STANDARD OPERATING PROCEDURE

Audree Infotech	**Title:** Software Development Life Cycle	SOP No.	AUD/SOP/DEV/001
		Version No.	00
		Department	Software Development
		Effective Date	31-07-2018
		Review Date	31-07-2020
		Page No.	1 of 16

1.0 OBJECTIVE

To lay down the procedure for Software development life cycle.

2.0 SCOPE

This procedure is applicable for Software development department of Audree Infotech Pvt. Ltd

3.0 RESPONSIBILITY

3.1 **Software development Head:** Development Team Lead is experienced and knowledgeable individual, responsible for a group of Developers and lead the development and delivery of code underlying release features and functionality.

3.2 **Developers**: are a group of software engineers, at various levels of proficiency and knowledge of the SDLC, under the supervision of a Development Team Lead they generate and pre-QA tests the code.

3.3 **Project Manager**: Manage Project Plan, Communication with the Business about project related

3.4 **QA:** Supporting and Approving the SDLC related documents

4.0 PROCEDURE

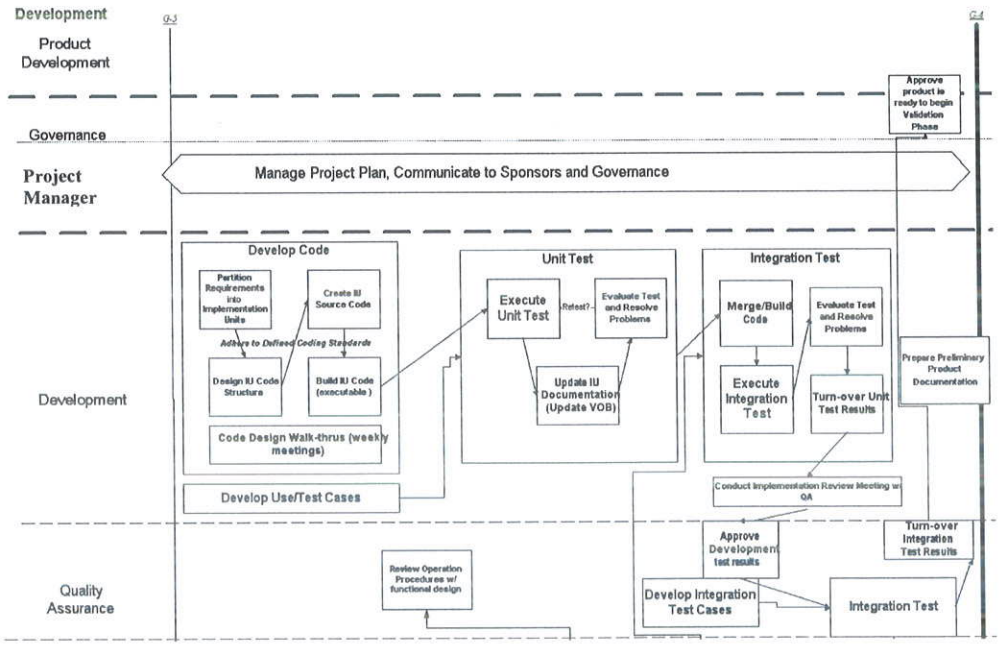
4.1 This procedure provides methodology for successful management of a software project with respect to on time delivery and with required quality.

PREPARED BY	REVIEWED BY	APPROVED BY
User Department	Head of the Department	Quality Assurance
Sign: A. Patil Reddy	Sign:	Sign:
Date: 31-07-2018	Date: 31-07-2018	Date: 31-07-2018

4.2 The systematic execution of a System Development Life Cycle (SDLC) for a release or projects that have significant impact on an organization's service delivery.

4.3 Throughout the technical design and interface specification, and software development activities interactions with areas outside Department will be funnelled through the Project Manager. The Project Manager will capture the issue, constraint or opportunity raised by the Developers to their Development Team Lead.

Procedure Activities are define as per the below Fig.



PREPARED BY	REVIEWED BY	APPROVED BY
User Department	Head of the Department	Quality Assurance
Sign:	Sign:	Sign:
Date: 31-07-2018	Date: 31-07-2018	Date: 31-07-2018

Audree Infotech	**Title:** Software Development Life Cycle	Doc. No.	AUD/SOP/DEV/001
		Version No.	00
		Department	Software Development
		Effective Date	31-07-2018
		Review Date	31-07-2020
		Page No.	3 of 16

4.4 This procedure oversees the SDLC execution; thus, it relies heavily on defined procedure activities and acceptance criteria for inputs and outputs

Software development process involves the following steps:

4.5 Project Initiation

4.5.1 The process starts with completion of initial contract with client by Business Development Division for development/ enhancement/ customization activity. In case of internal project the initiation can be done by any department, under approval of Management. The projects are numbered using following convention:

AUD/SD/2016/001

(Organization)/ (SD-Software Development)/ (Year)/ (Sequence No.)

