Detailed Technical Diagnostic Study of Software Developed by Sonata

STUDY REPORT (Final)
October 2006

Saral Services

Plot no. 17, Asha Cooperative Society
SBH Colony, Trimulgherry, Secunderabad
Andhra Pradesh (India)

Phone No: 91-40- 27743268 / 27742945

Email- info@saralservices.org; subodh@saralservices.org
Visit us: www.saralservices.org

Acknowledgments

We are grateful to Sri Anup Singh and Sri Rakesh Dubey of Sonata for giving a valuable opportunity, their insights and time to study the software. We also thank the Sonata IT team for their cooperation without which study would not have been possible. Lastly, our special thanks to Sri Ravi Narasimham of Bellwether Fund for encouraging us to take up this kind of assignment which was first of its kind for Saral and probably also for micro-finance sector.

Study Team

1. Subodh Gupta, ED Saral Services

2. G Solomon, Manager, Saral Services

3. R S K Sharma, Project Leader, Saral Services

4. Venkat Rao, Sr IT Executive, Safal Solutions

5. Subba Reddy, IT Executive, Safal Solutions

Contents

		Page
1.	Introduction	5
2.	Objectives/TOR	5
3.	Methodology for the diagnostic study	5
4.	Organogram	6
5.	Operational flow	6
6.	Requirements for MIS	7
7.	Problems stated by SONATA IT team	7
8.	E-readiness of SONATA	8
9.	Review of the documentation	8
10.	Review of database	8
11.	Review of code	8
12.	Requirement module wise review	9
13.	Implicit High Level Design	11
14.	Recommendations	12
15.	Annexures	
	Annexure – 1 Software: Sonata Microfin	13
	Annexure – 2 SONATA IT team	15
	Annexure - 3 SONATA Organogram	16
	Annexure – 4 Operational Flow diagram	17
	Annexure – 5 Screen wise code walk analysis	18
	Annexure – 6 Standards followed at Safal Solutions	20
	Annexure – 7 Technical code review of CDS	22
	Annexure – 8 General comments -code & database review	26

Detailed Technical diagnostic study of Software developed by Sonata

1. **Introduction**

   SONATA is a micro-finance company that is focusing on providing credit to poor in Allahabad district of UP. SONATA is headed by two experienced professionals – Anup and Rakesh. Both of them worked on micro-finance projects in Cashpor for many years and learnt the art of extending loans to poor on a commercial basis. Till September 2006, SONATA has disbursed more than 2 crore to 3600 clients in 6 branches. SONATA has the staff strength of 90.

   SONATA experienced the need of automated management information system. Based on the experience of Cashpor, they planned to develop software. They debated whether to outsource the development or to do it in-house. Since they had had bad experience in outsourcing the development of software, they opted to for in-house development of the software. Both Anup and Rakesh have more than five years experience in using automated MIS for micro-finance in Cashpor. They had the confidence to provide guidance for in-house software development.

   SONATA hired two MTech/MCA professionals who took the development work as consulting assignment. They promised to complete the software in three months. But in three months only few screens such as Branch, Center, Loan proposal were developed. These consultants left once they got employed in software companies outside Allahabad. Three more developers were recruited in February 06. Till now they have been able generate collection and disbursement sheet (CDS). They are also getting the past data entered in three branches. However, they are facing problems in developing the integrated accounting module. They are finding difficulties in developing modules like transfer of customer from one center to another; correction of errors in data entry.

   SONATA felt that it is taking more time to develop the package and contacted Saral Services to help resolve issues in the MIS.

2. **Objective/TOR**

   SONATA engaged Saral Services for conducting technical diagnostic study of software with the following objectives:

   1. To conduct technical diagnostic study of software developed for the MIS
   2. To suggest 2-3 alternatives to complete the MIS for SONATA

   Saral Services will attempt to answer the question – why software developed is not functioning as per the purpose and requirement and how can it work.

3. **Method for the diagnostic study**

   1. Interaction with senior management and IT team of SONATA: We interacted with SONATA team in its office in Allahabad for two days. We interacted with the team to understand – the system requirement based

on the need assessment; operational model, organizational structure, software development life cycle, IT team capability, e-readiness, Menu tree, high level logic of the software, expectation and problems in software in meeting them.

2. Technical code walk: We went through the code to understand the low level logic, problems in the code, identify code fat, technical problem with control and functions they have used.
3. Analysis of database: We reviewed the Ms Access database looking at the structure, concept of primary key definition, Indexing, and table entity
4. Prepared the screen and reports set to address expected requirement; it enabled one to access the amount of work done in terms of percentage of the total requirement
5. Identified the requirements that are not covered in the software
6. Prepared the report

4. Organogram

SONATA has a head-office based in Allahabad and it has 6 branches. SONATA plans to increase number of branches to 30 in next two years. SONATA is headed by Managing Director who is supported by Chief Finance Officer (CFO). There are three teams for human resource management (HR), Accounts and finance (A&F) and Information Technology (IT). IT Manager reports to CFO. IT team (Annexure – 2) is located at the head office.

Each Branch is headed by Branch Manager (BM) and has Business Executives (BE). There may be 1-2 Unit Managers (UM) under the BM. Each of these UM has 4-5 Business Executives (BE). Some of the BEs are assisted by Business Development Associates (BDA) for identifying customers in the target village (detailed organization chart is given in Annexure – 3). Software has been designed to suit the staff organization.

