a third party to ensure source code availability if the vendor goes out of business. The J M Baxi Group shall ensure that product updates and program fixes are also included in the vendor agreement

6.4.2. Program control area:

6.4.2.1. A separate program control area shall be created in the software library

6.4.2.2. Program source code under development shall be stored in the program control area. The IT Application Help desk shall not have an unrestricted access to the program control area.

6.4.2.3. The Program Source code under development shall be separated from the program source code in the live environment.

6.4.2.4. Program source code in the live environment shall not be available for modification. It shall be carefully archived and access to it shall be restricted.

6.4.2.5. Access to the source code library server shall be restricted and audit logs shall be maintained.

6.4.2.6. All developers shall be on a separate VLAN.

6.4.2.7. All developers shall not have access to portable media or internet.

6.4.3. Check OUT:

6.4.3.1. Programmers who want to make changes to the program source code shall fill up a change request form and send the form to the IT TEAM /Manager - Applications. (Application / Information /Process owner along with the Manager–Applications & Programmers shall document the changes in details)

6.4.3.2. The IT TEAM/ Manager – Applications after approval shall grant access to the required files

6.4.3.3. The program source code to be changed shall be removed from the program control area i.e. it shall be checked out and handed over to the programmer to make changes.

6.4.4. Check IN:

6.4.4.1. The programmer shall make the changes that are requested in the change request form and shall submit the program back to the Process / Application / Information Owner/ IT TEAM/Manager – Applications

6.4.4.2. The Process/Application/ Information owner shall verify the changes as against the following:

6.4.4.2.1. Only authorized changes are made to the program

6.4.4.2.2. The programmer has performed an acceptance testing after taking the change.

6.4.4.2.3. The IT TEAM/Manager - Applications shall put the changed

program source code in the program control environment i.e., check in.

6.4.4.3. Before checking in the program source code, the IT TEAM/Manager - Applications shall ensure that:

6.4.4.3.1. A proper version is assigned to the program

6.4.4.3.2. The configuration of the program is appropriately updated

6.4.4.3.2. All the configuration items shall have a common naming convention to identify the version numbers.

6.4.5. Access of Source code:

6.4.5.1. Access to source code repositories shall be granted only to authorized personnel who require such access to perform their job responsibilities.

6.4.5.2. Access rights to source code repositories shall be based on the principle of least privilege, ensuring that individuals have access only to the source code necessary for their roles.

6.4.5.3. Access to sensitive or proprietary source code shall be restricted to individuals with a legitimate business need, as determined by their job function and project requirements.

6.4.6. Outsource Development:

6.4.6.1. Contracts with third-party vendors shall include provisions related to the protection of intellectual property, confidentiality of source code, compliance with security standards, and access control measures.

6.4.6.2. All source code developed or customized by third-party vendors under outsourcing agreements shall be the exclusive property of the organization

6.4.6.3. Contracts with third-party vendors shall clearly define the organization's ownership rights and intellectual property rights over the source code, including provisions for source code escrow arrangements to safeguard against vendor bankruptcy or non-performance.

6.4.6.4. Access to source code repositories containing outsourced code shall be restricted to authorized personnel with a legitimate business need, such

as project managers, development team leads, and designated reviewers**.**

6.4.6.5. Third-party vendors shall adhere to strict confidentiality agreements and non-disclosure obligations regarding the organization's proprietary source code and intellectual property. Access to source code repositories shall be granted only to individuals who have signed appropriate confidentiality agreements

6.4.6.6. During development of applications in-house and those which have been developed by external parties, source code review shall be done by the identified IT Team from the JMBGRP to rule out any possibility of covert channels like Trojans, Back Doors does not exist.

6.4.6.7. Third-party vendors shall be required to adhere to the organization's information security policies, procedures, and standards, including requirements related to data protection, encryption, access control, and vulnerability management

6.4.6.7. Positive assurance report shall be provided by the vendor or the internal group head which is responsible for application development. Secure Source Code review shall be done at least one every year on new and existing applications.

