

# **Comarch Customer Feedback Management System**

## **Project Plan**

# Table of Contents

<b>I Introduction</b>	<b>2</b>
a. Abstract	2
b. Major software functions	3
c. Management and technical constraints	3
<b>II Objectives</b>	<b>4</b>
<b>III Background</b>	<b>4</b>
a. Problem	4
b. Solution	5
c. Value proposition	5
d. Initial market	5
e. Competition	6
f. Business model	7
<b>IV Project Schedule</b>	<b>7</b>
a. Development cycle	7
b. Tasks	7
I Analysis	7
II Technical project	9
III Development	9
IV Testing	19
V Deployment	23
c. Task dependencies	24
d. Timeline	25
<b>V Staff organization</b>	<b>26</b>
a. Team structure	26
b. Team member description	26
c. Accounting and role allocation	26
d. Reporting and communication	27
<b>VI Appendix</b>	<b>28</b>
a. Credit Card Loyalty	28
b. Auto Insurance Loyalty	28

## **I. Introduction**

### **a. Abstract**

In a new and fast growing customer driven economy, successful businesses are relying on customer feedback to provide the best products and services. Customer Relationship Management (CRM) and Customer Loyalty Management (CLM) tools and services are being implemented in businesses to aid in increasing customer satisfaction, maximizing profits per customer, and cheaper, more effective marketing. Comarch Customer Feedback Management (CCFM) solutions and software will help businesses in all markets develop targeted questionnaires to give businesses a better understanding of their customers' needs, lifestyles, and culture.

CCFM is customizable user friendly software developed for people of any skill level to design questionnaires for customer feedback, market research or any need for gathering customer data. CCFM's goal is to be an effective software solution for people who need effective questionnaires even those who need to survey thousands of people and require data that is easily categorized.

Businesses in today's marketplace and global economy are competing to bring the best products, services and solutions to their customers. Businesses need to keep customers as long as they can to save money, and earn more off return customers and customer loyalty. Most businesses implement a CRM and CLM department, and CCFM's questionnaire building software makes those departments more effective.

### **b. Major software functions**

CCFM is a simple, lightweight and user friendly software. People lacking in computer programming skills will enjoy CCFM's ease of use. CCFM will be interoperable with third party business intelligence analysis software. CCFM will contain built in wizards and templates making it even easier to use.

Users of CCFM have a variety of question types to work with including multiple choice questions, open ended questions, and rating scales. These questions will develop into a decision tree model, so every user gets a unique questionnaire based off the answers they give.

CCFM contains a central database for information storage. Users can upload questionnaires to aid others in their field create effective questionnaires. The database will contain a search engine and forum for users to post questions, tips and other advice for users.

### **c. Management and technical constraints**

CCFM will be developed within Comarch. Comarch is a rapidly growing international company based in Krakow, Poland, but with offices all over the world including Europe, the Middle East, South America, and three offices in the United States. Comarch designs software for all types of businesses. Comarch was founded in 1993 by Prof. Janusz Filipiack and embraces innovation and an entrepreneurial spirit that has made it successful since its

foundation. In 2006 Comarch hired 600+ employees and continues to open new offices worldwide including Chicago in 2007.

Comarch's CRM and CLM solutions have helped many businesses in Poland including Bank BPH, Proctor & Gamble, Unilever, and Sony Poland to name a few companies. Comarch's CRM software, AurumCRM, is a program containing a Marketing campaign management system, effective sales management system, customer loyalty systems, and a content management system.

Once the opportunity assessment has been completed CCFM will go into development. Time allowed will be 100 man hours in 4 months to develop the software. Money allocations will be 500,000 USD. While these seem short, there is enough confidence in Comarch's programmers, analysts and designers to develop the java based program within the allocated amount of time.

## **II. Objectives**

26/June/2007 – Project Plan due

- outline and lay out a roadwork for future plans
- analysis of the project

6/July/2007 – Midterm Report

- Progress report halfway through the project time
- 15 minute presentation with 5 minutes of Q and A.

23/July/2007 – Poster due

- Poster to be used as display for IPRO day presentation

24/July/2007 – Abstract and Brochure due

- Also to be used for IPRO day presentation

25/July/2007 – Team minutes due

- Final report due

27/July/2007 – IPRO day, final presentation

## **III. Background**

### **a. Problem**

Customer surveys are an effective way to understand your customers when used properly. Many consumers however do not always take these surveys seriously. Websites like surveymonkey.com and surveybuilder.com offer online survey builders that offer free development, but charge for the analysis. These surveys create simple straight line questionnaires, and are web-based. Businesses can develop their own questionnaire programs with the assistance of a programmer, a higher cost, but the programmer can design the exact questionnaire for the business as they like. Straight line surveys will bring in a wide range of data difficult for companies with thousands of customers to sort through and analyze.

Problems with ineffective surveys become ineffective customer service, mixed signals in feedback, and a CRM department with a bad focus. While most businesses believe they have the right CRM and CLM solutions, with good ideas focus on purchase quantity, number of transactions and spending amount rather than delivering quality products and services to suit the customers who bring them the most money. Effective feedback is necessary to provide the ideal products and services that customers expect. Loyalty should also come from the business to the customer as well as from the customer to the business.

## b. Solution

CCFM is a questionnaire building platform that is user friendly, customizable, and flexible. A simple interface makes it easy for users of all technical skills. CCFM features a decision tree style of questionnaire for easy to analyze data when millions of people will be taking the questionnaire. Decision tree models give unique surveys to takers based on their answers.

The CCFM database will contain surveys done by users in all professional fields and allow users to network amongst others in their field. People can research other firms and groups in their field and gather valuable data. Some data can be kept confidential for businesses to secure customer data.

## c. Value Proposition

The result of effective questionnaires and feedback is valuable data to any business with customers. CCFM will be an effective way to retain loyal customers, and maximize potential profits per customer. Loyal customers bring more money to the business, and more loyal customers mean greater profits.

Most businesses with customers will fall within the 20:80:30 rule. 20% of your most loyal customers will generate 80% of your revenue, on the contrary, 30% of new customers make up for almost 50% of your costs, advertising and attracting new customers. Companies need to focus on maximizing profits from loyal customers, and minimizing costs on new customers. For airlines 20% of frequent flyer members generate 80% of an airline's profit, additionally 2% to 4% of frequent flyer members generate 33% of an airline's profit (Charlton, 2004).