5. Operational flow

BDA and BE from a branch go around villages and hamlets and conduct survey and collect data on families and their asset base. Based on the survey a list of customers is prepared who want to join a center. Center is a group of 10-20 customers. Based on the data center recognition test is conducted by the BE and BM. Using the data from the CRT customer database is prepared. Loan proposal is prepared for 5-10 customers from the center. These proposals are then sanctioned and loan is disbursed to them at the branch through cheque. BE then goes for collection of repayments at the center meeting on weekly basis. Center leader may collect the predetermined amount from the members and deposit it in the Bank. The deposit slip is then collected at the center meeting and given to Branch Manager. Members who do not pay are followed up for repayment for some weeks. Later such defaulters are dropped from the center. (See the Operational flow chart in annexure – 4)

6. Requirements for MIS

It was surprising to note that there was no written list of requirement for which MIS was being developed. There has been discussion on the need assessment and requirement with the IT team and senior management. Mostly they were given verbal instructions and some time small written slips. In the absence of any list, we built the list through a consultative process. We facilitated a process of documenting the requirement. At the end of the process, we prepared a list of requirement (See Annexure – 1). Main objectives were first articulated and then the requirements were listed and subsequently requirements were divided into two phases.

The current software was reviewed with respect to these requirements. Only 25% of the work seems to be covered from the requirements of first phase. No work has been done from phase two.

7. Problems stated by IT team

1. Version management is a critical issue in any software development. There is no maintenance of versions in Sonata. There is no proper documentation on the software design. There are 2-3 copies of codes. At the branch level if software is being used for data input they do not use the executable binary code there. They use the source code which they compile and use. If there is a run time error they correct it and use it. HO has another version of the codes. The codes that are corrected at the branch have not been integrated. No discipline of version management is being used.
2. Base line data form has field category where caste and religion both have been used together. This needs to be corrected.
3. Currently, village comes from user entry. It is not calculating the percentage target families covered. Physically they are not collecting the data on 100% families therefore it is not possible to access the percentage market they are working with. In the Target ID, Village name needs to be taken from the census data.
4. Design discipline is required to be followed. Software development in SONATA did not follow the standard software development life cycle discipline. There is no design document such as screen design, screen-table mapping, logic state diagram, flow-chart and database design document. In absence of these it will be impossible to achieve modularity and scalability in the module.
5. Loan product only works for flat rate of interest. There is no logic for declining interest calculation.
6. Printing CDS is problem at branch level. The line alignment is often a problem. They are required to print one sheet per center per week. There is a need for finding a solution which is more reliable.
7. Run time error is a normal occurrence which is a problem.
8. You can print the latest report only after you close the day operation.

8. E-readiness of SONATA

E-readiness at the level of organization is level of readiness to take up automation task efficiently. In the case we need to understand whether SONATA is e-ready for implementing automation of MIS or it is ready for developing automation software. Top management is fully ready for using software for MIS. They have already invested approximately 4 lakh on the development of the software. They are interested and feel that they require suitable software. They have an IT team consisting of four persons (See annexure – 2). This team makes SONATA e-ready for implementing MIS in the organization of substantial order. However it does not make it ready to develop micro-finance software in-house.

For developing software, e-readiness requirement is of higher order. It requires people with understanding of software development discipline. Quality of software design improves with technical education and experience. Especially in software, technical documentation plays a vital role. It is critical especially, when IT staff is mobile and we need to track the logic of the code and databases. SONATA is not e-ready for software development.

9. Review of the documentation

There is no written document other than Users' Manual. It does not have design documents to practically do standard testing as to understand whether the software has achieved as per the design. Users' manual is fine. It currently explains the user data entry screen. However, there is room to improve the user manual also, especially from the point of sequence of tasks required to get a report.

10. Review of database

1. Database is not normalized. Therefore there is lot more repetition of data. Such design requires much more high order of documentation for making any modification. In such scenario, the action such as modification, delete, rectification become much more complicated.
2. There is no documentation on the database design. Therefore any change in the database is fully based on the memory of the developer. It is vulnerable to changes in the staff.
3. Data size is unscientific. All the Ids are of 50 characters. Thus there is waste of space.
4. Tables are not indexed properly.
5. Some table primary keys are not defined.
6. Data tables are so designed that it is hard coded for number of loans one person can take at a time. Hard coding at the level of database is highest order of constraints during upgradation. (See Annexure – 8 for details)

11. Review of code

a. No comments in the coding. In 350 lines of coding in Day initialization, there is no comment. It makes the coding weak from the point of review and modification. In 1918 lines of coding in CDS, there are only 2

comments. In Safal Solutions, we follow at least one comment per 30 lines of coding.

b. There is a scope to improve User friendly messages. Most of messages are like 'please fill this field' which is not sufficient for a message.
c. There is no System generated id for Branch-id, unit-id, Staff-id etc. For example in the Client screen after selecting center id user has to enter-client id which it is saving without entering center id, this will give a problem while retrieving center data. User entry of these Ids has to be done carefully.
d. Screens are not functioning properly because of lack of proper validations that are missing, for example when we open the Center Screen, data is loading into list view. While loading into list view if record set retrieves null values without checking that null value it is trying to store into list view this causes runtime error. Such errors are there in most of the screens.
e. In each screen, while using action command, it does not do enabling and disabling using a common function. Therefore there is repetition of codes. In such cases, correction is more painful and not at all full-proof.
f. It is not required to store AccountCode, opening and closing balance in the voucher table. They can be always calculated. Why should one store debit and credit in this VoucherParticular table when as we have already done so in AccountVoucher. Such redundancy, if not handled properly, can lead to errors.
g. Generally we can handle Save and Updating (Modify) functionality in Save button itself by checking conditions so that we can save the code, and avoid repetition of coding. We can especially save codes in most of the masters.
h. While saving record code is checking whether record already exits in the database or not and for that there is no need to select all the records of table but only select record with where condition ex: centerid=txtCenterID if rs.eof then save it else do not save it. There is while loop in the code that can be replaced with direct query.
i. There are several examples of hard coding in the code – 1) While saving CDS data it assumes 20 records. All actions are for 20 members. Such hard-coding could be removed by using variables. (More detailed review note is available at annexure – 7 and annexure – 8)
j. There is lack of testing. We encountered several run-time errors while doing the code walk.