6.4.6.8. Application vulnerability assessment and penetration testing – Black and Grey Box shall be conducted at least once in six months or when significant changes have been done to the application or when new vulnerabilities have been reported.

6.4.7. Secure Coding:

6.4.7.1. All software development activities shall adhere to industry-recognized secure coding standards.

6.4.7.2. Secure coding guidelines shall be integrated into the organization's development methodologies and practices to ensure consistent application across all projects.

6.4.7.3. Input Validation and Sanitization:

6.4.7.3.1 All transactions to be input to an application including user interface input, command line parameters, environment variables shall first be subjected to adequate reasonableness checks, edit checks, and/or validation checks. Transactions, which fail such checks, shall be rejected with a notification sent to the submitter and logged to an error file for review later.

6.4.7.3.2. It shall be ensured that input data validations are incorporated into the system to ensure correctness of data entered. These may include validations for:

a. Out of range value

b. Invalid characters in data fields

c. Missing or incomplete data.

d. Exceeding upper and lower data volume limits.

e. Unauthorized or inconsistent control data.

6.4.7.3.3. Input Validation Procedures for Rejected or Suspended Input: Input transactions, which are corrected for resubmission, or which were suspended and are now approved for resubmission, shall be subjected to the same validation procedures at the application level as original input transactions. The rejected transactions in the error log shall be deleted after these have been resubmitted and accepted.

6.4.7.3.4. It shall be ensured that output data validations are incorporated into the system to ensure that the information processing is correct, and the output is in the desired format. These may include validations for:

a. Reconciliation of data file after transaction updates.

b. Checks for checking integrity of data downloaded

c. Hash totals of records and files if necessary.

d. Checks to ensure that programs are run in the correct

order nd at correct time.

e. here are two things that shall always be done with data returned from a called function or method. First, always check the return codes of the call and take appropriate action. Second, always sanitize data returned from components.

f. Temp files: Temp files shall be created in a directory that is readable and writable only by the application program

6.4.7.4. Authentication and Authorization:

6.4.7.4.1. Strong authentication mechanisms, such as multi-factor authentication (MFA) or password hashing, shall be implemented to verify the identity of users accessing the application.

6.4.7.4.2. Role-based access control (RBAC) shall be enforced to restrict access to sensitive functionality and data based on users' roles and privileges.

6.4.7.4.3. Applications shall not allow unauthorized entries to be updated in the database. Similarly, applications shall not allow any modifications to be made after an entry is authorized. Any subsequent changes shall be made only by reversing the original authorized entry and passing a fresh entry

6.4.7.4.4. Shall be under appropriate controls. Users shall exercise such control if application cannot be configured to prohibit batch authorization. Batch transactions shall be reviewed on case-to-case basis.

6.4.7.4.5. Development teams are responsible for implementing secure coding practices in accordance with this policy and applicable secure coding standards.

6.4.7.4.6. QA teams are responsible for conducting security testing, code reviews, and vulnerability assessments to identify and remediate security flaws in the application code

6.4.7.4.7. Applications/Software shall be configured to logout the users after 5 minutes of inactivity. The application shall ensure roll over of incomplete transactions and otherwise integrity of data in case of logout

6.4.7.4.8. Inactive users of an application for over 60 days shall be disabled.

6.5. Database Management

6.5.1. Database Security Management:

6.5.1.1. Operating System/ Space Management: Application / Information Owner with Database Administrator shall ensure that:

6.5.1.2. Each database shall be installed in a separate operating system directory.

6.5.1.3. Database shall not start automatically with system start-up. The ability to start a database shall be restricted with authorized personnel only.

6.5.1.4. The block size parameter for an Oracle/Sybase/SQL database shall be the same as the operating system block size

6.5.1.5. Free space in each table-space shall be regularly monitored and new table space be added after considering the requirements in consultation with the Software/ Application Owner.

6.5.1.6. Archived redo log's disk space shall be continuously monitored as if the archived redo log disk fills up, the database is automatically stopped.