The Project Master (Refer AUD/SOP/DEV/001/A01) sheet is updated with the project Code and details.

4.6 Scope Understanding

4.6.1 Project Manager or Project Lead ensures to understand clearly the scope and delivery requirements. User requirements are discussed with Business Development and client to obtain clear understanding of project scope, objectives and deliverables.

4.6.2 A Project Initiation Form (Refer AUD/SOP/DEV/001/A02) is used to document the details.

4.7 Requirements Management

4.7.1 Project Lead defines requirements for hardware & software based on the URS/BRS.

PREPARED BY	REVIEWED BY	APPROVED BY
User Department	Head of the Department	Quality Assurance
Sign: Ch. Sahil Reddy	Sign:	Sign:
Date: 31-07-2018	Date: 31-07-2018	Date: 31-07-2018

4.7.2 These requirements are documented as per the template given in Functional Software requirement specification (Refer AUD/SOP/DEV/001/A03 or AUD/SOP/DEV/001/A04) template.

4.7.3 The initial requirements documented as Functional/Software Requirement Specification (Ref. AUD/SOP/DEV/001/A03 or AUD/SOP/DEV/001/A04) are analyzed for technical feasibility, completeness, errors and omissions etc. The detail analysis is documented by updating Functional/Software Requirement Specification (Refer AUD/SOP/DEV/001/A03 or AUD/SOP/DEV/001/A04). The FRS/SRS is sent to the client for review & approval.

4.8 Risk Management

4.8.1 The Project Manager/Project Lead is responsible to identify, analyze and control the risks involved in projects. Risks are categorized by impact and likelihood on a scale defined in Project Plan AUD/SOP/DEV/001/A05, The actions for mitigation of risks are reviewed for effectiveness.

4.9 Project Planning

4.9.1 During the project planning phase, the project team with representatives from software department, QA and project manager, provides a balanced perspective on the project plan as per AUD/SOP/DEV/001/A05. This plan is updated by Project Manager / Project Lead throughout the project life cycle for changes in requirements, scope of work, schedule.

4.9.2 If the project can be developed by reusing an application previously developed or at least a part of the same, such decisions are also documented in project plan. If any components/functions/ service/ utilities are available as external

Provider/ third party product or service, the same is also envisaged and documented in External Provider Item QA AUD/SOP/DEV/001/A06

4.10 High Level Design

4.10.1 The high level design activities include developing a High Level Design Document (Refer AUD/SOP/DEV/001/A05) for the project/ project package based on the requirement analysis using FRS/SRS (Refer AUD/SOP/DEV/001/A03 or AUD/SOP/DEV/001/A04)

4.10.2 The modules list, broad flow charts, business logic, functional and performance aspects are understood, reviewed and drawn up as part of High Level design.

4.11 Detailed Design

4.11.1 The detailed design activities are documented in Detailed Design Document (Refer AUD/SOP/DEV/001/A08) which includes conceptualization and design aspects, related Fields and Field Characteristics Identification, Database Design, Relationship Diagrams, Validations and Flags, Security, Scalability, Usability, etc.

4.11.2 Project Lead/QA develops and defines the procedures and criteria which elaborate the testing, verification, acceptance criteria, outputs of interfaces, performance parameters etc. for the integration process.

4.12 Design of interfaces

4.12.1 Project Lead and Design team develops the interfaces for various modules and product based on requirements such as origination, destination, data characteristics, event sequencing, exceptions handling, external items/ services as required. QA Analysts and Design team ensure to include list of interfaces,

Mapping of interfaces to product components, review of interface test data, input-output protocols and details for ensuring suitability of successful integration.

4.13 Implementation of Product Design (Coding)

4.13.1 Based on high level and detailed design as well as FRS/SRS, the tasks are allocated to the project team members for coding and development of required functionalities.

4.13.2 The Project Team during the coding phase ensures that the specified requirements are met, acceptance criteria are satisfied, appropriate regulatory requirements are followed and those characteristics of the design that are critical to the safe and proper functioning of the product/ service are fulfilled.