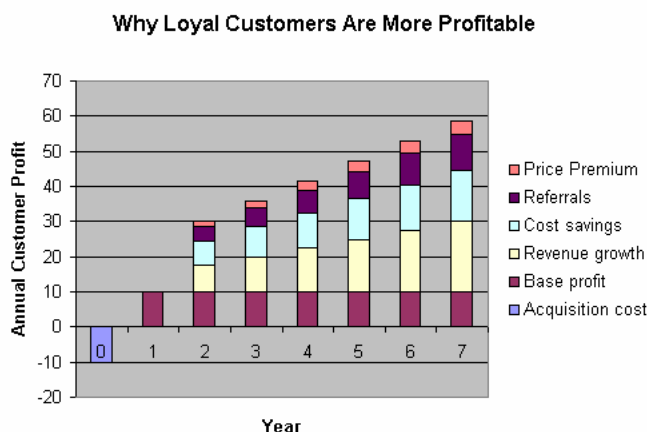


Figure 1 (Poche, 2006)

CCFM is simple and user friendly, anybody with enough skill to turn on a computer and install software can use it. It's lightweight drag and drop interface simplifies its design. Decision tree structured questionnaires makes survey data easier to organize, categorize, and analyze.

Surveys help marketing divisions make informed decisions; it is easier and more effective when to hear primary data from potential and current customers. These surveys can be used by businesses to determine the spending habits of its customers, knowing what consumers are buying helps companies put the right products on the shelves, and offer the right services to keep them ahead of the curve.

CCFM's database of information will contain a wealth of knowledge of all its users who upload their surveys, Comarch employs smart business professionals, but the body of knowledge of CCFM users will bring more valuable data.

#### **d. Initial Market**

Customer focused businesses that require follow up:

- Retailers
- Medical Questionnaire/follow up
- Emergency Reporting
- Job Interviews
- Helpdesk and Troubleshooting solutions
- Government surveys (US Census report)
- Educational/class exams
- Frequent Flyer Programs
- Telecommunications
- Restaurants
- Hotels
- Websites
- Technicians

While the retail industry can develop products based around customer feedback, the service industry thrives off feedback.

Current Comarch CRM solution customers

- Bank BPH
- McLane Polska
- ECR Polska
- Real Group Metro
- Auchan
- Unilever
- Jeronimo Martins Distribution
- Proctor & Gamble
- Sony Poland
- Carrefour
- Stella Pack

Businesses looking to collect customer data would benefit from CCFM's ease of use, and many layers to export data to.

## e. Competition

Other companies are also developing innovative solutions at obtaining customer feedback.

Microsoft TextMiner – Microsoft tech support was receiving up to 50,000 emails a week, to fix this problem they came up with TextMiner, a program that sorted through emails and looked for key words and filter through the emails, allowing more time for analysis and less time reading individual emails.

Fizzback – Fizzback focused on generating instant feedback at the touch point using mobile technology. If customers were unhappy with an experience, they could instantly SMS that into the company. This is an effective tool to also study the psychology of customers at the time of transaction.

	Stamatrix Composer	Surveyz!	Microsoft SharePoint	Vovici EFM	Fizzback!	Right Now	Comarch
Example templates	x	x		x		x	x
Wizards				x		x	x
Personalized invitations	x	x		x			
Thank you notification	x	x		e-mail			x
Decision - trees	x	x	x			x	x
Database	x	x		x	x		x
Web-based survey	x	x	x	x	x	x	x
E-mail Survey			x		x	x	x
Paper Survey	x		x				x
Correlations		x	x	x		x	x
Export data	x		x	x		x	x
Team Collaboration	x			x			
Import/export templates	x	x	x	x		x	x
Mobile devices					x		x
Rewards program	x	x					x
Data Consistency Tool							x
Random function		x			x		x
Support for third-party tools		x		x			x

## **f. Business Model**

CCFM will sell its rights to use the software to develop their own questionnaires. Software is a form of intellectual property that can be copied and distributed.

## **IV. Project schedule**

This section contains the development cycle, tasks, task dependencies and the timeline for the project.

### **a. Development cycle**

Following is a chronological list and description of all the stages involved in the development of the application. This project will follow the waterfall model of development.

#### **1. Analysis**

This phase includes a feasibility study of the project and a detailed business analysis. Also the key functionalities of the solution will be determined in this stage.

#### **2. Technical project**

This stage of the project will concentrate on determining the technical aspects of the project such as the technologies to be used in the solutions. Also detailed technical functionalities of the project will be determined in this stage.

#### **3. Development**

This phase consists of building a fully functional prototype of the solution.

#### **4. Testing**

This stage will focus on testing and validating the prototype to ensure that all the specifications have been met.

#### **5. Implementation**

This is the final stage of the development in which the solution will be deployed into the real world with either Comarch or one of its clients.

### **b. Tasks**

This section contains a list of all the tasks that need to be performed in order to develop a fully functioning system.

## **I Analysis**

### **1. Project definition**

This stage will define and provide a top-level overview of the project,

1.1 Project overview

1.2 Scope of the project

### **2. Business requirements**

This stage will focus on understanding the problem that businesses face and its solution.



- 2.1 The problem that businesses face
- 2.2 Solution to the problem

### **3. Project Management**

This stage will focus on the creation of a project plan to efficiently and effectively manage the project.

- 3.1 Identification of tasks and resources
- 3.2 Identification of resource and task dependencies
- 3.3 Project plan

### **4. Target market**

This stage will primarily focus on identifying the initial target market and potential customers for the solution.

- 4.1 Preliminary marketplace research
- 4.2 Identification of initial target market
- 4.3 Identification of potential clients

### **5. Market analysis**

This stage will focus on market research and on identifying potential competitors in the marketplace.

- 5.1 Identification of existing solutions
- 5.2 Identification of potential competitors
- 5.3 Future growth opportunities

### **6. Value proposition**

This stage will focus on understanding the advantages that the solution will bring to the clients.

- 6.1 Identification of key features and functionalities
- 6.2 Positive effects of the application on the client's business

### **7. Business model**

This stage will focus on developing a successful business model for the solution.

- 7.1 Identification of feasible business models
- 7.2 Evaluation of the various business models
- 7.3 Comparison of the various business models
- 7.4 Selection of the best business model

### **8. Risk management**

This stage will identify and evaluate the risks associated with the project.

- 8.1 Identification of risks
- 8.2 Risk assessment
- 8.3 Risk mitigation plan

## **9. Budget**

This stage will focus on creating a preliminary budget for the project.

- 9.1 Cost estimation
- 9.2 Revenue forecasts
- 9.3 Preliminary budget

## **10. Project presentation**

This stage will focus on proposing the project to the Comarch management team for funding and technical development.

- 10.1 Abstract
- 10.2 Project presentation
- 10.3 Project report

## **II Technical Project**

This stage will focus on designing the application.