6.5.1.7. On critical databases redundancy shall be built on disk subsystems (RAID) to factor recovery from system, file, or component failure

6.5.2. Database access:

6.5.2.1. Application/information Owner with Database Administrator shall ensure that Operating System level file and directory permissions are as restrictive as possible and shall be set considering application requirements. Ideally permissions for data, logs and control directories shall be set as:

6.5.2.1.1 Read, Write to Owner

6.5.2.1.2. Read to Group (Oracle / Sybase / SQL DBA group)

6.5.2.1.3. Users can access database and related files from within application only. E.g. at Oracle /SQL/UNIX installations, access to these directories shall be restricted to concerned administrators. Generic IDs shall not be used to access back end.

6.5.2.1.4. Passwords for default users (sys, system, oradsys, mdsys, SA etc.) shall be changed. All demo users and demo (sample) databases shall be removed.

6.5.2.1.5. Default table space for users shall not be 'System Table space'.

6.5.2.1.6. The system shall prompt the user for database password whether the access is through command line or through a program

6.5.2.1.7. Appropriate controls on resources e.g.: sessions per user & other controls defined in hardening document shall be implemented.

6.5.2.1.8. Access to database tables through links shall be minimized as they complicate database management and Creation of users shall be on need basis with as restrictive permissions granted as possible. Granting of rights enabling user to further grant rights shall be avoided.

6.5.2.1.9. Application design and database configuration shall ensure audit trails. Also, logging for user and session auditing shall be activated.

6.5.2.1.10. Technology users who have server access (as mandated by their roles) shall not create any new databases / load new applications on the servers unless it's been intimated to Datacenter Manager / Application /Information Owner / Business Head and updated in the DC System Inventory List.

6.5.2.1.10. Access rights shall be granted through roles only Users shall not be assigned rights directly.

6.5.2.1.11. Inventory of data bases across various applications shall be maintained. This shall be the responsibility of the DC Manager.

6.5.2.1.12. Each database administrator who requires DBA privileges to access or change database parameters/configuration shall be identifiable at database level or with appropriate records maintained by the Datacenter Manager.

6.5.2.1.13. Number of users having DBA privileges at database level shall be restricted to minimum

6.5.2.1.14. All database access activity shall be monitored.

6.5.2.1.15. Audit logs shall be enabled and stored for a minimum period of 3 months or as required by business logic.

6.5.3. Backup and Recovery from Failure: Database Administrator shall ensure:

6.5.3.1. Suitable documented backup/ recovery procedures are in place, which shall cover type of backup, periodicity, location, testing and other relevant aspects

6.5.3.2. Log files for database reconstruction (Redo log files in case of Oracle) are mirrored so as to keep at least two current copies for disaster recovery.

6.5.3.3. DBMS is configured so as to enable proper storage of recovery information. E.g. in Oracle, the checkpoint process shall be activated to ensure that recovery information is written to the data file headers and special control files which record the recovery check point.

6.5.3.4. Regular file system backup are also taken. If hot backup is taken, it shall preferably be taken when database update operations are minimal.

6.5.4. Database Administrator shall follow these Internal procedures such as:

6.5.4.1. Keep track of the latest security patches for the databases. Database

security patches shall be applied in a timely manner. In the case of emergency patches, updates shall be made outside of the maintenance window.

6.5.4.2. Production database environment to be separate from development database environment & no sensitive production data is populated into the development environment unless authorized. All production data shall be sanitized before using in the development or test environment.

6.5.4.3. Proper security controls shall be on tape backups, data mirrors or any derived exported data so that unauthorized access to data stored on alternate media can't be obtained.

6.5.4.4. A hardening document shall be prepared for each type of database platform. All databases created for / being moved to production environment shall be subjected to the hardening process as specified in this document

6.5.4.5. Databases shall be monitored to prevent overloading and identify performance issues using appropriate tools.

6.5.4.6. Database administrator shall be responsible for configuration of the database and managing access to the database.

6.4. Testing:

6.4.1. System test design shall be based on established testing methodologies, such as black-box testing, white-box testing, and gray-box testing, depending on the nature of the application and its requirements.

6.4.2. Test scenarios and test cases shall be developed to validate the functional, performance, and security aspects of the software application.