4.14 Coding Standards

4.14.1 Coding standards are important as they lead to greater consistency within the code. A greater consistency lead to code that is easier to understand, which in turn means it is easier to develop and to maintain. In general code will exist for a long time, long after the programmer moved on to other projects. An important goal during development is to ensure easy transition of work to another developer, or to another team of developers, so that they can continue to maintain and enhance the work without having to invest an unreasonable effort on existing code. Code that is difficult to understand runs the risk of being scrapped and rewritten, raising the cost of development and maintenance.

4.15	**Good programming Style**
4.15.1	With the help of good style in programming, we can have code reuse and easier maintenance of larger projects.
4.15.2	Programming style typically includes the following:
4.15.3	A comment summarizing each class/file.
4.15.4	Concise comments summarizing each method's expected input and output.
4.15.5	Good use of white space to separate sections of code and improve clarity.
4.15.6	Descriptive variable and method names.
4.15.7	Logically well-organized classes/files and methods/procedures.
4.15.8	Consistent indentation.
4.15.9	No long, wrapped lines.
4.15.10	No over commenting.
4.16	**Standards and Guidelines**
4.16.1	**Naming Conventions**
4.16.2	Though naming conventions are language-dependent, the following guidelines can be adopted if the language permits:
4.16.3	Meaningful names are used. Typically, they are formed by concatenating words or abbreviations.
4.16.4	Names longer than 15 characters are avoided.
4.16.5	Short names such as 'i' are only acceptable for loop indexes or temporary variables with a scope of only a few lines.
4.16.6	The names of constants are all upper case with underscores separating each word/abbreviation

4.16.7	Care must be taken to avoid clashes of global names.
4.16.8	Names with leading and trailing underscores are reserved for system purposes and should not be used for any user-created names
4.17	**Formatting Style and Comments**
4.17.1	**Identification**
4.17.1.1	Each source file should contain identification details like the name of the project, the name of the component and name of the program at the top of the file.
4.17.2	**Description**
4.17.2.1	Brief description of the purpose of the file. This description includes purpose of this file, its significance and which requirement this file satisfies.
4.18	**Revision History**
4.18.1	A clear revision history of source file provides change flexibility and greater maintenance of code.
4.19	**Copyright**
4.19.1	A copyright notice can be inserted. The format of the notice is specific to the project.
4.20	**File Body (content)**
4.20.1	The source file body, normally consists of the following type of items:
4.20.1.1	References to other files
4.20.1.2	Global data declarations
4.20.1.3	Code (Function/ Method/procedures)
4.20.1.4	Comments

4.20.2	Function/Method/Procedure declarations
4.21	**Function / Method / Procedure header**
4.21.1	Function/Method/Procedure header generally consists the following parts:
4.21.2	Function/Method/Procedure Name with Modifier/Scope
4.21.3	Function/Method/Procedure Description
4.21.4	Input Parameters
4.21.5	Output Parameters
4.21.6	Return Type
4.21.7	Updating History with the details like Updated on, Updated by, brief description about the change etc.
4.22	**General Rules for Programming**
4.22.1	Following are the rules for general programming Standards.
4.22.2	When modifying code, adapt to the general standards of language and programming rather than imposing your own.
4.22.3	In general, every function/method/procedure, unless trivial should have its own comments that describe the purpose of the function/method/procedure, the input and output arguments that the function /method/procedure produces.
4.22.4	Within a function/method/procedure, separate major blocks of code and identify their purpose with a leading comment. Do not bother to comment individual source lines, unless there is an underlying discretion that needs to be explained.
4.22.5	Condensed code might seem clever when you write it, but it probably will not, over a period of time.
4.22.6	An equal importance should be given for code clarity and code efficiency.

STANDARD OPERATING PROCEDURE

4.22.7	Attempt to write linear code - i.e. code that starts at the first executable statement and flows down to a final "return" or end of block statement.
4.22.8	Check the return values for function/method/procedure and do not make any assumptions.
4.22.9	If the function/method/procedure operates only on a limited range of input values, check at the beginning of the function
4.22.10	Avoid complicated "if" constructs. It is better to use several simpler nested "if" constructs rather than a complicated compound one.
4.22.11	Avoid implicit precedence rules and use parentheses to identify the components of such constructs. You can remember all the precedence rules, but this helps the programmers who do not know the precedence rules.
4.22.12	Indent each nested block in order to highlight the start and end of each of them.
4.22.13	Avoid implicit data-type conversions and mixed mode arithmetic operations. Use explicit operators (e.g. (float)...). The use of explicit conversions will have no effect on code efficiency and might avoid any inadvertent rounding or truncation errors. This will highlight the data type conversions to anyone attempting to understand the code.
4.22.14	From project to project standards to be followed may vary depending on factors like client's insistence on a particular set of standards. But, whenever a project team goes against organizational standards, the variations should be well documented.