- 1. Requirement analysis
- 2. Integration with existing Comarch CRM and CLM solutions
- 3. Data design
- 4. Architecture design
- 5. Interface design
- 6. Procedural design
- 7. Technical project document

## **III Development**

### **1. Application**

This is the main application that will be used to create the surveys.

#### **1.1 User interface**

The main application will include a fully graphical user interface with a shallow learning curve for new users. The interface will be consistent with Comarch's other CRM and CLM offerings.

##### **1.1.1 User interface framework**

The user interface will be consistent throughout the application and will follow Comarch's human interface guidelines to ensure consistency with Comarch's CRM and CLM solutions.

##### **1.1.2 Cut, copy and paste**

The cut, copy and paste functionality will be provided throughout the application for text, media such as pictures, audio and video and for survey builder modules. This functionality will utilize the standard Windows™ clipboard utility which will provide interoperability with other Windows™ applications. Also, standard keyboard shortcut such as "Ctrl+C", "Ctrl+X" and "Ctrl+V" will be supported.

### **1.1.3 Undo**

The undo functionality will be supported throughout the application to undo any changes made to the survey and elsewhere. The standard “Ctrl+Z” keyboard shortcut will be used to access this functionality.

### **1.1.4 Drag and drop**

The application will support the drag and drop functionality throughout the application for text, media such as pictures, audio and video and for survey builder modules. The application may also support drag and drop capabilities in conjunction with other Windows™ applications.

### **1.1.5 Dashboard**

The application will include an easy-to-use dashboard as its default home screen featuring easy access to built-in templates, recently viewed or modified files, help and status on currently active surveys.

### **1.1.6 Help**

The application will feature a fully-functioning help system that can be used by new and advanced users to discover the full potential of the solution. Following are the features that will be built into the help system.

1.1.6.1 Search engine to search within the help files

1.1.6.2 Rank articles based on their usefulness

1.1.6.3 Automatically check for updated help articles online

1.1.6.4 Quick start guide for new users

## **1.2 File support**

The application will support custom file types for survey templates, user created surveys and survey reports as well as a built-in webpage (HTML) viewer with Java support to preview the surveys. The custom file types should preferably be based on the open XML format so as to support interoperability with other third-party software.

### **1.2.1 File type for survey templates**

This file type will be used to store built-in and user created survey templates. The file type should preferably support the open XML standard to allow interoperability with third-party software.

### **1.2.2 File type for surveys**

This file type will be used to store user created surveys. The file type should preferably support the open XML standard to allow interoperability with third-party software.

### **1.2.3 File type for survey reports**

This file type will be used to store reports created by the analysis tools such as correlation data, tables, graphs etc. The file type should preferably support the open XML standard to allow interoperability with third-party software.

### **1.2.4 Webpage viewer**

The built-in webpage viewer is used to preview surveys within the application before publishing them. The viewer must support HTML with Java web pages.

## **1.3 Customizable modules**

The entire survey is created using easy-to-use, customizable modules that can be dragged and dropped into place. These modules can be interconnected to each other allowing easy creation of complex rules and models. All modules can be easily customized and support a large number of question types and fully support pictures, audio and video.

### **1.3.1 Start Module**

This module is used to create the first screen of any survey. This screen is used to provide general information about the survey. This module also creates a new database to store the survey results. There can only be one start module per survey. Following are the features that this module should ideally support.

- 1.3.1.1 Support for pictures, audio and video
- 1.3.1.2 Display unique traits such as the name of the survey-taker
- 1.3.1.3 Automatic creation of a new database
- 1.3.1.4 Automatic generation of HTML + Java code for the survey

### **1.3.2 Help Module**

This module is used to create a screen that displays detailed instruction on how to fill the survey. This module can be connected to all other kinds of modules to provide survey-takers with instructions and help on all the pages of the survey. This module should ideally support the following features.

- 1.3.2.1 Support for pictures, audio and video
- 1.3.2.2 Display unique traits such as the name of the survey-taker
- 1.3.2.3 Automatic generation of HTML + Java code for the survey

### **1.3.3 Questions Module**

This module is used to ask a question to the survey-takers. Like all other modules, pictures, audio or video can be used within the question. The answer field of this module is directly linked with a field in the database so that the answers from the survey-takers can easily be stored in the database. This module should support the following types of questions.

- 1.3.3.1 Multiple Choice (One answer)
- 1.3.3.2 Multiple Choice (Multiple answers and "Other" text box)
- 1.3.3.3 Matrix of Choices (One answer per row)
- 1.3.3.4 Matrix of Choices (Multiple answers per row and "Other" text box)
- 1.3.3.5 Matrix with Drop-down menus
- 1.3.3.6 Rating Scale

- 1.3.3.7 Single Textbox (open-ended)
- 1.3.3.8 Single Textboxes (open-ended)
- 1.3.3.9 Comment/Essay Box (open-ended)
- 1.3.3.10 Multimedia questions with pictures, audio and/or videos
- 1.3.3.11 Built-in questions for demographics, date and time
- 1.3.3.12 Attachments to be uploaded by the survey-taker

This module should ideally support the following features.

- 1.3.3.13 Support for pictures, audio and video
- 1.3.3.14 Display unique traits such as the name of the survey-taker
- 1.3.3.15 Automatic creation of a new field in the database to store the answer
- 1.3.3.16 Automatic generation of HTML + Java code for the survey that is linked to the appropriate field in the database

#### **1.3.4 End Module**

This module is used to end the survey, provide feedback and/or closing remarks to the survey-taker. There can be multiple end modules in a survey based on the number of possible paths. This module should ideally support the following features.

- 1.3.4.1 Support for pictures, audio and video
- 1.3.4.2 Display unique traits such as the name of the survey-taker
- 1.3.4.3 Feedback on number of points earned by the survey-taker

#### **1.3.5 Random Basket**

The random basket can contain a number of different question modules. The random basket is used to randomize the order in which the questions that are contained in the basket are asked.

#### **1.3.6 Module interconnects**

Module interconnects are used to connect the various modules together to create a fully-working survey. Following are the different kind of interconnects that need to be implemented.

##### **1.3.6.1 Start module-to-question module interconnect**

This is used to connect the start module of the survey to the first question of the survey. The first question of the survey can be varied based on the information that already exists in the database.

##### **1.3.6.2 Question module-to-question module interconnect**

This is used to connect a question in the survey to the next question of the survey. The next question in the survey can be varied based on the answer of the previous question or based on some other information in the database. This leads to the tree based model of the survey.

##### **1.3.6.3 Question module-to-end module interconnect**

This type of interconnect is used to end a survey. The type of ending can be varied with the answer of the question or based on some other information in the database.

#### **1.3.6.4 Start module-to-help module interconnect**

This type of interconnect can be used to display the help page or the instruction page at the start of the survey.