12. Requirement module wise review

We tried to match the requirement and the module prepared. Following is the result.

Sn	Requirement	How it achieved	Remark
1	Need to understand the size of market, its penetration and later use to capture the impact	Target and Client screen	It is not using census code therefore does have total village data.

	also. Capture the ongoing Base line data- need to link with the census data.		Survey is never done for all in the village. Thus requirement is not achieved.
2	Percentile of the objective met - % of the BPL penetration against the Base line/ Census data	Status compares with the families surveyed only	The picture is not correct picture as the requirement is not met
3	Status of the Branch/ Unit/ Center/ Members information	Master screens – Branch, Unit, Staff, Center and Client	Requirement met (run-time errors to be removed)
4	Various Loan product and its pricing - Upfront deduction, Flat Interest rate method, Payment frequency	Loan product screen	It is hard coded, it does not address the pricing issue. Flat calculation is hard-coded and payment frequency is also hard-coded. It will only take weekly frequency of repayment. Thus requirement is not met
5	Purpose vs. risk management	Activity master and reports	Requirement met
6	Tacking of the various loan product	CDS and reports	Aging analysis is yet to be done. Requirement met partially.
7	Liquidity status report		Not done
8	Daily performance review	Status of client, center	Requirement met at the Branch level
9	Daily track of the activity- Cheque issued, center Advance collection, Collection Disbursement Sheet		Requirement met partially.
10	Staff performance	Staff status and	Requirement met

		history	
11	Various reports; Demand chart, Loan ledger, Center Meeting schedule, Clients and the center credit history, No. of active member, Active loan client, Arrears report, Aging of the arrears, Company report, Operational Activity report on the various date range	Reports	Three reports yet to be done – Aging of the arrears, Company report, Activity reports on date range – Requirement met partially.
12- 17	(See annexure – 1)		Not done

Out of 17 requirements, in phase – 1, only 11 are done partially. Difficult modules such as change of schedule, rectification, integration with accounts are yet to be done. In our estimate software is only 25% complete for phase – 1.

13. Implicit High Level Design

Although not stated it appears that Sonata has adopted the following High Level Design –

Design	Advantage	Disadvantage
no distinction between Branch and Head-Office modules	Only one design	Problems in consolidation if there is no unique code
At the time of day begin, copies the summary data of all those customers whose meeting is due. Changes are made on that if there is transaction and only at the time of day close these transactions are added into the main transaction tables.	Quick transaction response	One can take reports with the latest entry only after the day close. If one has the meeting day on a holiday, one has to change the date of the meeting temporarily, as design can not handle such scenario.
Summary data are maintained at all levels such as Branch, Unit, Staff, Village and client	Quick report generation	Rectification is complicated and always leaves a doubt
SONATA will never give more than two loans to one person	Quick report generation	Slows down query and leaves no room for increasing the number of loans to a client

Such a design strengthens the software in certain areas but it also provides constrains in growth.

14. Conclusion:

Currently, SONATA does not have the capabilities to develop in-house software. They went for in-house development because they had poor experience of out-sourcing and there was no experienced vendor available in Allahabad to develop such packages. The modules that have been developed are essentially simple. Although low grade coding, they are working with the present design limited to the master screens and CDS screen. Ones that are difficult are still not done. They have achieved printing of CDS. CDS is not fully working; it is giving error in calculation of arrear. CDS does not compute arrear correctly in case of two contiguous defaults. Work done is 25% of the Phase – 1 requirement. Main problem is in the overall and database design. Unless they are corrected MIS will not achieve the objective of meeting the requirements. If the software has to work, we need to change the design of the database. Once the database design is changed then the current code will not work and in such event, it would be advisable to do the coding again.

In the current scenario, there are two options available, namely:

a. Out-sourcing the design and project management: SONATA should out-source the design and project management. SONATA can, if it wants, continue to do the coding for the development of software. With the current policy of IPR, not many vendors will come forward for such assignment. In this case, SONATA needs to consider sharing the IPR. Then the vendor/consultant will take the complete responsibility of delivering the result. The vendor/consultant will use the developers at SONATA for coding. Since the current coding is of little use, they are also likely to develop the code all over again.

b. Out-source complete software development: Since the problem is basically at the design level, there is likelihood of little improvement if Sonata wants to continue with the present code. Complete outsourcing will rest the complete responsibility with the vendor to supply quality software that works and gives all the reports for tracking loan and insurance.

Cost implications:

Cost implication in option I is higher as one will be developing things from scratch. SONATA will have to bear the cost of the consultant and also of the coders. There will be delay if the coders leave, a possibility one cannot rule out. It will take 4-6 months to complete. Total cost may be of the order of 10 lakh.