4.23 Unit Testing

4.23.1 Applicable coding standards as per the technology such as Xunit Tool are followed and developer after completing coding, carries out basic self- testing to check the functionality of the module and can perform corrections then and there.

4.23.2 Code review is conducted by the Project Lead to detect defects at an early stage of developed functional units. Any defects identified are intimated for resolution.

4.24 Integration Testing

4.24.1 Different functions are developed and tested during unit testing. The unit-tested programs are integrated into modules/sub systems. Integration testing is done by the Development team according to the integration plan developed during the planning stage.

4.24.2 These functions are integrated as per project plan (Refer AUD/SOP/DEV/001/A05) the functioning of integrated products/ modules is tested by Development Team. The defects and resolution of the defects is allocated and tracked.

4.25 Testing

4.25.1 System testing is done by QA team as per SOP No: QA/SOP-001. This validates the complete system against the client requirements.

4.26 Validation

4.26.1 Validation is performed by Audree/Client/Third Party as per client requirement in client environment after completion of all testing activities like unit testing, integration testing & system testing.

PREPARED BY	REVIEWED BY	APPROVED BY
User Department	Head of the Department	Quality Assurance
Sign: ch. Sunil Reddy	Sign:	Sign:
Date: 31-07-2018	Date: 31-07-2018	Date: 31-07-2018

4.26.2 Product validation is performed in i) Pre-production and ii) Production environment according to requirements. Validation outcome is reviewed for ensuring that the software/product is functioning correctly, consistently and reliability and is complete with respect to release.

4.27 User Acceptance Testing

4.27.1 When contractually required, acceptance testing is performed by the customer against the plan, drawn up by Audree Infotech in collaboration with the customer.

4.27.2 Pre-production version releases may be given to client as alpha and beta versions for testing and feedback. This may involve installation at customer site for active testing and evolving the software into production version. When contractually required, an installation plan is prepared & details documented in Project Plan (Refer AUD/SOP/DEV/001/A05)

4.28 Test Tools

4.28.1 The usage of automated test tools is identified by Project Leader during the Project Planning stage (Refer AUD/SOP/DEV/001/A05) for each project. Tools also include, the testing software tools internally developed, customer supplied, third party provided or provided by any external provider to verify and validate the developed software product/ application.

4.28.2 The capability and configuration of automated test tools for the test cases and objectives are evaluated by QA team and the planning for use of such tools is documented in Project Plan (Refer AUD/SOP/DEV/001/A05 & AUD/SOP/DEV/001/A11)

4.28.3	When such automated tools are used on regular basis the reference to a previous validation and suitability of such tools is given in Project Plan.
4.29	**External Providers related Controls**
4.29.1	External Providers/ Customer supplied product/ standards/ methodology/ as/ programs/ data are reviewed for compatibility, intellectual property rights and suitability to the project requirements by Project Manager/Lead. The tracking and Quality Assurance activities of such items are carried by Project Manager/Lead.
4.29.2	An updated list of such External Provider items or any software (Refer AUD/SOP/DEV/001/A06) to be included is maintained by the Project Manager/ Project Lead. The inventory, suitability and handover related to external provider's items is maintained.
4.30	**Release, Deployment and Delivery**
4.30.1	Final version of software is released against Software Release Note (AUD/SOP/DEV/001/A09). AUD/SOP/DEV/001/A01) for details.
4.31	**Project Monitoring and Control**
4.31.1	During the SDLC, it is ensured that clear instructions are provided by Project Manager/ Team Lead to Project Team personnel performing tasks as per Project Plan (AUD/SOP/DEV/001/A05) and related documents are reviewed
4.32	**Interaction Management**
4.32.1	Throughout the SDLC, interactions with Business Development, clients, management and teams are ensured for coordination and proactive focus on project goals.
4.33	**Corrective Actions**

4.33.1 The corrective actions can be initiated during the SDLC when the results are not as per the plans and requirements or there are customer complaints regarding dissatisfaction of Audree Infotech services.

4.34 Support and Enhancement

4.34.1 During the support phase, the major activities include problem fixing and enhancements as initiated by clients or initiated by Audree Infotech as part of improvement of product proactively or based on problems reported in similar functionalities of their projects. Such defects are logged and resolved.

4.34.2 Enhancements are considered as extensions/ modifications to the original product. The enhancements are executed as per the Change Management procedure.