#### **1.3.6.5 Question module-to-help module interconnect**

This type of interconnect can be used to display the help page or the instruction page on any of the question of the survey.

#### **1.3.6.6 End module-to-help module interconnect**

This type of interconnect can be used to display the help page for the results of a survey.

#### **1.3.6.7 Start module-to-random basket interconnect**

This is used to connect the start module of the survey to the random module containing questions that can be asked randomly.

#### **1.3.6.8 Question module-to-random basket interconnect**

This is used to connect a question in the survey to a basket of questions that can be asked at random.

#### **1.3.6.9 Random basket-to-question module interconnect**

This is used to connect a basket of randomly asked question to a particular question after all the questions in the basket have been asked.

#### **1.3.6.10 Random basket-to-random basket interconnect**

This is used to connect two different baskets of questions that can be asked randomly. Questions from the second basket are asked only after all the questions from the first basket have been asked.

#### **1.3.6.11 Random basket-to-end module interconnect**

This is used to connect a basket of randomly asked questions to the end module. The survey will automatically end after all the questions in the random basket have been asked.

### **1.4 Templates and wizard**

The application will include a number of built-in templates and a wizard to help users quickly create professional looking surveys. The application will also be bundled with a template creator for users to create their own templates.

#### **1.4.1 Built-in templates**

Following are the built-in templates that the application will contain.

1.4.1.1 Customer satisfaction survey

1.4.1.2 New product/service survey

- 1.4.1.3 Helpdesk questionnaire
- 1.4.1.4 Point of sale survey
- 1.4.1.5 Medical diagnosis survey
- 1.4.1.6 Emergency services questionnaire
- 1.4.1.7 Job screening questionnaire
- 1.4.1.8 Educational testing questionnaire

#### **1.4.2 Built-in wizard**

The application will also feature an intuitive wizard that can help users quickly create a survey from scratch. The wizard will create a survey by asking the user the following questions.

- 1.4.2.1 Purpose of survey
- 1.4.2.2 Type of database to be used (Oracle™, Microsoft™ SQL or MySQL)
- 1.4.2.3 Publishing method (web-based, printed, interview etc.)
- 1.4.2.4 Type of questions
- 1.4.2.5 Type of answers

#### **1.4.3 Template creator**

Using the template creator, users will be able to save their surveys as templates. These templates can be easily shared with other users by exporting them and are automatically added to the list of built-in templates when imported. Templates can be easily modified to create entirely new surveys while retaining most of the formatting. Following is the functionality that should be supported by the template creator.

##### **1.4.3.1 Save survey as a template**

This saves the current survey as a template and adds it to the list of built-in templates.

##### **1.4.3.2 Export template**

This is used to export a template into a file that can be easily shared with other users.

##### **1.4.3.3 Import template**

This is used to import a template from a file and the imported template is automatically added to the list of built-in templates.

#### **1.5 Search engine**

The application will also feature a system-wide search engine. Following is a list of functionality that needs to be built into the search engine.

##### **1.5.1 Metadata search**

The search engine should be able to quickly search within the metadata tags of files created using the application for hits.

##### **1.5.2 Help system search**

The search engine, by default, should also search within the help system for articles containing the relevant keywords.

### **1.5.3 Template search**

The search engine must also search for relevant user-created and built-in templates.

### **1.5.4 Database search**

The search engine must search the databases of surveys created using the application for relevant information.

### **1.5.5 Survey search**

The search engine must search through all the surveys created using the application for similar questions or answer choices.

## **1.6 Specialized model support**

The application will support two specialized models to create intelligent surveys. Any one or both of these models can be used in a survey.

### **1.6.1 Decision tree model**

The decision tree model is implemented via the use of the various types of interconnects (see 1.3.6). In a decision tree model, the next question or the ending is determined in real-time using the answer of the previous question or based on some other data that already exists in the database.

### **1.6.2 Escalating incentives model**

Escalating incentives models will be implemented by adding a credits field to the database. The user can assign credits to each question in the survey. Whenever the survey-taker answers a question, he is automatically awarded the credits for that question by updating the credits field. The ending of a survey can be varied based on the number of credits so as to fairly reward all the survey-takers.

## **1.7 Automatic data consistency checker**

The automatic data consistency check not only verifies that the survey-taker is a human but also verifies the accuracy of the data entered by the survey-taker. The data is automatically marked invalid if the survey-taker fails the following tests.

### **1.7.1 Human user verification**

To verify that a survey-taker is a human, a random picture is displayed randomly during a survey and the survey-taker is asked to describe the picture using a multiple choice or a text box question. The valid field of the database is automatically updated based on the answer. This can be used to filter out all the inaccurate data from a survey.

### **1.7.2 Data accuracy verification**

To ensure that the survey-taker is accurately responding to the question, the same or a similar question is randomly asked again and the response



is compared with the previous response. If the two responses do not match, then the data is tagged invalid.

### **1.8 Redundant question remover**

Before asking the survey-taker a question, the application automatically checks the corresponding field in the database for an answer. If an answer is already present, then the application automatically skips the question and moves to the next one.

### **1.9 Random function**

The random function is used to randomize the order in which the questions are asked or to randomize the location of the answers in a multiple choice question.

#### **1.9.1 Random function for questions**

The random basket (see 1.3.5) can be used to randomize the order in which the questions are asked.

#### **1.9.2 Random function for answers**

This feature can be used to randomize the location of the answers for multiple choice questions. For example in the first run, the correct answer might be (a) while in the second run, the correct answer might be (c).

### **1.10 Publishing tools**

The application supports various methods in which the survey can be published and made available to the survey-takers.

#### **1.10.1 Validation date support**

This feature can be used to enable the times during which the survey is active. This feature is implemented by locking the database to changes whenever the survey is supposed to be inactive.

#### **1.10.2 Target group support**

This feature can be used to determine the target audience for the survey. This feature is implemented by only allowing survey-takers who match the target audience requirements to add data to the database.

#### **1.10.3 Survey generation**

The application supports the following methods for generating the survey.

##### **1.10.3.1 Web-based survey**

The code for an HTML + Java based survey is generated and published on a specified web server. The survey is also automatically linked to a database that stores all the responses. CSS and AJAX technologies should be automatically used to increase performance and improve the survey-taker's experience.

##### **1.10.3.2 Paper survey**

A paper survey that can be printed and filled out by hand is generated in this method.

#### **1.10.3.3 Script**

A customer service representative (CSR) script is generated using this method. This script will be compatible with a wide range of CRM and CLM solutions including Comarch's solutions.

#### **1.10.3.4 E-mail**

The application will also support exporting the survey to a Microsoft Word™ (.doc) or a PDF (.pdf) document. This survey can be distributed and can be returned when filled via email.