In option 2, cost will be reasonable as now the market is competitive. Vendors may have some software ready and they can customize it to suit to the requirement of SONATA. It will take 45-60 days in completing the data entry screen, so that data entry can be started. It will take additional 60 days in completing the integration, reports, accounts and HO modules. Cost of this will be about 6 lakh.

Annexure - 1
Software: Sonata Microfin
(Note has been prepared by Sonata team facilitated by Subodh Gupta, ED, Saral Services on 6 October 2006 at SONATA office in Allahabad)

Objective of developing MIS:

1. To get timely, Accurate, Information
2. To build a paper less company so that it saves the time of the staff to do more creative work
3. To generate the report as per statutory requirement
4. To track the loan & Insurance

System requirement based on need assessment:

Phase - I

1. Need to understand the size of market and its penetration and later to capture the impact also.
    - Capture the ongoing Base line data- need to link with the census data.

2. Percentile of the objective met
    - % of the BPL penetration against the Base line/ Census data
3. Status of the Branch/ Unit/ Center/ Members information
4. Various Loan product and its pricing
    - Upfront deduction
    - Flat Interest rate method
    - Payment frequency
5. Purpose vs. risk management
6. Tacking of the various loan product
7. Liquidity status report
8. Daily performance review
9. Daily track of the activity- Cheque issued, center Advance collection, Collection Disbursement Sheet
10. Staff performance
11. Various reports;
    - Demand chart
    - Loan ledger
    - Center Meeting schedule
    - Clients and the center credit history
    - No. of active member, Active loan client, Arrears report,
    - Aging of the arrears
    - Company report
    - Operational Activity report on the various date range
12. Clients switching
13. change of the center schedule
14. Rectification of the errors

15. Building up a integrated software where the date automatically transferred accounts module – Balance Amount, Interest, Due date, Payment due, Hypothecation of the books
16. Data updating process (Day wise upsizing of the data)
17. Data security

Phase II

18. Financial statement on the various format

19. Data consolidation
20. Change in Interest rate and Declining method of interest calculation
21. Yearly closing process
22. Loan reschedule and principal collection
23. Mapping of the village in the center map for the purpose of the surprise visit
24. Repayment schedule for the funding agency
25. Client Insurance
26. Data Synchronization (Import the data from HO to Branch)
27. Change of database platform

(Underlined requirements are not yet covered in the existing software)