6.4.3. Development, test, and production environments shall be logically and physically separated to prevent unauthorized access and minimize the risk of unintended changes impacting live systems.

6.4.4. Operational data and test data shall not reside on one machine. The testing group shall be segregated from the operational group. Copying of operational data to test facility shall be through change control/ exception control.

6.4.5. Test data shall be representative of real-world scenarios and shall include both valid and invalid data to ensure comprehensive test coverage

6.4.6. Access to system test data shall be provided to identify personnel as per the application group and Test results/logs shall be documented and maintained for audit references.

6.4.7. Test data blocks shall be archived and disposed through a formal process as approved by the application owner/ Head Application Development Team as per the regulatory guidelines.

6.4.8. Physical access to test facilities shall be regulated. Access shall be provided through a written approval stating reasons. Access shall be on a case-to-case basis only. Biometric Access shall be considered if necessary.

6.5. Deployment and Maintenance:

6.5.1. When the program source code in the development environment is frozen for changes the program source code shall be given to the application administrator for escalating the code to the live environment

6.5.2. The Application Owner/ Information Owner shall ensure that:

6.5.2.1. A change impact analysis is performed on the program source code.

6.5.2.2. The User Acceptance testing is performed on the program source code. After the successful test a sign, off is taken from the appropriate authority.

6.5.2.3. Load testing is performed on the live environment wherever possible

6.5.2.4. A final quality review is done before it is sent back to the Process / Application/Information owner/ Manager – Applications

6.5.2.5. The application system shall be rolled back to a consistent state and the discrepancies shall be reported to the development team in the event of non-compliance for the above.

6.5.2.6. After successful reviews and tests to the development program source code a signed approval shall be given to the Application / Information /Process Owner/ IT TEAM/Manager - Applications, following which the administrator shall put the program source code to the live environment.

6.5.2.7. Daily backup of configuration server on which source code library shall be

taken. Monthly back up of configuration server shall be taken at the end of the month. Offsite backup shall be kept at another location. (for details refer to the Backup, Restoration and Media Handling Procedure).

6.5.3. Patch management:

6.5.3.1. Application/Database Administrator shall track the patches for the Application/Database. Also, shall be part of the application vendor's and other security site's mailing lists.

6.5.3.2. Application/Database administrator shall ensure that Application of patches are prioritized and scheduled depending on the criticality.

6.5.3.3. Patches are tested in a test environment before applying them in the live environment. Patches shall be tested for the integrity of the patch and stability of the application/database system shall be tested after applying the patches

6.5.3.4. Patches shall be rolled out in live environment, initially on less critical systems and then on all the systems.

6.5.3.5. Backups shall be taken before application of patches and Software documentation shall be updated accordingly.

6.6. Documentation:

6.6.1. Process Owner / Application Owner in consultation with the IT Application Team shall maintain the program documentation and ensure that program documentation and its versions are updated corresponding to the changes to the program files. Program documentation shall include:

6.6.1.1 Source code (wherever possible).

6.6.1.2. User Manual

6.6.1.3. Technical manual

6.6.1.4. Configuration files

6.6.1.5 Test plans, cases, and results

6.6.2. The IT TEAM/ Manager–Applications shall ensure that, users authorized by the Application owner / Information Owner / Process owner / Department Head and the developers shall be given an access to the program documentation

6.6.3. The documentation shall be made available for access on a need-to-know basis to the identified members in the Application Development Team and Business Owners as the case may be.

6.6.4. Access activity to the documentation portal shall be audited and user activity shall be logged.

6.7. Other associated controls:

6.7.1. Access Controls:

6.7.1.1. Access to the applications of J M Baxi Group shall be restricted to authorized users. Access to each application shall be authorized by the respective the Department Head and where they are different, by both. Users shall submit an access request form clearly indicating the options / levels in the application to which access is required with appropriate justification. A separate access request form shall be designed by each application / information owner for applications owned by him / her.

6.7.1.2. User access to applications shall be based on the principle of least privilege and a "need to know". As per this principle, access to an application and the associated data shall be given only to those people who have a legitimate business need for the same. Accordingly, access rights of users shall be commensurate with their job responsibilities.

6.7.1.3. Access to the directories holding database and application programs shall be as per the vendor specifications, which shall be guided by the principle of least privileges or need to know basis.

6.7.2. Change Management:

6.7.2.1. During the cut over period during implementation of developed / procured new software application (old system has been stopped and new system has not gone on live) it is the responsibility of the Process Owner/Department

Head**.**

6.7.2.2. To provide accurate information/data for uploading into new system as per the necessary format.

6.7.2.3. Validation of accuracy and completeness of uploaded data/ information is the responsibility of the Process/ Application/Information owner/Department Head and sign off shall be given on the correctness and completeness of the data/information.

6.7.2.3. **Emergency change management:** All emergency changes to the application (major code movement, server hardware change, Operating System /application version upgrade) which cannot follow the change request procedure shall be approved by IT Team Head (on mail) & Application & Infra Head (if the server is Internet facing) and formalized the next day as per the Change Management Procedure. Subsequent mail or telephonic approvals shall be entered into the Ticketing Tool application and the mail copy of the request and the approval shall be uploaded in to DC Governance

7. Responsibilities

The responsibilities lie with the following personnel:

7.1. **Application Owner / Information Owner / Process Owner / Head of Department:**

7.1.1. Document the security requirements for the application.

7.1.2. Verify the implementation of documented security requirements.

7.1.3. Approve upgrades to the application (Along with the IT Team).

7.1.4. Ensure that application design meets business requirement.

7.1.5. Ensure that there are appropriate security controls for the application.

7.1.6. Oversee the development of the software under development.

7.1.7. Ensure a proper version control of the source code of the software in production and under development.

7.1.8. Maintain documentation of the software till final completion and handover.

7.1.9. Perform user acceptance testing.

7.2. **Application Administrator:**

7.2.1. Install applications.

7.2.2. Ensure secure configuration of applications.

7.2.3. Implement changes to the application.

7.2.4. Track application patches.

7.2.5. Update patches as per the procedure

7.3. **Database Administrator:**

7.3.1. Configure and administer database.

7.3.2. Grant access to database.

7.3.3. Track database patches.

7.3.4. Update patches as per the procedure.

7.4. **Manager – Applications/Head – Security In-charge:**

7.4.1. Ensure that the business and information security requirement is fulfilled by the application design in consultation with process owner.

7.4.2. Approve upgrades to the applications (Along with the Application/ Information Process owner).

7.4.3. Evaluate controls to be built into the applications.

7.4.4. Approve the change request made by the User.

7.4.5. Approve changes requested to the programs in the software program library.

7.4.6. Approve implementation of changes to the live environment along with Process/Application/Information owner.

7.4.7. Ensure that security requirements and data validations are implemented in the application.

7.4.8. Perform application testing.

7.4.9. Conduct risk assessment during evaluation of the application.

7.4.10. Control access to program libraries.

7.4.11. Version control of program files.

7.4.12. Maintain software documentation with access control.

7.5. **IT Team Application Group:**

7.5.1. Ensure that the business requirement is fulfilled by the application design in consultation with process owner.

7.5.2. Approve upgrades to the applications (Along with the Application/ Information / Process owner).

7.5.3. Approve implementation of changes to the live environment along with Application / Information / Process owner.

7.5.4. Approve the change request made by the user.

7.5.5. Conduct risk assessment during evaluation of the application.

7.6. **Application Developer:**

7.6.1. Ensure that security features as mentioned in the procedure are incorporated during the application design and development.

7.7. **Program Librarian:**

7.7.1. Maintain the security of program control area.

7.7.2. Carry out the check-in, check-out and induction of a program to live environment as per the specified procedure

8. Enforcement

8.1 . This policy and procedure is applicable for all the employees of the company who have access to and use the information assets and IT assets as listed in the Information Asset. Classification sheet which has been created for all the departments. Any violation shall be dealt in accordance with the disciplinary action process as laid down in the Code of Conduct.

8.2. Management's interpretation of the clauses in this procedure shall be final and binding. Management reserves the rights to alter or amend any clause in this document at any time as per its discretion.