1.10.3.4.1 Export survey to .doc

1.10.3.4.2 Export survey to .pdf

#### **1.10.4 Database generation**

The application will be able to generate at least three kinds of database to store all the responses of the surveys. The fields of these databases are automatically generated based on the survey that is created using the application.

1.10.4.1 Oracle™ database

1.10.4.2 Microsoft™ SQL database

1.10.4.3 mySQL database

#### **1.10.5 Mobile devices support**

This feature can be used to create a mobile version of web-based surveys that can render correctly on mobile devices. The application will include built-in support for a variety of mobile browsers such as Pocket Internet Explorer for Windows Mobile devices and Palm Web Browser for Palm based devices.

#### **1.10.6 Interoperability with third-party tools**

The application will also support a wide range of third-party solutions for publishing the surveys as the application uses industry standard technologies.

### **1.11 Business intelligence tools**

The application will not only include correlation and reporting tools, but will also support a wide range of advanced third-party business intelligence tools.

#### **1.11.1 Correlation tool**

The application will feature a correlation tool to understand and visualize the correlation between multiple fields of the database.

#### **1.11.2 Reporting tool**

The application will also feature a real-time graphing tool to visualize the data collected by the survey.

### **1.11.3 Interoperability with third-party tools**

Since the application already uses popular types of databases to store the data collected by the survey, the application will support a wide-range of third-part business intelligence tools such as Comarch's CRM and CLM solutions that also use Oracle to store and access information.

## **2. Survey interface**

The application will automatically generate HTML + Java code to publish the surveys on the internet. Also CSS and AJAX will be used to enhance the survey-taker's experience.

### **2.1 Standards compliance**

The code generated by the application should be compliant with the latest HTML, Java, CSS and AJAX standards and should render correctly with all the popular web browsers.

- 2.1.1 HTML standards compliance
- 2.1.2 Java standards compliance
- 2.1.3 CSS compliance
- 2.1.4 AJAX compliance

### **2.2 Web server support**

The code generated by the application should be compatible with the following kinds of web servers.

- 2.2.1 Apache HTTP Server
- 2.2.2 Microsoft Internet Information Services (IIS)<sup>TM</sup>
- 2.2.3 Sun Java System Web Server<sup>TM</sup>

### **2.3 One-click publishing support**

This will allow users to publish their surveys in a single click by automatically uploading the survey to a pre-configured web server via FTP.

- 2.3.1 Configuration wizard for the web server setup
- 2.3.2 Manual configuration panel for the web server setup

## **3. Information storage**

All the information collected from a survey is stored in a central, automatically generated database. The database can either be an Oracle<sup>TM</sup>, Microsoft SQL<sup>TM</sup> or a mySQL solution. Following are the key features that the solution will support.

### **3.1 Database search engine**

The application will feature a powerful search engine to search for information within the database. This search engine can also be accessed by the system-wide search tool.

- 3.1.1 Search engine for Oracle<sup>TM</sup> Database
- 3.1.2 Search engine for Microsoft<sup>TM</sup> SQL Database
- 3.1.3 Search engine for mySQL Database

### **3.2 Backup tool**

The application will also feature an automated, incremental backup tool to backup and archive all the information stored within the databases.

3.2.1 Backup tool for Oracle™ Database

3.2.2 Backup tool for Microsoft™ SQL Database

3.2.3 Backup tool for mySQL Database

### **3.3 Data redundancy tool**

The application will feature support for simultaneous maintenance of multiple, identical copies of a database in different locations. This can be used as a built-in disaster recovery option instead of resorting to third-party solutions.

3.3.1 Data redundancy tool for Oracle™ Database

3.3.2 Data redundancy tool for Microsoft™ SQL Database

3.3.3 Data redundancy tool for mySQL Database

## **IV Testing**

This phase involves testing of the developed application.

### **1. Unit testing**

This stage will test each software component, or module to verify that the detailed design for the unit has been correctly implemented.