4.35 Project Documentation

4.35.1 Project Documents Master List Sheet (AUD/SOP/DEV/001/A10) is prepared which incorporates for all the documents and their versions as part of project management activities, which are maintained in the form of softcopy. Any important hardcopy documents are scanned and stored in the project folders.

4.36 Customer Feedback

4.36.1 Refer Procedure - Procedure for obtaining customer feedback (MKT/PRO/001).

4.37 Project Closure

4.37.1 Once all deliverables and services are provided as required, the projects are considered as Closed Projects & details documented in Project Master (Refer AUD/SOP/DEV/001/A01).

PREPARED BY	REVIEWED BY	APPROVED BY
User Department	Head of the Department	Quality Assurance
Sign: Ch. Sahil Reddy	Sign: Regie.	Sign:
Date: 31-07-2018	Date: 31-07-2018	Date: 31-07-2018

STANDARD OPERATING PROCEDURE

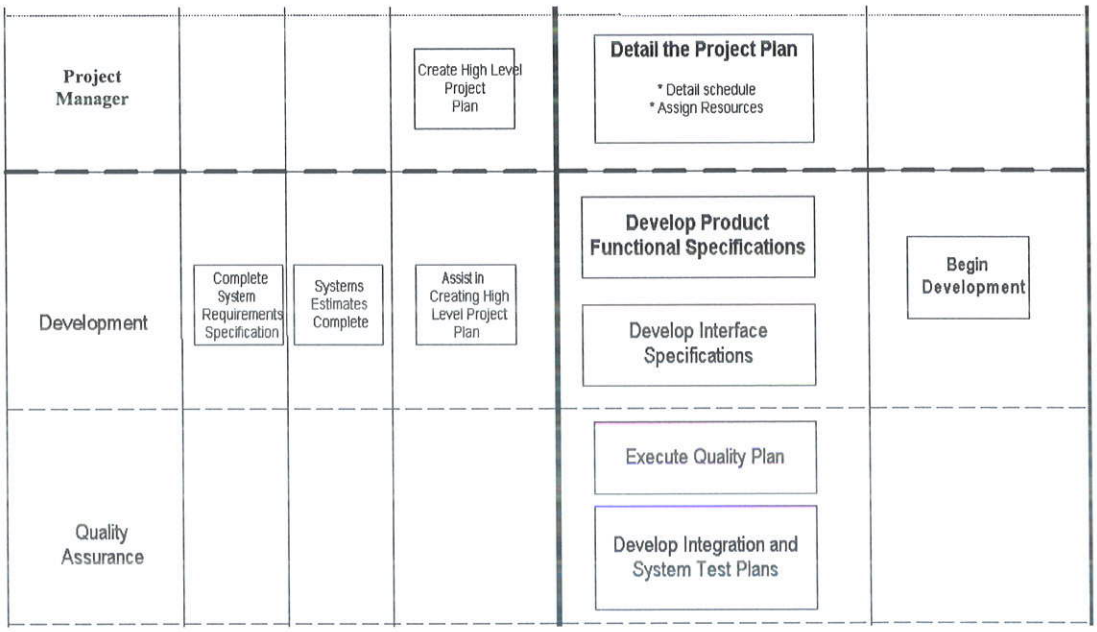
5.0 ABBREVIATION(s)

Abbreviation	Full Description
SOP	Standard operating procedure
QA	Quality Assurance
URS	User Requirement Specification
FRS	Functional Requirement Specification
SRS	System requirement Specification

6.0 REFERENCE(s)

Not Applicable

7.0 FLOWCHART(s)



PREPARED BY	REVIEWED BY	APPROVED BY
User Department	Head of the Department	Quality Assurance
Sign: el. Lalit Reddy	Sign:	Sign: 31-07-20
Date: 31-07-2018	Date: 31-07-2018	Date: 31-07-2018

MASTER COPY

STANDARD OPERATING PROCEDURE

8.0 ANNEXURE(s)

Annexure No.	Details/Title of Annexure	Format No. (Current version)
AUD/SOP/DEV/001/A01	Project Master	AUD/SOP/DEV/001/F01
AUD/SOP/DEV/001/A02	Project Initiation Form	AUD/SOP/DEV/001/F02
AUD/SOP/DEV/001/A03	Functional Requirement Specification	AUD/SOP/DEV/001/F03
AUD/SOP/DEV/001/A04	System Requirement Specification	AUD/SOP/DEV/001/F04
AUD/SOP/DEV/001/A05	Project Plan	AUD/SOP/DEV/001/F05
AUD/SOP/DEV/001/A06	External Provider Items QA	AUD/SOP/DEV/001/F06
AUD/SOP/DEV/001/A07	High Level Design Document	AUD/SOP/DEV/001/F07
AUD/SOP/DEV/001/A08	Detailed Level Design Document	AUD/SOP/DEV/001/F08
AUD/SOP/DEV/001/A09	Software Release Note	AUD/SOP/DEV/001/F09
AUD/SOP/DEV/001/A10	Documents Master List	AUD/SOP/DEV/001/F10
AUD/SOP/DEV/001/A11	Test Cases Sheet	AUD/SOP/DEV/001/F11