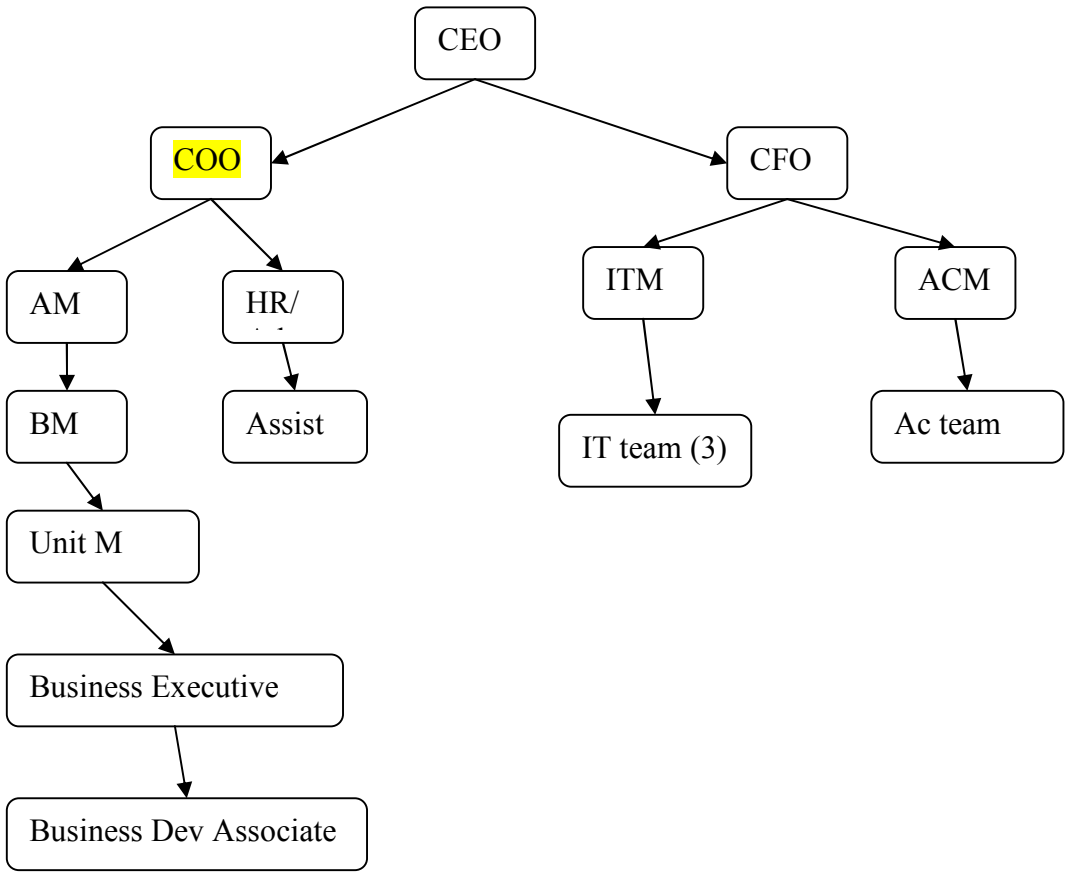
Annexure – 2

SONATA has an IT team consisting of

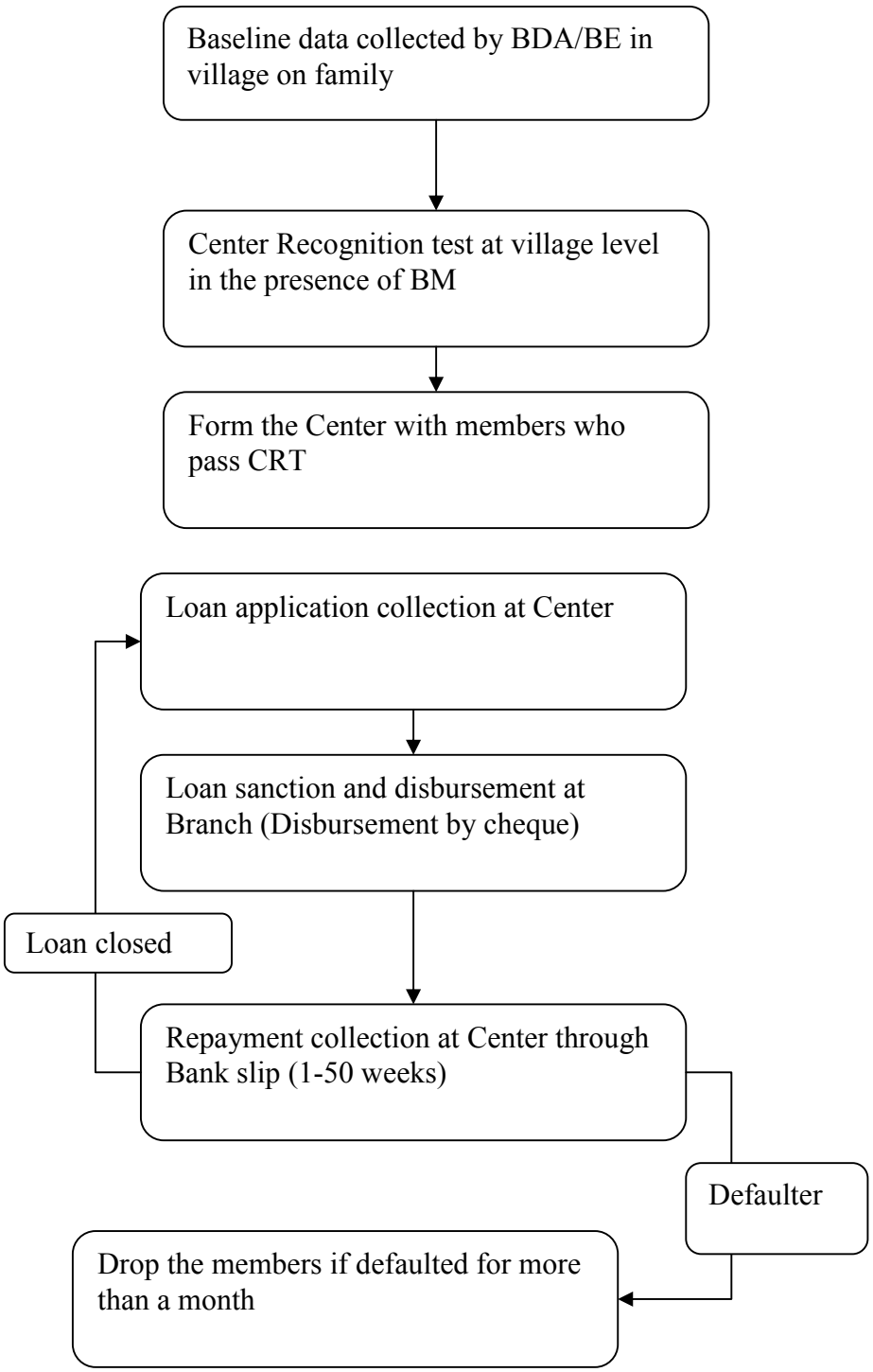
1. Sanjeev – who is currently appearing for MCA final examination from Allahabad University
2. Abhishek – who is currently doing MCA
3. Yadav – has completed MCA
4. Santosh – who has 7 years of experience of testing software in Cashpor

IT team is located at the HO and is working on the software. Because they are studying for MCA or are MCA, they know the SDLC. But they do not have a clue of how SDLC is practically. They only know how to create table in Access and write code in VB. However, the quality of code requires great improvement.

Annexure - 3
ORGANOGRAM



Annexure – 4
Operational Flow



Annexure – 5

Screen wise code walk analysis

Screen Name	User Interface related	Validations	Functionality	Database related
Branch	Ok	Using Last focus Event causing Infinite loop	Branch id should be auto generated code	No Separate Master tables for location state/district/Block..
			Same places code is repeated ex: same validation checks are there in save & update buttons	In Branch Table transaction related fields should be separated
Unit	Ok	Specific validations require for each field	Unit id should be auto generated(Bra-id & auto-num)	In Units Table transaction related fields should be separated
Staff-BDA	ok	Meaning full messages should be present	Staff-id should be auto generated	Staff Performance should be measured with based on transactions, generally we should not store in same table we have to use temporary table while generating reports
Target	Ok	Navigation handled properly for Enter key, not for mouse clicks	Ok	
CRT	OK	Ok	OK	MRT table doesn't f following normalization. If the number of members allowed per center increases beyond 20 problems will arise. And also null values are unnecessarily stored
Center			While loading into list view not handled null values if null is retrieved will it stored into list view control?	No primary keys are maintained which is causes duplication and allowing nulls
			For Generate ClientID your following fixed approach(for 3digits) if clients crosses more than	