#### **1.1 User Interface**

1.1.1 User interface framework

1.1.2 Cut, copy and paste

1.1.3 Undo

1.1.4 Drag and drop

1.1.5 Dashboard

1.1.6 Help

#### **1.2 File Support**

1.2.1 File type for survey templates

1.2.2 File type for surveys

1.2.3 File type for survey reports

1.2.4 Webpage viewer

#### **1.3 Customizable modules**

##### **1.3.1 Start Module**

1.3.1.1 Support for pictures, audio and video

1.3.1.2 Display unique traits such as the name of the survey-taker

1.3.1.3 Automatic creation of a new database

1.3.1.4 Automatic generation of HTML + Java code

##### **1.3.2 Help Module**

1.3.2.1 Support for pictures, audio and video

1.3.2.2 Display unique traits such as the name of the survey-taker

1.3.2.3 Automatic generation of HTML + Java code

### 1.3.3 Questions Module

1.3.3.1 Multiple Choice (One answer)

1.3.3.2 Multiple Choice (Multiple answers and “Other” text box)

1.3.3.3 Matrix of Choices (One answer per row)

1.3.3.4 Matrix of Choices (Multiple answers per row and “Other” text box)

1.3.3.5 Matrix with Drop-down menus

1.3.3.6 Rating Scale

1.3.3.7 Single Textbox (open-ended)

1.3.3.8 Single Textboxes (open-ended)

1.3.3.9 Comment/Essay Box (open-ended)

1.3.3.10 Multimedia questions with pictures, audio and/or videos

1.3.3.11 Built-in questions for demographics, date and time

1.3.3.12 Support for attachments to be uploaded by the survey-taker

1.3.3.13 Support for pictures, audio and video

1.3.3.14 Display unique traits such as the name of the survey-taker

1.3.3.15 Automatic creation of a new field in the database to store the answer

1.3.3.16 Automatic generation of HTML + Java code

### 1.3.4 End Module

1.3.4.1 Support for pictures, audio and video

1.3.4.2 Display unique traits such as the name of the survey-taker

1.3.4.3 Feedback on number of points earned by the survey-taker

### 1.3.5 Random Basket

#### 1.3.6 Module interconnects

1.3.6.1 Start module-to-question module interconnect

1.3.6.2 Question module-to-question module interconnect

1.3.6.3 Question module-to-end module interconnect

1.3.6.4 Start module-to-help module interconnect

1.3.6.5 Question module-to-help module interconnect

1.3.6.6 End module-to-help module interconnect

1.3.6.7 Start module-to-random basket interconnect

1.3.6.8 Question module-to-random basket interconnect

1.3.6.9 Random basket-to-question module interconnect

1.3.6.10 Random basket-to-random basket interconnect

1.3.6.11 Random basket-to-end module interconnect

## 1.4 Templates and wizards

### 1.4.1 Templates

1.4.1.1 Customer satisfaction survey template

1.4.1.2 New product/service survey template

1.4.1.3 Helpdesk questionnaire template

1.4.1.4 Point of sale survey template

1.4.1.5 Medical diagnosis survey template

1.4.1.6 Emergency services questionnaire template

1.4.1.7 Job screening questionnaire template

1.4.1.8 Educational testing questionnaire template

### 1.4.2 Wizards

- 1.4.2.1 Purpose of survey wizard
- 1.4.2.2 Type of database wizard
- 1.4.2.3 Publishing method wizard
- 1.4.2.4 Type of questions wizard
- 1.4.2.5 Type of answers wizard
- 1.4.3 Template creator
  - 1.4.3.1 Save survey as a template
  - 1.4.3.2 Export template
  - 1.4.3.3 Import template
- 1.5 Search engine
  - 1.5.1 Metadata search
  - 1.5.2 Help system search
  - 1.5.3 Template search
  - 1.5.4 Database search
  - 1.5.5 Survey search
- 1.6 Specialized model support
  - 1.6.1 Decision tree model
  - 1.6.2 Escalating incentives model
- 1.7 Automatic data consistency checker
  - 1.7.1 Human user verification
  - 1.7.2 Data accuracy verification
- 1.8 Redundant question remover
- 1.9 Random function
  - 1.9.1 Random function for questions
  - 1.9.2 Random function for answers
- 1.10 Publishing tools
  - 1.10.1 Validation date support
  - 1.10.2 Target group support
  - 1.10.3 Survey generation
    - 1.10.3.1 Web-based survey
    - 1.10.3.2 Paper survey
    - 1.10.3.3 Script
    - 1.10.3.4 E-mail
      - 1.10.3.4.1 Export survey to .doc
      - 1.10.3.4.2 Export survey to .pdf
  - 1.10.4 Database generation
    - 1.10.4.1 Oracle Database Support
    - 1.10.4.2 Microsoft SQL database support
    - 1.10.4.3 mySQL database support
  - 1.10.5 Mobile devices support
  - 1.10.6 Interoperability with third-party tools
- 1.11 Business intelligence tools

- 1.11.1 Correlation tool
- 1.11.2 Reporting tool
- 1.11.3 Interoperability with third-party tools
  
- 1.12 Standards compliance
  - 1.12.1 HTML standards compliance
  - 1.12.2 Java standards compliance
  - 1.12.3 CSS compliance
  - 1.12.4 AJAX compliance
  
- 1.13 Web server support
  - 1.13.1 Apache HTTP Server support
  - 1.13.2 Microsoft Internet Information Services (IIS) support
  - 1.13.3 Sun Java System Web Server support
  
- 1.14 One-click publishing support
  - 1.14.1 Configuration wizard for the web server setup
  - 1.14.2 Manual configuration panel for the web server setup
  
- 1.15 Database search engine
  - 1.15.1 Search engine for Oracle Database
  - 1.15.2 Search engine for Microsoft SQL Database
  - 1.15.3 Search engine for mySQL Database
  
- 1.16 Backup tool
  - 1.16.1 Backup tool for Oracle™ Database
  - 1.16.2 Backup tool for Microsoft™ SQL Database
  - 1.16.3 Backup tool for mySQL Database
  
- 1.17 Data redundancy tool
  - 1.17.1 Data redundancy tool for Oracle™ Database
  - 1.17.2 Data redundancy tool for Microsoft™ SQL Database
  - 1.17.3 Data redundancy tool for mySQL Database

## **2. Integration testing**

This stage will expose defects in the interfaces and interaction between integrated components or modules. Progressively, larger groups of tested software components corresponding to elements of the architectural design are integrated and tested until the software works as a whole.

- 2.1 User interface
- 2.2 File support
- 2.3 Customizable modules
- 2.4 Templates and wizards
- 2.5 Search engine
- 2.6 Specialized model support
- 2.7 Automatic data consistency checker
- 2.8 Redundant question remover
- 2.9 Random function

- 2.10 Publishing tools
- 2.11 Business intelligence tools
- 2.12 Standards compliance
- 2.13 Web server support
- 2.14 One-click publishing support
- 2.15 Database search engine
- 2.16 Backup tool
- 2.17 Data redundancy tool

### **3. System testing**

This stage will test the integrated system to verify that it meets the requirements.

- 3.1 Functional testing
- 3.2 Non-Functional testing

### **4. System integration testing**

This will verify that the system can be successfully integrated with external and third party systems as defined in the system requirements.

## **V Deployment**

This stage will involve that actual deployment of the solution in a real life client environment.

### **1. Acceptance testing**

This will be performed by the client to determine acceptance of the product.

### **2. Production Environment Installation**

This stage will be used to produce the fully working software product.

### **3. System installation and configuration**

This stage will involve installing and configuring the system for the client in the real-world environment.

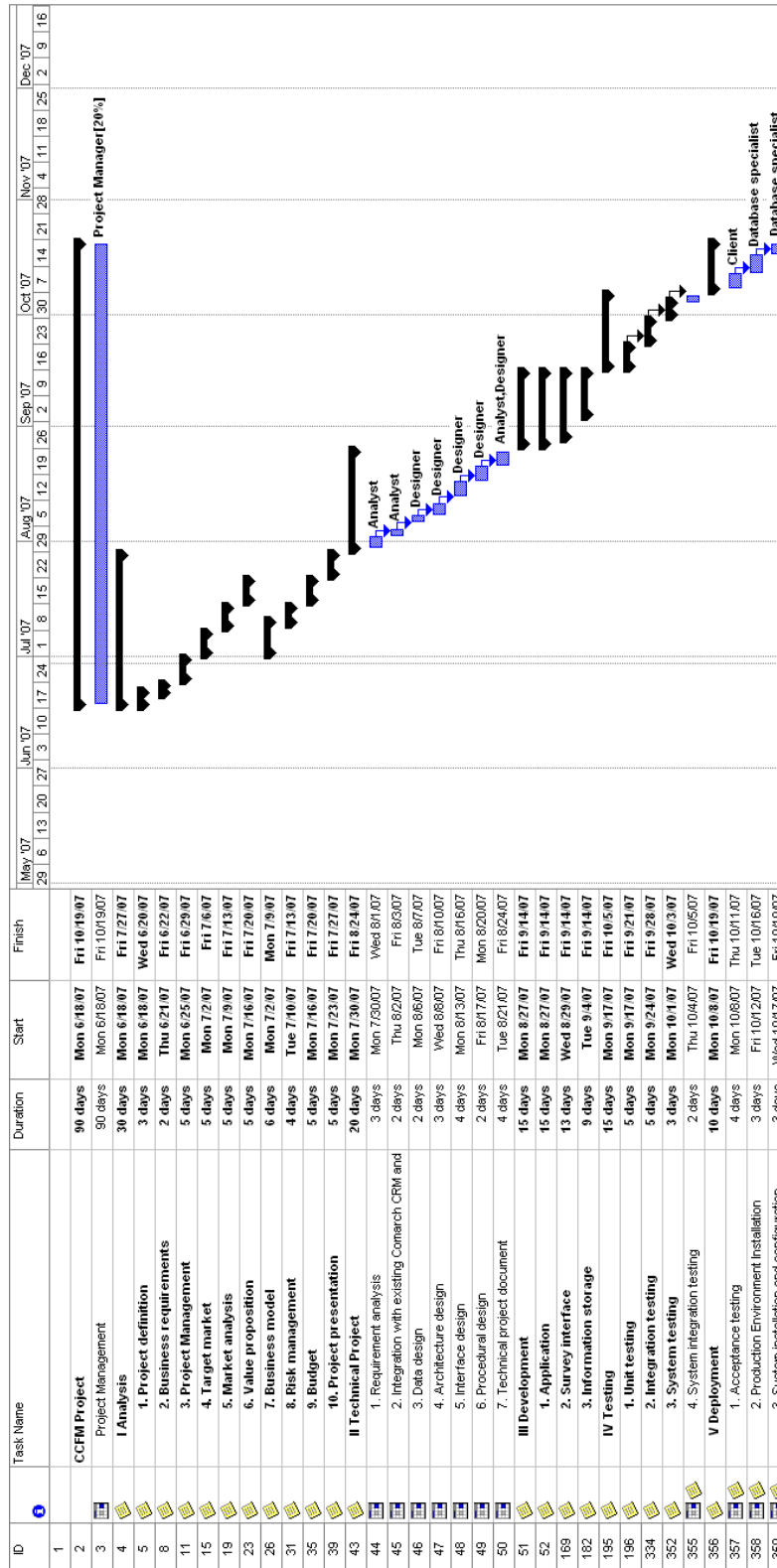


### c. Task dependencies

ID	Task Name	Duration	Start	Finish	Jan '07	Jul '07	Aug '07	Sep '07	Oct '07	Nov '07	Dec '07	Jan '08														
1					3	10	17	24	31	7	14	21	28	4	11	18	25	2	9	16	23	30	6	13	20	
2	<b>CCFM Project</b>	<b>90 days</b>	<b>Mon 6/18/07</b>	<b>Fri 10/19/07</b>																						
3	Project Management	90 days	Mon 6/18/07	Fri 10/19/07																						
4	I Analysis	30 days	Mon 6/18/07	Fri 7/27/07																						
5	<b>1. Project definition</b>	<b>3 days</b>	<b>Mon 6/18/07</b>	<b>Wed 6/20/07</b>																						
6	1.1 Project overview	1.5 days	Mon 6/18/07	Tue 6/19/07																						
7	1.2 Scope of the project	1.5 days	Tue 6/19/07	Wed 6/20/07																						
8	<b>2. Business requirements</b>	<b>2 days</b>	<b>Thu 6/21/07</b>	<b>Fri 6/22/07</b>																						
9	2.1 The problem that businesses face	1 day	Thu 6/21/07	Thu 6/21/07																						
10	2.2 Solution to the problem	1 day	Fri 6/22/07	Fri 6/22/07																						
11	<b>3. Project Management</b>	<b>5 days</b>	<b>Mon 6/25/07</b>	<b>Fri 6/29/07</b>																						
12	3.1 Identification of tasks and resources	1 day	Mon 6/25/07	Mon 6/25/07																						
13	3.2 Identification of resource and task de	1 day	Tue 6/26/07	Tue 6/26/07																						
14	3.3 Project plan	3 days	Wed 6/27/07	Fri 6/29/07																						
15	<b>4. Target market</b>	<b>5 days</b>	<b>Mon 7/2/07</b>	<b>Fri 7/16/07</b>																						
16	4.1 Preliminary marketplace research	2 days	Mon 7/2/07	Tue 7/3/07																						
17	4.2 Identification of initial target market	1.5 days	Wed 7/4/07	Thu 7/5/07																						
18	4.3 Identification of potential clients	1.5 days	Thu 7/5/07	Fri 7/6/07																						
19	<b>5. Market analysis</b>	<b>5 days</b>	<b>Mon 7/9/07</b>	<b>Fri 7/13/07</b>																						
20	5.1 Identification of existing solutions	2 days	Mon 7/9/07	Tue 7/10/07																						
21	5.2 Identification of potential competitors	1.5 days	Wed 7/11/07	Thu 7/12/07																						
22	5.3 Future growth opportunities	1.5 days	Thu 7/12/07	Fri 7/13/07																						
23	<b>6. Value proposition</b>	<b>5 days</b>	<b>Mon 7/16/07</b>	<b>Fri 7/20/07</b>																						
24	6.1 Identification of key features and fun	2.5 days	Mon 7/16/07	Wed 7/18/07																						
25	6.2 Positive effects of the application on	2.5 days	Wed 7/18/07	Fri 7/20/07																						
26	<b>7. Business model</b>	<b>6 days</b>	<b>Mon 7/23/07</b>	<b>Mon 7/9/07</b>																						
27	7.1 Identification of feasible business mc	2 days	Mon 7/23/07	Tue 7/30/07																						
28	7.2 Evaluation of the various business m	1.5 days	Wed 7/4/07	Thu 7/5/07																						
29	7.3 Comparison of the various business	1.5 days	Thu 7/5/07	Fri 7/6/07																						
30	7.4 Selection of the best business model	1 day	Mon 7/9/07	Mon 7/9/07																						
31	<b>8. Risk management</b>	<b>4 days</b>	<b>Tue 7/10/07</b>	<b>Fri 7/13/07</b>																						
32	8.1 Identification of risks	1.5 days	Tue 7/10/07	Wed 7/11/07																						
33	8.2 Risk assessment	1.5 days	Wed 7/11/07	Thu 7/12/07																						
34	8.3 Risk mitigation plan	1 day	Fri 7/13/07	Fri 7/13/07																						
35	<b>9. Budget</b>	<b>5 days</b>	<b>Mon 7/16/07</b>	<b>Fri 7/20/07</b>																						
36	9.1 Cost estimation	2 days	Mon 7/16/07	Tue 7/17/07																						
37	9.2 Revenue forecasts	2 days	Wed 7/18/07	Thu 7/19/07																						
38	9.3 Preliminary budget	1 day	Fri 7/20/07	Fri 7/20/07																						
39	<b>10. Project presentation</b>	<b>5 days</b>	<b>Mon 7/23/07</b>	<b>Fri 7/27/07</b>																						
40	10.1 Abstract	5 days	Mon 7/23/07	Fri 7/27/07																						
41	10.2 Project presentation	5 days	Mon 7/23/07	Fri 7/27/07																						
42	10.3 Project (Analysis) report	5 days	Mon 7/23/07	Fri 7/27/07																						
43	<b>II Technical Project</b>	<b>20 days</b>	<b>Mon 7/30/07</b>	<b>Fri 8/24/07</b>																						

Note: See attached Microsoft Project™ Document for detailed task dependencies.

## d. Timeline



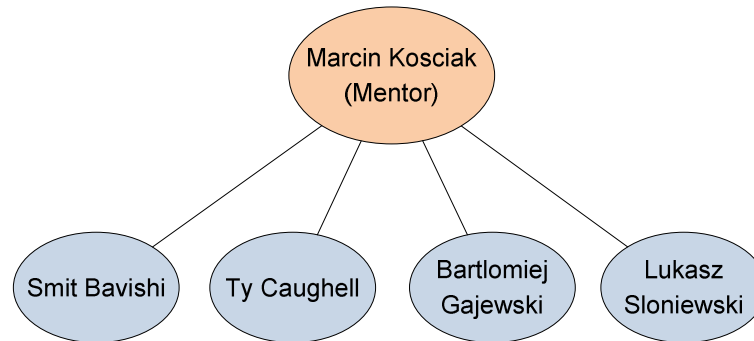
Note: Please refer to the Microsoft Project™ file for a detailed timeline.

## V. Staff organization

This section includes information about the team members, the team structure, accountability, role allocation, reporting and communication.

### a. Team structure

This project will follow a flat team structure where each of the team members will directly communicate and report to the team mentor.



### b. Team member description

Following is a short description of each of the team members.

#### 1. Smit Bavishi

Smit is a fourth year Electrical and Computer Engineering student at the Illinois Institute of Technology. His prior work experience includes working with an IT solutions provider a power engineering company and the Entrepreneurship Program at the Illinois Institute of Technology. Smit spends much of his spare time listening to Metal Shop, the internet's pre-eminent heavy metal radio station.

#### 2. Ty Caughell

Ty is a fourth year Business and Entrepreneurship student at the Illinois Institute of Technology. He plays and manages his own band called The Secondary. Ty spends much of his free time playing the guitar and writing creative, new music.

#### 3. Bartłomiej Gajewski

Bartek is fourth year Electrical and Computer Engineering student at the University of Science and Technology. Bartek spends most of his free time running around the market square with his friends while drinking beer and dancing with beautiful Polish girls till sunrise. Bartek also likes cycling in strange places.

#### 4. Lukasz Sloniewski

Lukasz is a third year Management and Marketing student at the University of Science and Technology and a second year Computer Science student at the Academy of Economics. Lukasz spends most of his free time playing football or playing computer games when the weather is bad. Lukasz likes listening to rap music.

### c. Accounting and role allocation

Following is the job description for each of the team members for the project.

#### 1. Smit Bavishi

Smit is responsible for determining the key functionalities that need to be built into the application.

## **2. Ty Caughell**

Ty is responsible for conducting a project feasibility study and for the business analysis of the project.

## **3. Bartłomiej Gajewski**

Bartek is responsible for creating an executive presentation on the project for the Comarch management team.

## **4. Lukasz Sloniewski**

Lukasz is responsible for developing an intelligent mockup of the solution.

### **d. Reporting and communication**

Following are the reporting and communication schemes that will be used in this project.

#### **1. Formal Reporting**

The mid-term report and the final report will be used for formal reporting on the status of the project.

#### **2. Informal Reporting**

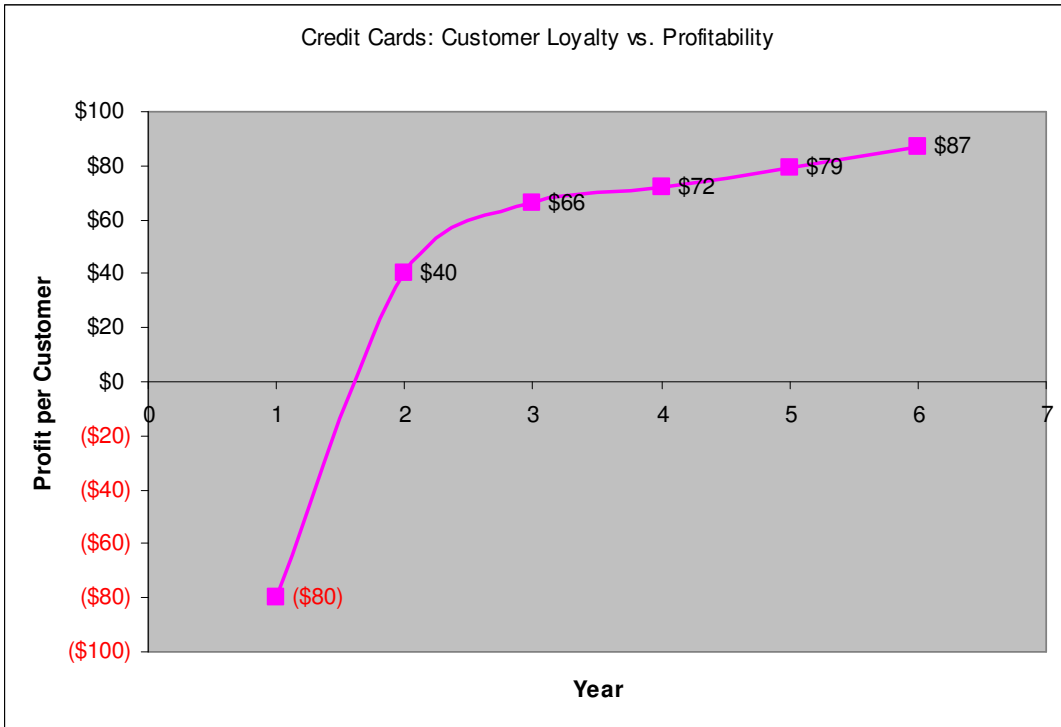
Informal reporting on the status of the project will occur through the use of email and conference calls.

#### **3. Communication**

Communication between the team members can be verbal, via email or via Skype. Minutes for all team meetings will be recorded and archived.

## VI. Appendix

### a. Credit Cards: Customer Loyalty vs. Profitability



### b. Auto Insurance: Customer Loyalty vs. Profitability