PREPARED BY	REVIEWED BY	APPROVED BY
User Department	Head of the Department	Quality Assurance
Sign:	Sign:	Sign:
Date: 31-07-2018	Date: 31-07-2018	Date: 31-07-2018

MASTER COPY

Audree Infotech Pvt. Ltd.

Change History

Document / SOP Title: Software Development life cycle			
Department: Software Development		Document / SOP No.: AUD/SOP/DEV/001	Version No.: 00
Effective Date: 31-07-2018		Review Date: 31-07-2020.	

Current Version	Obsoleted version	S. No.	Changes made
DEV/SOP/001 AUD/SOP/DEV/001	New DEV/SOP/001	1	New Procedure is made Change Control Number: CRN/AUD/18/004. Effective Date: 31-07-2018.
		1	Updated the procedure as per the new format proposed as per SOP AUD/SOP/QA/001 and accordingly contents such as Coding standards good programming style naming conventions etc. have been incorporated.
		2	Change Management annexure is excluded from this procedure and covered as part of SOP No. QA/SOP/004

Summarized By	Reviewed By	Approved By
User Department	Head of the Department	Quality Assurance
Sign: ch. Lakshmi Reddy	Sign:	Sign:
Date: 31-07-2018	Date: 31-07-2018	Date: 31-07-2018

Project Master

STATUS AS ON:

S.No.	Client	Date of Initiation	Project Code	Project Details	Date of Completion

Project Initiation Form

Name/ Address of Client	:	
Reference of Proposal/ Work Order	:	
Contact Person	:	
Profiling of Customer	:	

Project Description:

Objective of Project:

Scope of Project:

Brief Description of need, constraints and expectations:

Expected Completion Date	:	

Instructions for billing:

Any other preference of the client:

Project Code Allocated	:	
Initiated By	:	

FUNCTIONAL REQUIRMENT SPECIFICATION

	Prepared By	Reviewed By	Approved By
Name			
Designation			
Sign			
Date			

Table of contents:

1. Quality Assurance

1.1 Change history

Version	Change Request	Effective Date	Reason for Change

1.2 Related Documents

Doc. No	Document Name	Notes

2. Introduction

3. Process description

Process Step	Role	Tasks
Create change request	Reporter	• Describe the change • Add related attachments • Fill out customizable questionnaires
Carry out initial review	Approver (supervisor, manager, reviewer)	• Check completeness of the change request • Edit request to improve input • Decide viability of the change request • Approve submission of the change request
Approve change request	Approver	• Check completeness of the change request • Edit the change request to improve input • Decide if the change request is relevant for the process • Reject the change request if it is not relevant
Drive change request	Coordinator	• Release the change request • Define an appropriate activity hierarchy, timelines and activity owners • Trigger activities in the defined sequence • Monitor the change request • Close the change request when all activities are completed
Execute activities	Activity owner (approver, reviewer)	Execute activities by teams or individuals, for example, reviews, approvals using checklists or questionnaires

4. Configuration Overview

5. Terms & Definitions

6. Realization of user requirements

User requirement	Ref. No	Sub Process	Description	Remarks	Realized By	Realization Statement	Document reference (Configuration Document or Design Specification)

7. Annexure

SYSTEM REQUIREMENT SPECIFICATION

Prepared by	Reviewed & Approved by

Revision Chart

Version	Primary Author(s)	Description of Version	Date Completed

PREFACE

CONTENTS

1. INTRODUCTION

1.1.Purpose

1.2.Scope

Objective:
Benefits:
Goal:

1.3. Definitions, Acronyms and Abbreviations

1.4 References

1.5 Overview

2. PRODUCT DESCRIPTION

2.1 Product Perspective

2.1.1 System Interfaces

S.No.	Type of Activities

2.1.2 User Interfaces

2.1.3 Hardware Interfaces

2.1.4 Software Interfaces

Software Type	Product	Mnemonic	Version	Source

2.1.5 Communications Interfaces

2.1.6 Memory Constraints

2.1.7 Operations

2.1.8. Site Adaptation Requirements

2.2 Product Functions

2.3 User Characteristics

- Educational level:
- Experience:
- Technical Experience

2.4. Constraints

2.5. Assumptions and Dependencies

2.6. Apportioning of Requirements

3. SPECIFIC REQUIREMENTS

3.1 External Interface Requirements

3.2 Product Features

3.3 Business Rules

3.4 Performance Requirements

3.5 Design Constraints

3.6 Software System Attributes

3.6.1 Reliability

3.6.2 Availability

3.6.3 Security

3.6.4 Maintainability

3.6.5 Portability

4. INDEX (IF ANY)

5. APPENDICES (IF ANY)

6. RECORDS / FORMATS

Req. No.	Requirements specified by client	Client's Reference	Modules addressing the requirement	Remarks
Req. No.	Requirements not specified by client but necessary	Our reference	Modules addressing the requirement	Remarks
Req.No.	Statutory/Regulatory Requirements	Reference	Modules addressing requirement	Remarks

Req. No.	Performance Requirements	Reference	Modules addressing requirement	Remarks

Req. No.	Compatibility Requirements	Our reference	Modules addressing requirement	Remarks

Detailed Requirements for various Objects/ Fields:

Object/ Field Name	Mandatory ?	Data Type	Control Type	Maximum Length	Database Table	Database Column	Validations	Business Rules	Reference Modules	Comments

Any other requirements/remarks/conclusion:

Req. No.	Requirements:

Following checklist to be referred while analysis of Requirements:

S. No.	Check Point	OK/ Not OK	Remarks
A	Organization and Completeness		
1.	Are all cross-references to other requirements correct?		
2.	Are all requirements written at a consistent and appropriate level of detail?		
3.	Do the requirements provide an adequate basis for design?		
4.	Is the implementation priority of each requirement included?		
5.	Are all hardware, software, and communication interfaces defined?		
6.	Have algorithms intrinsic to the functional requirements been defined?		
7.	Does the specification include all of the known customer or system needs?		
8.	Is the expected behavior documented for all anticipated error conditions?		
B	Correctness		
9.	Do any requirements conflict with or duplicate other requirements?-		
10.	Is each requirement written in clear, concise, unambiguous language?		
11.	Is each requirement verifiable by testing, demonstration, review, or analysis?		
12.	Is each requirement in scope for the project?		
13.	Is each requirement free from content and grammatical errors?		
14.	Is any necessary information missing from a requirement? If so, is it identified as TBD (to be decided)?		
15.	Can all of the requirements be implemented within known constraints?		
16.	Are any specified error messages unique and meaningful?		
C	Quality Attributes		
17.	Are all performance objectives properly specified?		

18.	Are all security and safety considerations properly specified?		
19.	Are other pertinent quality attribute goals explicitly documented and quantified, with the acceptable tradeoffs specified?		
D	Traceability		
20.	Is each requirement uniquely and correctly identified?		
21.	Is each software functional requirement traceable to a higher-level requirement (e.g., system requirements)		
E	Other Issues		
22.	Are all requirements actually requirements, not design or implementation solutions?		
23.	Are all time-critical functions identified, and timing criteria specified for them?		
24.	Have internationalization issues been adequately addressed?		
25.	Any other point:		

Checked by: Approved by:

Date: Date:

Project Plan

Project Code:	**Date Initiated:**	**Initiated By:**
Project Description:		
Objective of Project:		
Scope of Project:		
Life Cycle Model Description/ Reference:		
Planned Start Date:	Expected Completion Date:	

Project Deliverables

S. No.	Details	Remarks

Modules List:

S. No.	Details	Whether the module/ component is reusable from Previous Projects?	Reusability Remarks

Resources Required:

Software Resources/ Tools	Hardware Resources	Manpower	Required Skills/ knowledge	Other Resources/ Tools	Resource availability	Remarks

Any special procurement activity:

Any special training needs/ competence development activity:

Project Plan

Project Team:

Person Name/ Department name	Role	Responsibility	Other projects Handled	Role in other projects	Remarks for avoiding overloading/

Stakeholders Involvement Matrix:

SDLC Phases/ Activities	Client	Management	Business Development	IT Services	Others
Requirements Management	Y/N	Y/N	Y/N	Y/N	Y/N
Design	Y/N	Y/N	Y/N	Y/N	Y/N
Coding	Y/N	Y/N	Y/N	Y/N	Y/N
Testing	Y/N	Y/N	Y/N	Y/N	Y/N
Deployment	Y/N	Y/N	Y/N	Y/N	Y/N
Maintenance	Y/N	Y/N	Y/N	Y/N	Y/N
Project Management	Y/N	Y/N	Y/N	Y/N	Y/N