			1000 for center will it work	
			While saving center your checking whether center is already exits or not for that not required to select all the records of center, select record with where condition centerid=txtCenterID if rs.eof then save it .here you are doing unnecessary traverse the data using while condition	
Client			Client Id ->after selecting center id user has to enter-client id it is saving without entering client id which will give problem while tracking clients data	In center table lot of fields are stored together without normalizing which will cause to data redundancy
Loan Product	Units(.Rs) are not mentioned for data entry		Product ID should not be user entry,	
Loan purpose	Ok	Ok	While generating LoanPurposeID you should append LoanPurposeGroupId so that no need to store LoanPurposeGroup In Loan purpose table	Separate master table required for Loangroup and Loanpurpose so that we can track loans LoanGroupWise and purpose wise
Daily Data Entry			CDS details table not getting updated when principle and interest has been paid	
			From cdsclient loan subscription is getting updated and from loansupscription cdsclient is updated this is not a process oriented approach	

Annexure - 6

Standards are following by Safal Solutions across the Packages

Design standards for Forms

1. Form orientation –
   - ✓ Form should be centered in screen.
   - ✓ Maximize, minimize buttons should be disabled unless required specifically.
   - ✓ Form caption (if any) should match with one shown in menu selection.
2. Controls –
   - ✓ Tab sequence to be maintained between controls in order of top-bottom and left-right.
   - ✓ Controls with input as mandatory should have back color "H80000018"
   - ✓ Controls with input as optional should have back color "White"
   - ✓ Controls used for displaying values should have back color "HFFC0FF"
   - ✓ Labels -- Font name – "Verdana"; Font style – "Bold"; size – 9; color – "Black"
     Buttons – Font name – "Verdana"; Font style – "Bold"; size – 10; color – "Black"
     Textbox/selection box – Font name – "Verdana"; Font style – "Regular"; size – 10; color – "Black"

Naming conventions

1. Variable declaration
   - ✓ Any variable declaration should follow "Camel casing" e.g. conAcc i.e. the first character of the first word should be in lower case while the consecutive words should start in Upper case and remaining letters of the word should be in lower case.
   - ✓ Global variable should start with "g_" (offline)
   - ✓ Form level variable should start with "f_" (offline)
2. Controls
   Every variable should have meaningful name with convention being followed. Below are some examples for VB.
   - ✓ Buttons – btn
   - ✓ Textbox – txt
   - ✓ Listcontrols – lst
   - ✓ Combobox – cmb
   - ✓ Label – lbl
   - ✓ Option buttons – opt ; Radio buttons – rdo
   - ✓ Frame – fra
   - ✓ Form – frm
   - ✓ Reports – rpt
   - ✓ FlexGrid-- flx

Others should follow conventions specific to software language being used.

Design standards for Reports

- ✓ Report heading should be same as shown in relevant menu tree option and should not have any Caption.
- ✓ Should be displayed in 'Maximum' mode

- ✓ Font specifications --

  Report heading →
  Font name – "Garamond"; Font style – "Bold" ; Size – 14
  For labels →
  Font name – "Garamond"; Font style – "Bold" ; Size – 12
  For displaying data values →
  Font name – "Garamond"; Font style – "Regular" ; Size – 11

Coding related standards and conventions

1. Comments
   Following structure of comments should be followed for each routine in the order specified below —
   - ✓ Creation date:
   - ✓ Created by:
   - ✓ Purpose:
   - ✓ Output:
   - ✓ "Revision history"
     - ■ modification date :
     - ■ modified by:
2. Any global routine should be written only after discussion on a common place with other developers and technical lead.
3. All the variables instancing any memory resource should be disconnected/deallocated at the end of the routine/form.
4. Instructions should be written so that they fit into the screen and do not need dragging to see the whole instruction. In such a case, the instruction should be split into the next line.
5. Instructions should be indented accordingly so that they clearly speak of their hierarchy level.
6. All the local variables required in a subroutine should be declared at the beginning of the subroutine only (Even though the editor gives facility to declare them any where in between).
7. Any Sql query written to access data from database should request only the required columns and "select * from" should be used only if all the fields are really required to be retrieved.

Database design standards

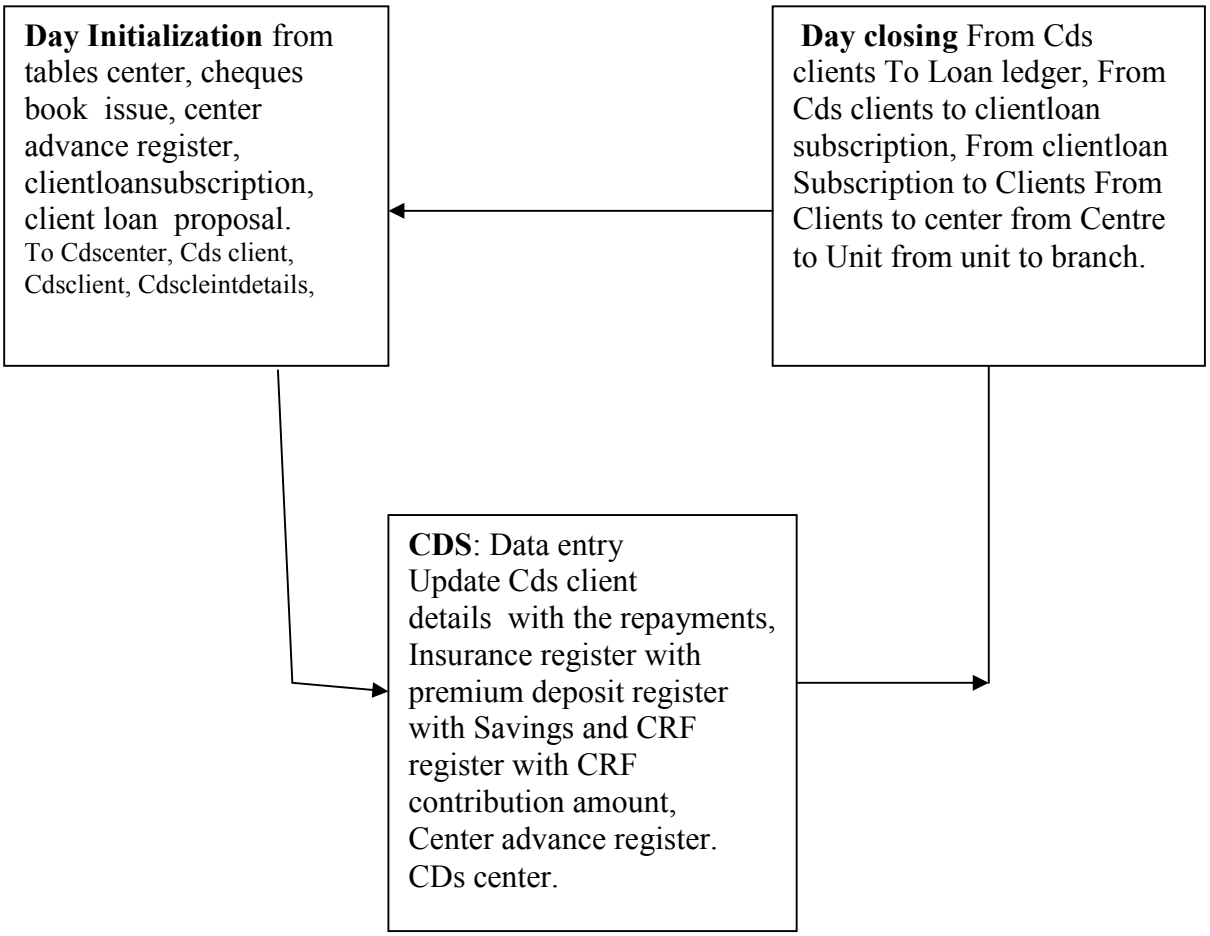
1. Master tables should be named as starting with "Ms"
   Transaction tables should be named as starting with "Trn"
   System tables should be named as starting with "Sys"

2. Each field should have its description mentioned by the designer.

Annexure - 7
Technical code review of SONATA Microfin software

Code review case: The daily transaction cycle is defined as a set comprising of Day initialization, CDS data entry, Day Close processes. This Daily transaction is taken as a case for in-depth code review purpose for Sonata to find the errors on logical and technical front.

Existing logic flow:



Reporting of the in-depth review follows a format of describing in sequence of – 1) Name of the screen/ report, 2) Symptom of the problem, 3) Diagnostic comment and 4) How to correct them. The following errors were found out while doing the code review for the daily transaction cycle:

1. Module name - daily transaction cycle

1.1 Symptom: When one accessed client no 1 there is no problem and when one accesses client 2 after that there is no problem . Now when one tries to access client 1 again there is a run time error.

1.2 Cause: This is caused because txtInsurancePolicyName(Val(cbocdsclientno.Text) -1). Enabled value is false when you trying set the focus to txtInsurancePolicyName(0).

2. Module name: In Day closing the following errors were observed .

2.1 Symptom:The number of MissedInstallments is not getting updated due to the following reason.

2.2 Cause and solution: One should pick up from record set or get it from stored variables but neither is happening while checking condition at the time of updating ClientLoanSubscription.

[Code Snippet:

```
If ClientLoan1PrincipalInstallment > ClientLoan1PrincipalCollection Then
Module1.cnn.Execute "update ClientLoanSubscription set MissedInstallment =
MissedInstallment + 1 where ClientLoanID = '" & rsCDSClientsDetails!ClientLoan1ID
& "'"
End If ]
```

2.3 Effect: Due to this reason the number of Installments missed =0 even if there exists arrears

3 Module name –Day close

3.1 Symptom There is an error in Numberofinstallment in arrear function for loan1. In case of loan2 the value is being calculated correctly.

3.2 Reason and solution: Existing funtion

```
rp!NumOfInstallmentInArrear = Val(Int(rp!PrincipalAmountInArrear /
rp!PrincipalInstallmentAmount) + 0.99)
```

Correct Function

```
rp!NumOfInstallmentInArrear = Int (rp!PrincipalAmountInArrear /
rp!PrincipalInstallmentAmount + 0.99)
```

3.3 Effect : In the above logic, it will always result in installment arrear one less.

4. Module name – Day close

4.1 Symptom For loan2 the amount in arrears is being calculated wrongly.

4.2 Cause and solution:

Existing logic(code).

PrincipalAmount in arrear is = Principalamountdue- principalamountoutstanding.

Correct logic is

PrincipalAmount in arrear = principalamountdue - principalamountcollected.

5. Module name – Day close

5.1 Symptom: There is an error in entering the data into loan ledger. That is the principal arrear is being entered wrongly.
5.2 Cause and solution:
Principal arrear = PrincipalinstallmentAmount- Principal Collection amount.
It should be (Sum of PrincipalinstallmentAmount)-(Sum of Principal Collection amount) for the given LoanID.

5.3 Effect the cascading effect of this is shown in an example. In sectionII

6. Module name: Day close
6.1 Symptom :In calculating the number days in arrear for loans there is a mistake happening.
6.2 Cause: The number of Days in arrear is = Present date – Arrear coming date.(This is correct)
If the principalamountdue>Princpalamountcollected the the current opening date becomes the arrear coming date.
If the client does not repay in the successive installment then the Arrear coming date is reset once again to the current opening date which is wrong.

6.3 Solution:
Instead the logic should be
If the arrear date is Not null and If the principalamountdue>Princpalamountcollected then don't change the arrear date else if its null then the arrear coming date= current opening date.

6.4 Effect:
These above mistakes seem to be small but they have greater implications because

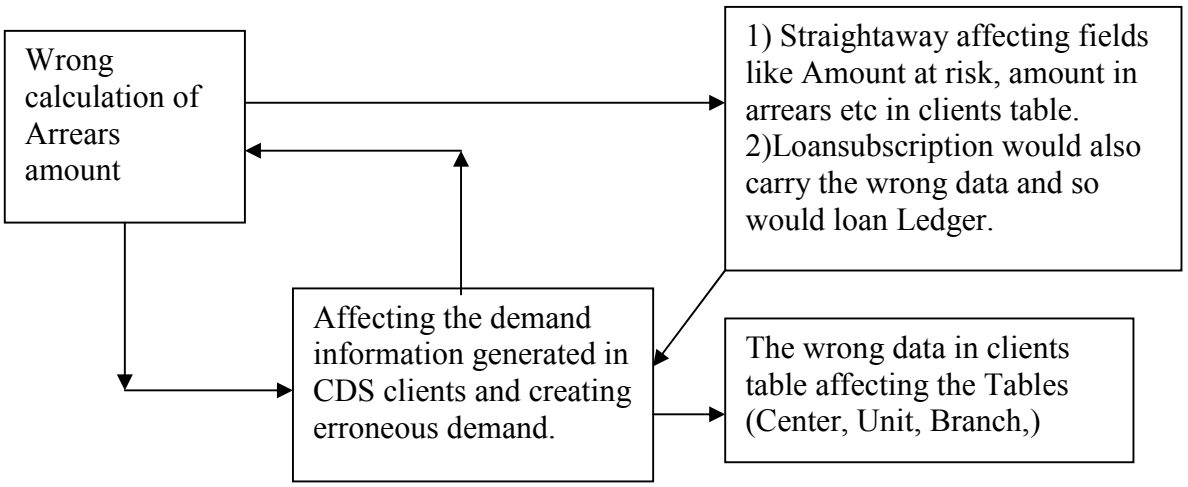
- Once the number of installments in arrear goes wrong then fields depending on it like - Risk coming date, Numloan at risk F1weekarrear ,F2,F3,F4,F5 plus loan in arrear can go wrong. If this happens in client table it shall carried forward to center, unit, branch,

- If Amount in arrears goes wrong as it is happening with the second loan. Amount at risk can go wrong. F1,F2,F3,F4,F5plusamount in arrear can go wrong. Once the amount in arrear goes wrong the subsequent repayment demand going toCdsClient details will go wrong once the repayment data error goes wrong the entire loan cycle will go wrong and the database will be corrupted by the wrong data and will go beyond repair within no time.

- If the number of days overdue goes wrong then it will fall in the wrong slot of no of days overdue in client table and this mistake will carry forward into center ,unit, branch.

- This cascading effects are due to the fact that the 3rd rule of normalization is not implemented that is Transitive dependencies should be eliminated.Eg: The

amount in arrear can be calculated from loan ledger rather then using the roundabout way of updating ClientLoansubscription from Cdsclients during day closing andthen updating Cdscleints from Loansubscription during the day close.



Annexure - 8
General code/database analysis

1. Design wise this model of design caters to fixed days. If repayments are made on meeting date that happens to be a holiday (eg Aug15) then there would be problem in the data entry. To accommodate this design itself needs alteration.

2. In InitializeVariable procedure first declaring arrays and again clearing array variables. This is not required when you declare an array fresh memory block will be created There is no need to clear it .

3. We should not use memory variables unnecessarily. We do not require to store 20 clients bank information in array variables when we have that information available in CDSClients table.

4. We are storing all the 20 client's information at the same time and the coder has used lot of controls that consume lots of memory and makes coding complex. Instead of storing entire 20 clients information at the same time, one can write a procedure which will get the data from database for that particular client and fills the data into CDS screen, based on the value selected (change) in cbocdsclientno.

Problems at database:

- Same code is repeated for two loans in DayClose while updating ClientLoanSubscription and also at many places. Instead it can be done in one place using loop but this data base is designed in such way that we can not save the code We cannot give more than two loans per client because it is fixed at the database level. Instead of repeating same set of fields for Loan1, Loan2 if you store a loan-id as a part of primary key we can store any no of loans for client but this requires design change.

- In ChequeIssue table CenterName, ClientName, StaffName, UnitName, BankName not required to store when id's are stored. If required one go to particular master table get it from the that master table using respective ids.

- Ms Access database is alright in the Branch but we certainly need stronger database in the head-office. There is a size issue and comfort in accessing the database. We recommend MySQL database in place of Ms Access database in Branch and HO. This strategy enables one to use database internal features of achieving and synchronization.