Estimation, Schedule and Work Breakdown Structure:

S. No.	Tasks	Complexity (High/	Dependencies	Size	Responsibility	Expected

Risk Management Plan:

S.No.	Details of Risk	Severity	Likelihood	Risk Value	Risk Mitigation Plan

Project Plan

Severity Scale:

1=Low impact on Quality, Schedule or Budget of Project, 2=Medium Impact, 3=High Impact, 4=can lead to customer complaints 5=Can lead to project cancellation,

Likelihood Scale:

1=Rare Possibility, 2=Likelihood of 1-3 times during the project (typically 3 months project), 3=Likelihood of 3-5 times during the project (typically 3 months project), 4= Likelihood of 5-10 times during the project (typically 3 months project), Likelihood of >10 times during the project (typically 3 months project)

Risk Value=Severity X Likelihood

Risks are critical if Risk Value is ≥8

All critical risks are required to have mitigation plan. Monitoring to be done during project reviews.

Communication/Review Plan:

Review or meeting/ Reporting	Purpose	Frequency/ Stage	Required Attendees/ Addressee

Integration Plan:

Integration Product Components/ Modules	Integration Approach	Responsibility	Remarks

Project Plan

Testing, Verification, Validation and QA activities Plan

SDLC Stage	Verification, Validation and QA Activity	Responsibility	Remarks

Installation/ Deployment Plan

Deployment Type	Location	Responsibility	Remarks

Project Data Management and Configuration Plan

Data/ Records/ Configuration Item	Responsibility	Review Frequency/ Stage	Configuration Management		
			Back-up by	Integration by	Change Management/ Storage and Archival by

Any other remarks/ plans for the project

External Provider Items QA

S. No	Date	Item Name (Hardware, Software, Services)	Details	Providers Name (Client, Supplier, Service Provider, Third Party)	Providers details	Parameters checked/ verified	OK/ Not OK	Actions	Remarks for suitability of use

High Level Design

Project Code
Project Description
Client's Business Process/ Work Flow (Create/ paste or refer):
Constraints and limitations of Client's current system/ practices:
Audree Infotech's Design Goals:
Audree Infotech's Solution Architecture (Create/ paste or refer possible architectures and recommended one):
User Interface/ Screens (Create/ paste or refer):
Data Flow (Create/ paste or refer):
Checked By: Date:
Can the High Level Design be approved and project can move to detailed design phase:
Approved By: Date:

Detailed Level Design

Project Code:
Project Description:
Detailed design information:
Assumptions: Constraints: Dependencies: Client Components: External Provider's Components: Any Other:

Modules, Component, Pages, Fields and other details (Create/ paste or refer):

Module/ Component	Module Description	Pages/ Screens for	Pages/ Screens Description	Fields for the Pages/ Screens	Field Descriptions

Detailed Level Design

ER Diagram (Create/ paste or refer):

Class Diagram (Create/ paste or refer):

Data Flow Diagram (Create/ paste or refer):

Use Cases (Create/ paste or refer):

S. No.	Name	Description	Users	Pre-conditions	Course of Events	Post-conditions	Alternatives/ Exceptions	Remarks/ Notes

Interfaces Development:

External Interfaces detailing:
Users Interfaces detailing:

Detailed Level Design

Hardware Interfaces detailing:
Software Interfaces detailing:
Communication Interfaces detailing:

Security Features:

Any other information:

Prepared by: Date	Approved by: Date

Software Release Note

Date:

Name of Client:	
Project Name:	Project Code:

Note on achievement of Development Objectives:
Instructions to deploy the software:
User Manual/ Help File reference:
Limitations and known bugs:

Release Details

S. No.	Product/ Module Name	Version No.	Version Date	Release Media	Remarks

Software Release Note

Sign – Off

QA: Date	Authorized by: Date:

Document Master List

Project Code:

Project/ Process Name:

Status as on:

S.No.	Document No.	Document Name	Issue date	Revision No.	Revision Date	Submission Status to configuration team

Test Cases

Date:	Report No:
Name of Project:	Project Code:
Client:	

S.No	Requirements	Test Case	Conditions		Result	Actions for bugs	Retesting Results	Status of Bug
			Pass if	Fail if				

Any other remarks/conclusion/further course of action:

Approved by: