



# **Professional Growth Assessment (PGA) System**

---

## **Final Report**

**June 23<sup>rd</sup>, 2005**  
**Version 1.6**

**Submitted to:**  
**QuantumPM, LLC**

**Prepared by:**  
**Genie Peshkova, Hector Reyes, Sean Siemsen**

---

## Executive Summary

QuantumPM has proposed a development of Professional Growth Assessment (PGA) system for the Field Session 2005. The project entails building a web-based system for users that allows for anonymous exchange of constructive feedback. The purpose of the PGA system is to allow users to submit constructive feedback about other QuantumPM employees in order to use this information for personal and professional growth.

The PGA system supports two user roles: Anonymous Collector of Information (ACI) and regular users. The ACI user's role is to submit requests to other users for evaluation, to filter and cleanse the feedback users submit to ensure it is constructive, and to analyze the evaluation results. The regular users of the system include Evaluatees (employees being evaluated) and Evaluators. Every QuantumPM employee acts as an Evaluatee and Evaluator. Regular users are responsible for submitting evaluations based on the ACI requests and can also view the evaluations that have been submitted for them. Due to the nature of the system, it requires maintaining a level of anonymity and, therefore, all evaluation answers submitted to the database are encrypted to provide the maximum information security.

As requested by QuantumPM, the following technologies were used in the system implementation:

- Microsoft (MS) Sequel (SQL) Server 2003 database – used to store all information submitted into the system, including requests, evaluations, user information, and so on.
- Visual Basic (VB) .NET – used for creating user interface and controls used to display evaluation data on the web.
- Active Server Pages (ASP) .NET – used to display the user interface pages.
- Microsoft InfoPath – application used for gathering and sharing information. InfoPath provides integration with web services, which allows it to efficiently communicate with the MS SQL Server 2003 database and information sharing.
- Web Services – are an accepted standard for integrating web applications and is used for information flow and communication between the ASP .NET, MS SQL Server 2003, and InfoPath.

The PGA project had several risks and constraints. The first constraint is the use of InfoPath and .NET technologies as it required additional time for the learning curve and research. The second constraint was the limited time to complete the project – six weeks including all phases of software development lifecycle. The Field Session group had to mitigate several risks that could have potentially threatened the success of the project. The learning curve required in order to use the technologies requested by QuantumPM has decreased the overall time the team had to spend on the system design and implementation. Throughout the project, the Field Session group has ran into multiple problems with the original system design but has successfully came up with alternatives that allowed them to maintain most of the required system functionality and satisfy the main requirements of the client.

As a result, the Field Session group has created a web-based system to be used from within the QuantumPM network. The Field Session Group was able to accommodate

all system requirements for Phase 1.0 development, but had to compromise the use of InfoPath in some cases. The alternative that allowed the team to still complete the project on time and accommodate majority of the requirements was to alternate the use of VB .NET and InfoPath forms for displaying information to the users. In many cases the use of VB .NET has required less time to implement and provided similar results to those of InfoPath. The team also provided the use of InfoPath in the original display of the request form to the users, which serves as a good starting point for Phase 2 implementation of the PGA system.

---

# Table of Contents

|   |           |
|---|-----------|
| <b>EXECUTIVE SUMMARY .....</b>                        | <b>2</b>  |
| <b>TABLE OF CONTENTS .....</b>                        | <b>4</b>  |
| <b>TABLE OF FIGURES.....</b>                          | <b>6</b>  |
| <b>REVISION HISTORY .....</b>                         | <b>7</b>  |
| <b>ABSTRACT .....</b>                                 | <b>9</b>  |
| <b>1.0 INTRODUCTION.....</b>                          | <b>10</b> |
| 1.1 PURPOSE OF THIS DOCUMENT .....                    | 10        |
| 1.2 SYSTEM OVERVIEW .....                             | 10        |
| 1.2.1 THE VISION .....                                | 10        |
| 1.2.2 MEASURE OF SUCCESS.....                         | 10        |
| 1.2.3 LONG TERM MEASURE OF SUCCESS .....              | 10        |
| 1.2.4 DEFINITION OF DONE.....                         | 10        |
| 1.2.5 PHASE 1 .....                                   | 11        |
| 1.2.6 PHASE 2 .....                                   | 12        |
| 1.3 ASSUMPTIONS .....                                 | 12        |
| 1.4 REFERENCES .....                                  | 13        |
| <b>2.0 SYSTEM DESCRIPTION .....</b>                   | <b>14</b> |
| 2.1 SYSTEM ATTRIBUTES.....                            | 14        |
| 2.1.1 SECURITY .....                                  | 14        |
| 2.1.2 ENCRYPTION.....                                 | 14        |
| 2.1.3 RELIABILITY, AVAILABILITY, MAINTAINABILITY..... | 14        |
| 2.1.4 USABILITY .....                                 | 14        |
| 2.2 PRODUCT FUNCTIONS .....                           | 14        |
| 2.3 SYSTEM PROCESS .....                              | 15        |
| 2.3.1 REQUESTED EVALUATION PROCESS .....              | 15        |
| 2.3.2 AD HOC EVALUATION PROCESS .....                 | 16        |
| 2.4 USER ROLES.....                                   | 17        |
| 2.5 CONSTRAINTS .....                                 | 17        |
| <b>3.0 PHASE 1.0 SYSTEM REQUIREMENTS .....</b>        | <b>18</b> |
| 3.1 FUNCTIONAL REQUIREMENTS .....                     | 18        |
| 3.2 NON-FUNCTIONAL REQUIREMENTS.....                  | 22        |
| <b>4.0 PHASE 2.0 REQUIREMENTS .....</b>               | <b>25</b> |
| 4.1 FUNCTIONAL REQUIREMENTS .....                     | 25        |
| <b>5.0 PHASE 1.0 PGA SYSTEM DESIGN .....</b>          | <b>27</b> |
| 5.1 PGA SYSTEM PROJECT SCHEDULE .....                 | 27        |
| 5.1.1 PGA WORK BREAKDOWN STRUCTURE.....               | 28        |
| 5.1.2 PGA GANTT CHART VIEW .....                      | 30        |
| 5.1.3 PGA CRITICAL TASKS LIST .....                   | 32        |
| 5.1.4 PGA SYSTEM MILESTONES .....                     | 33        |
| 5.2 DATABASE DESIGN .....                             | 34        |
| 5.2.1 DATABASE DETAILS .....                          | 34        |
| 5.2.2 DATABASE STORED PROCEDURES.....                 | 36        |
| 5.3 SYSTEM INTERACTION .....                          | 38        |
| 5.4 ENCRYPTION DESIGN .....                           | 39        |
| 5.5 USER INTERFACE DESIGN .....                       | 40        |
| 5.5.1 ACI UI SCREENS.....                             | 40        |
| 5.5.2 REGULAR USER UI SCREENS .....                   | 43        |
| 5.5.3 PGA EVALUATION FORM.....                        | 45        |
| 5.5.4 PGA EVALUATION REQUEST .....                    | 46        |
| <b>6.0 PHASE 1.0 IMPLEMENTATION DETAILS.....</b>      | <b>48</b> |
| 6.1 PGA SYSTEM IMPLEMENTATION OVERVIEW .....          | 48        |

|   |           |
|---|-----------|
| 6.2 ASP .NET IMPLEMENTATION .....                                 | 48        |
| 6.3 VB .NET IMPLEMENTATION .....                                  | 49        |
| 6.4 WEBSITE IMPLEMENTATION.....                                   | 49        |
| 6.4.1 THE LOGIN PAGE .....  | 49        |
| 6.4.2 THE HOME PAGE .....   | 49        |
| 6.4.3 ACI FUNCTIONS.....  | 49        |
| 6.4.4 THE USER FUNCTIONS.....                                     | 50        |
| 6.5 INFOPATH IMPLEMENTATION .....                                 | 51        |
| 6.5.1 INFOPATH OPERATION .....                                    | 51        |
| 6.6 WEB SERVICES IMPLEMENTATION .....                             | 53        |
| 6.8 PGA ENCRYPTION IMPLEMENTATION.....                            | 54        |
| 6.9 FUTURE RECOMMENDATIONS AND LESSONS LEARNED.....               | 54        |
| 6.9.1 LESSONS LEARNED .....                                       | 54        |
| 6.9.2 FUTURE RECOMMENDATIONS .....                                | 55        |
| <b>QUESTIONS AND CONCERNS .....</b>                               | <b>56</b> |
| <b>APPENDIX A – EVALUATION QUESTIONS .....</b>                    | <b>57</b> |
| <b>APPENDIX B - DEFINITIONS, ABBREVIATIONS, AND ACRONYMS.....</b> | <b>66</b> |

---

## Table of Figures

|  |    |
|--|----|
| Figure 1 – PGA Iteration Based Evaluation Process.....   | 15 |
| Figure 2 – PGA Ad Hoc Evaluation Process .....           | 16 |
| Figure 3 Database ER Diagram.....                        | 34 |
| Figure 4 Full System Interaction Diagram .....           | 39 |
| Figure 5 Encryption Design Diagram .....                 | 39 |
| Figure 6 PGA ACI Main Screen .....                       | 41 |
| Figure 7 PGA ACI Request Screen.....                     | 41 |
| Figure 8 PGA ACI Submitted Evaluations Screen.....       | 42 |
| Figure 9 PGA ACI Modified Evaluations Screen.....        | 43 |
| Figure 10 PGA Regular User Main Screen.....              | 44 |
| Figure 11 PGA Regular User Evaluations To Do Screen..... | 45 |
| Figure 12 PGA Regular User My Evaluations Screen.....    | 45 |
| Figure 13 Evaluation Form Example (InfoPath) .....       | 46 |
| Figure 14 PGA New Evaluation Request InfoPath Form.....  | 47 |
| Figure 15 InfoPath Web Service Communication .....       | 51 |
| Figure 16 Evaluation Request InfoPath Data Fields.....   | 52 |

---

## Revision History

| <b>Revision Number</b> | <b>Description of Change</b>                    | <b>Version Date</b> | <b>Changed by</b> | <b>Affected Sections</b>  |
|------------------------|---|---------------------|-------------------|---|
| 0.1                    | Document Created, content added to all sections | 05/19/2005          | Genie Peshkova    | All   |
| 0.2                    | Content modified                                | 05/20/2005          | Quantum PM        | All   |
| 0.3                    | Content added                                   | 05/22/05            | Sean Siemsen      | Introduction, Phase 1.0 Requirements  |
| 0.4                    | Content added                                   | 05/22/05            | Hector Reyes      | Phase 1.0 Requirements  |
| 0.5                    | Content added                                   | 05/22/05            | Genie Peshkova    | All, added Appendix A   |
| 0.6                    | Content modified                                | 05/27/05            | Genie Peshkova    | Phase 1.0 Requirements  |
| 0.7                    | Content added                                   | 05/30/05            | Genie Peshkova    | Design section created, project schedule added  |
| 0.8                    | Content added                                   | 05/30/05            | Sean Siemsen      | Database design diagrams  |
| 0.9                    | Content added                                   | 05/30/05            | Hector Reyes      | System interaction diagrams   |
| 1.0                    | Content added                                   | 05/30/05            | Genie Peshkova    | User Interface Mockup Screens added   |
| 1.1                    | Content updated                                 | 06/03/05            | Genie Peshkova    | Separated Requirements section into Phase 1 and Phase 2, also modified the requirements per client review |
| 1.2                    | Content added                                   | 06/14/05            | Genie Peshkova    | Phase 1.0 requirements changes updated, Phase 1.0 Design updated to reflect new design.                   |
| 1.3                    | Content added                                   | 06/14/05            | Genie Peshkova    | Phase 1.0 Implementation Details section added  |

| <b>Revision Number</b> | <b>Description of Change</b>  | <b>Version Date</b> | <b>Changed by</b>             | <b>Affected Sections</b>   |
|------------------------|---|---------------------|-------------------------------|--|
| 1.4                    | Content added   | 06/17/05            | Hector Reyes and Sean Siemsen | Phase 1.0 database and encryption design updated, Phase 1.0 implementation of ASP.NET and VB .NET added. |
| 1.5                    | Reorganized final report to include all required sections; Executive Summary and Abstract added   | 06/19/05            | Genie Peshkova                | All  |
| 1.6                    | Incorporated feedback received from the report review, updated some minor details, and proof-read | 06/20/05            | Genie Peshkova                | All  |



---

## Abstract

The Professional Growth Assessment (PGA) system is a web-based application that allows anonymous exchange of feedback between QuantumPM employees with the purpose of using it for personal and professional growth. The PGA system supports two types of users: Anonymous Collector of Information (ACI) and regular system users that include Evaluatees (employees being evaluated) and Evaluators. The ACI is responsible for submitting evaluation requests to users to evaluate other employees in the company. The ACI is also responsible for cleansing the submitted evaluations to ensure that only constructive feedback is sent to the Evaluatees. The regular system users are able to submit evaluations about specified QuantumPM employees, based on the ACI requests. The regular system users are also able to view the ACI-modified evaluations submitted about them in order to use that information for personal and professional improvement. The PGA system is built using Microsoft SQL Server 2003, ASP.NET, VB .NET, InfoPath, and Web Services. The use of these technologies was chosen to accommodate system flexibility and ability to further expand and enhance the system with new options.

---

## 1.0 Introduction

### 1.1 Purpose of This Document

The purpose of this document is to detail the requirements, assumptions, constraints, system design, and implementation of the Professional Growth Assessment (PGA) System. The system specifications include functional and non-functional requirements for Phase I and II implementation. The system design includes the design of main system components, updated to reflect the final decisions and changes to the original design. The system implementation includes details about the system implementation that were not included in the design section and also provides a more technical description of the system.

### 1.2 System Overview

The System Overview section is meant to provide a high level system overview, including the vision behind the system, the measure of success defined by the client, definition of done, and high level original requirements.

#### 1.2.1 The Vision

QuantumPM wants to provide its employees the ability for individual self-improvement. QPM would like individuals to receive anonymous feedback about how they work that is specific, actionable, and provides examples. The feedback received should not be tied to job retention or salary decisions in any way, thus it will be a non-threatening way of providing input by one's peers. Frequency of feedback could be on an ad hoc or requested basis.

#### 1.2.2 Measure of Success

The measure of success for the PGA system project is defined as follows:

Build a system that facilitates all QPM personnel to evaluate their peers in a constructive manner.

#### 1.2.3 Long Term Measure of Success

The long term measure of success for the PGA system is defined as follows:

The PGA system is used by everyone in the organization. Everyone in QPM has participated and has found the system useful.

#### 1.2.4 Definition of Done

QuantumPM defines the software development of PGA as done when the following high level items are completed:

- All QPM personnel must be able to fill out and submit peer evaluations
- Constructive feedback is provided to the Evaluatee
- A distribution list facilitates evaluation requests

- A central database is used to store the encrypted evaluations
- Outstanding evaluation requests and evaluations submitted are reported to the Anonymous Collector of Information (ACI)

### 1.2.5 Phase 1

Phase 1 release will be completed by 06/20/2005 and will include an executable architecture of the entire system. The functionality provided by this first iteration must meet the initial (minimal client requirements), and encompass all of the components of the final system, as described by the Definition of Done.

---

**Note**

Priorities of some of the requirements described in this section have been lowered by the client in order to accommodate unexpected changes to system design and the learning curve of the Field Session Group. See Phase 1.0 Requirements section of this document for the final Phase 1 requirements.

---

Phase 1 implementation includes:

- System provides a pre-built form for the Evaluators to use.
- System is accessed on the network and obtains users credentials from the Windows Authentication.
- System includes a list of all QPM employees and allows Evaluator to select the user to evaluate:
  - The structured evaluation should be sent to X Evaluators.
  - Ad hoc evaluation can be performed by anyone for anyone (might be aggregated into the structured one or sent as needed).
- Upon the selection of a peer to evaluate, a screen with a list of questions appears, allowing the Evaluator to fill out the form and submit it for the records.
- System provides a list of reports, based on user permissions:
  - User with the ACI (see User Role section for definition) permission will have ability to view all evaluations.
  - Users with regular user permissions can submit evaluations based on ACI requests and to view the evaluations that were submitted for them.
  - User with the ACI permission will have ability to submit requests for evaluation to system users with the people they need to evaluate.
- Reports/evaluations include date of submission and answers to questions provided.
- Structured/formal evaluation should also include an iteration identifier.
- The reports/evaluations must also keep a historical record of any modifications made to the evaluations by the ACI.
- System must use InfoPath software for user interface.
- System must have a central data store in a form of a MS SQL Server database.

- System must encrypt the evaluation form answers filled out by Evaluators to ensure privacy of the answers and the Evaluatee.

### 1.2.6 Phase 2

Phase 2 contains the requirements that are “nice to have” and that can be implemented if Phase 1 implementation is ahead of schedule. Otherwise, Phase 2 requirements are considered out of scope for the original project proposal and are not required for success of the project.

Phase 2 requirements include:

- System provides a simple maintenance screen that allows users to setup a profile that contains a description of personal goals, and so on.
- A mechanism to automate the matching of who should evaluate who is implemented. The mechanism is based on the volume of work performed together by two employees. For example, if two people have worked on a lot of projects together in X period of time, then they will be selected to evaluate each other.
- System provides an administration screen that allows setting user permissions, updating users, and so on.
- ACI submission of the “cleansed” evaluation to the Evaluatee via an automated system.
- ACI receives alerts about evaluations that include ratings out of range. The out of range ratings are defined on specifying 1, 2, or 5 (complete scale of 6).

### 1.3 Assumptions

The assumptions for Phase 1 are as follows:

- All system users have network access to the system and have InfoPath installed on their Remote Desktop or local computer.
- System must be operable, but it does not have to include all of the functionality requested by QuantumPM. Changes to the scope will be mutually agreed upon in advance between the CSM team and QuantumPM and documented via the QuantumPM change order process.
- System functionality may be limited to support the short time frame of the Phase 1 implementation. Changes to the scope will be mutually agreed upon in advance between the CSM team and QuantumPM and documented via the QuantumPM change order process.
- System success is based on the Measure of Success and the Definition of Done provided by QuantumPM.
- Any additional changes or modification requests to the system requirements by QuantumPM have to be handled using a formal change request process and QuantumPM assumes responsibility for delays in schedule and scope covered.

## 1.4 References

QuantumPM Proposal – QuantumPM has submitted a proposal to the Field Session team on 05/16/05. The QuantumPM Proposal contains the description of the system and high level set of requirements that the client has provided and can be seen in the System Overview section of this document.

QuantumPM Professional Growth Assessment Final Details – QuantumPM has provided this document with detailed questions that must appear in the evaluation.

---

## 2.0 System Description

The System Description section provides a more detailed description of the PGA system, the perspective of the system, its users, product functions, and constraints.

### 2.1 System Attributes

This section describes the various system attributes that may not be covered in the system requirements, but are important to note.

#### 2.1.1 Security

Username and password protection will be enabled on the system. This authentication scheme will ensure the filtering of evaluation requests submitted by the ACI, as well as limiting the evaluations seen by Evaluatees to ones which were submitted only for them.

#### 2.1.2 Encryption

All completed evaluation questions are encrypted to allow for maximum privacy and to maintain the concept of “anonymous constructive feedback” for the Evaluatees.

#### 2.1.3 Reliability, Availability, Maintainability

The system will be accessible 24 hours, 7 days a week with the exception of scheduled maintenance and unexpected outage as outlined in the Service Level Agreement. Data backups will be taken on a nightly basis with the backup of the server.

#### 2.1.4 Usability

Usability describes items that will ensure the user-friendliness of the software. Examples include error messages that direct the user to a solution, input range checking as soon as entries are made, and order of choices and screens corresponding to user preferences.

## 2.2 Product Functions

PGA System includes 8 main functions:

- ACI assignment of evaluation requests to Evaluators
- Completed evaluation submission to a database
- ACI review of submitted evaluations
- ACI modifying submitted evaluations to remove non-constructive feedback
- ACI modified evaluations do not replace the original evaluation
- Reports of evaluations submitted, requests submitted, and modified evaluations for the ACI user
- Encryption of the evaluation questions for privacy

- Central data store for all evaluations that allows for nightly backups and historical record keeping

## 2.3 System Process

The PGA System supports two process flows for Evaluation submission:

- Requested Evaluation Process
- Ad Hoc Evaluation Process

### Note

The Ad Hoc process documented in this section demonstrates the proposed information flow, but was not actually implemented in Phase 1 due to time constraints.

### 2.3.1 Requested Evaluation Process

The Requested evaluation process allows the ACI user to submit an evaluation request to the Evaluator with a list of employees to evaluate.

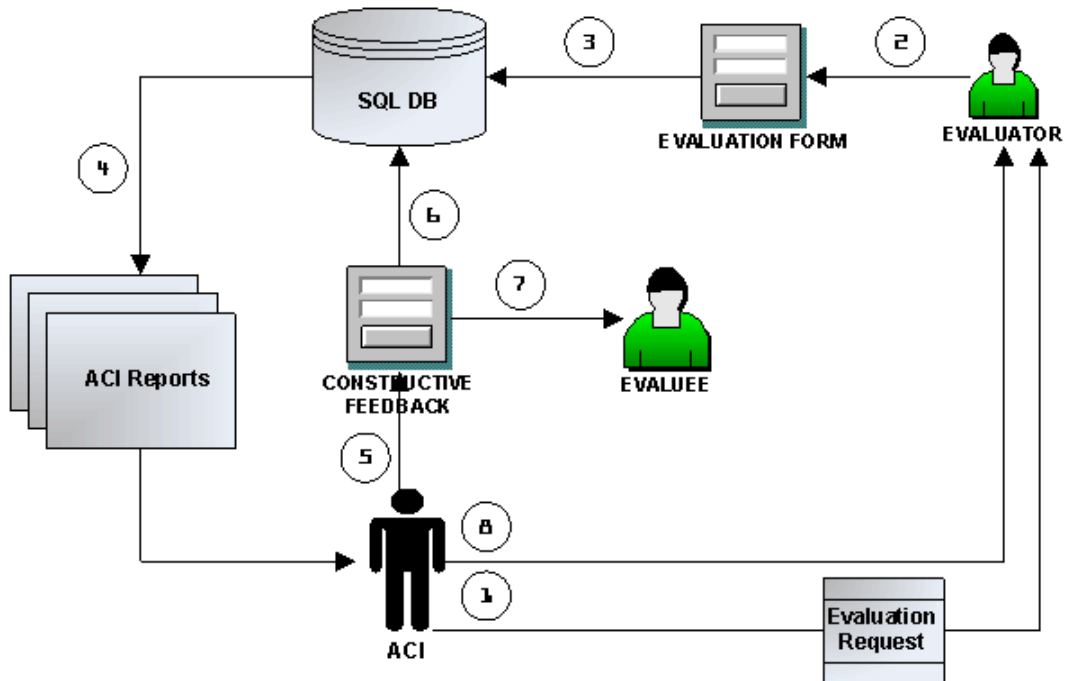


Figure 1 – PGA Iteration Based Evaluation Process

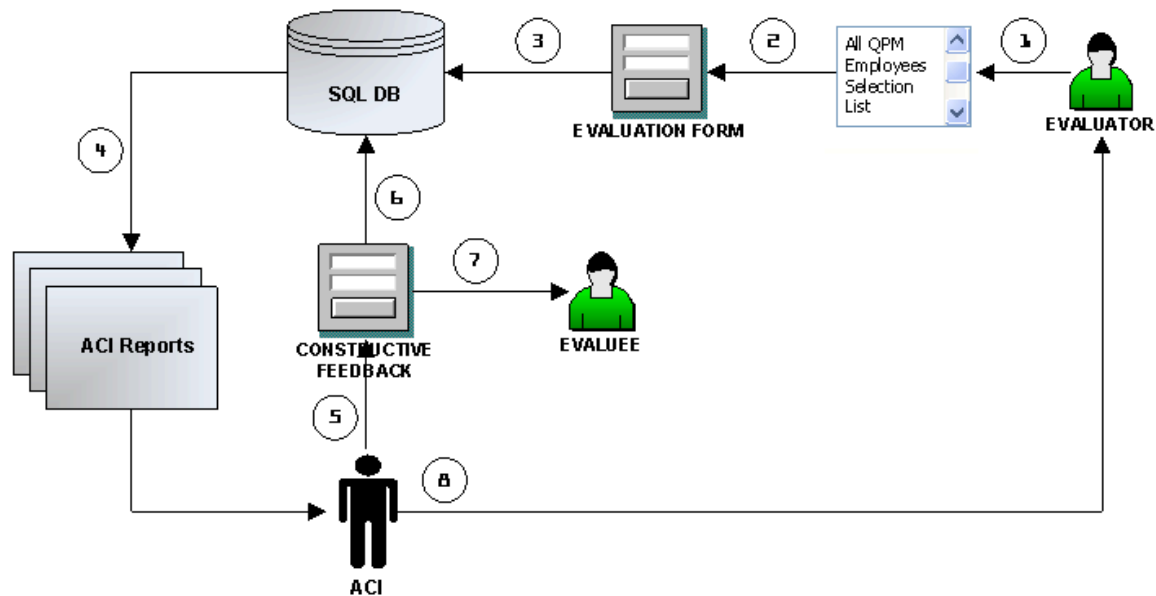
The Requested evaluation process consists of the following steps:

1. ACI sends an evaluation request to the Evaluator.
2. Evaluator fills out a pre-defined evaluation form for a chosen Evaluatee.

3. Evaluation form is encrypted and submitted to the database.
4. ACI retrieves the evaluation from the database.
5. ACI removes non-constructive feedback from the form.
6. The updated evaluation form is saved to the database as a separate copy.
7. ACI sends the constructive feedback to the Evaluatee via submitting a modified evaluation to the Evaluatee. The evaluation must appear on the Evaluatee report.
8. ACI sends comments to the Evaluator about the submitted evaluation (email process not supported by PGA).

### 2.3.2 Ad Hoc Evaluation Process

The Ad hoc evaluation process allows any employee to submit an evaluation about any other employee at any point in time.



**Figure 2 – PGA Ad Hoc Evaluation Process**

The Ad hoc evaluation process consists of the following steps:

1. Evaluator selects an Ad Hoc Evaluation form.
2. Evaluator selects the Evaluatee from a list of all QPM employees.
3. Evaluation form is encrypted and submitted to the database.
4. Anonymous Collector of Information (ACI) retrieves the evaluation from the database.
5. ACI removes non-constructive feedback from the form.
6. The updated evaluation form gets saved to the database as a separate copy.
7. ACI sends the constructive feedback to the Evaluatee.
8. ACI sends comments to the Evaluator about the submitted evaluation (email process not supported by PGA).



## 2.4 User Roles

PGA System supports two types of user roles:

- Regular System Users:
  - Evaluator – all QPM employees. This role is responsible for submitting an evaluation on someone else in the company.
  - Evaluatee – QPM employee being evaluated.
- Anonymous Collector of Information (ACI) – all feedback goes to this person and it is identified as to who it came from; information gets cleaned – non-constructive feedback gets excoriated (possibly sent back to the Evaluator) and useful feedback gets passed on to the individual being reviewed. The ACI is also responsible for analyzing the evaluation results and providing additional comments to the Evaluatee.

## 2.5 Constraints

The following constraints apply to the development and design of the PGA system:

- The Field Session Team has only 6 weeks to complete design, analysis, development, testing, and release of the PGA system.
- InfoPath must be used to create, fill, and view forms. Alternative methods may need to be implemented as an alternative in conjunction with InfoPath.
- An MS SQL server must be used to store the data contained in the forms.
- InfoPath has limitations of being able to display web-based forms.
- For optimum success of the system, it is imperative that the users use the system.

---

## 3.0 Phase 1.0 System Requirements

### 3.1 FUNCTIONAL REQUIREMENTS

The requirements conform to RFC 2119, "Key words for use in RFCs to Indicate Requirement Levels." Functional requirements are broken down by the main system components.

| ID | Description |
|----|-------------|
|----|-------------|

#### **3.1.1**      **Phase 1 Functional Requirements**

|                |   |
|----------------|---|
| <b>3.1.1.1</b> | <b><i>Evaluation Mode</i></b>   |
| 3.1.1.1.1      | The PGA System supports one type of the evaluation mode: <ul style="list-style-type: none"><li>▪ Requested Evaluation</li></ul> <hr/> <b>Note</b> <p style="text-align: center;">Additional types of the evaluations can be added in the subsequent phases.</p> |
| 3.1.1.1.2      | Requested Evaluation follows the Scheduled Evaluation Process as shown in Figure 1.   |
| <b>3.1.1.2</b> | <b><i>Evaluation Request</i></b>  |
| 3.1.1.2.1      | Evaluation Request is used for the Requested Evaluations only.  |
| 3.1.1.2.2      | Evaluation Request allows ACI to select the Evaluatee from a dynamic pre-populated list of all QPM employees and to select corresponding Evaluators from a dynamic, pre-populated list of all QPM employees.  |
| 3.1.1.2.3      | Upon the submission of the Evaluation Request, each Evaluator will see an evaluation request appear on their PGA screen.  |
| 3.1.1.2.4      | ACI must also select a Due Date, provided in the form of a date picker, by when the evaluation must be completed.   |
| 3.1.1.2.5      | ACI must also select an Iteration Number to indicate the evaluation period – an iteration that the request is part of.  |
| 3.1.1.2.6      | An additional comments field is included on the Evaluation request that accepts text to allow the ACI to enter comments for the request.  |
| <b>3.1.1.3</b> | <b><i>Evaluation Form</i></b>   |
| 3.1.1.3.1      | The Evaluation Form must include today's date, the Evaluatee, Evaluation Iteration Number, and the list of questions.   |

| ID   | Description  |
|--|--|
| <b>3.1.1</b>   | <b><u>Phase 1 Functional Requirements</u></b>  |
| <p>3.1.1.3.2</p> <p>3.1.1.3.3</p> <p>3.1.1.3.4</p> <p>3.1.1.3.5</p> <p>3.1.1.3.6</p> <p>3.1.1.3.7</p> <p>3.1.1.3.8</p> | <p>The Requested Evaluation Form provides a pre-populated Evaluatee name, selected by the ACI using the Evaluation Request.</p> <p>The Requested Evaluation Form provides an extensive list of questions for the Evaluator that is provided by QuantumPM.</p> <p>The Requested Evaluation Form provides two functions:</p> <ul style="list-style-type: none"> <li>▪ Save Changes – allows the user to save the evaluation to the database, but not submit it to the ACI. This functionality allows the user to come back and continue/modify the form at a later time.</li> <li>▪ Submit – submit the form to the ACI and does not allow any further modifications.</li> </ul> <p>The Requested Evaluation Form provides the following fields:</p> <ul style="list-style-type: none"> <li>▪ Date* – Today’s date, automatically filled in</li> <li>▪ Iteration Number* – automatically filled in based on the value entered by the ACI in the Evaluation Request</li> <li>▪ Evaluatee* – automatically filled in based on the ACI selection in the Evaluation Request</li> <li>▪ For each question: <ul style="list-style-type: none"> <li>○ Rating from 1 – 6* (Poor, Fair, Average, Good, Outstanding, Unable to Rate) – Required</li> <li>○ Expandable Comments field – allows the Evaluator to supply additional comments under each question rating</li> </ul> </li> </ul> <hr/> <p><b>Note</b></p> <p style="text-align: center;"><u>Fields marked with an asterisk (*) are required fields.</u></p> <p>Error checking for required fields in the Requested Evaluation Form is performed.</p> <p>Evaluation form must include areas of the evaluation and questions, as provided in Appendix A. Each area will appear on a separate page, contained within the same evaluation form to allow the separation of the evaluation areas.</p> <p>Evaluation form also contains ACI Notes field that is invisible to all other users and will be used by the ACI to record notes to himself.</p> |
| <b>3.1.1.4</b>   | <b><i>Edit Mechanism</i></b>   |
| 3.1.1.4.1  | An Edit Mechanism is available to the ACI user to allow for removal of the non-constructive feedback.  |

| ID             | Description   |
|----------------|---|
| <b>3.1.1</b>   | <b><u>Phase 1 Functional Requirements</u></b>   |
| 3.1.1.4.2      | Edit Mechanism is accessed from the Requested Evaluations Report (see Requirement 3.1.1.6.3).   |
| 3.1.1.4.3      | The Evaluation Edit screen allows ACI to modify any information displayed in the form and submit it to the database upon clicking the Submit button.  |
| 3.1.1.4.4      | <p>The Edit screen allows for two functions:</p> <ul style="list-style-type: none"> <li>▪ Save Changes – allows ACI to save evaluation to the database, but not submit it to the Evaluee. This functionality allows the ACI to come back and continue/modify the form at a later time.</li> <li>▪ Submit – submits the form to the Evaluee and does not allow any further modifications.</li> </ul> |
| <b>3.1.1.5</b> | <b><i>Database</i></b>  |
| 3.1.1.5.1      | A central MS SQL Server database is used to store the Evaluation Forms.   |
| 3.1.1.5.2      | <p>The database contains 2 copies of the evaluation forms:</p> <ul style="list-style-type: none"> <li>▪ Original Evaluation form submitted by the Evaluator</li> <li>▪ Edited Evaluation form submitted by the ACI</li> </ul>   |
| 3.1.1.5.3      | Answers (rating and text comments) of the evaluation form are encrypted to ensure privacy.  |
| 3.1.1.5.4      | A list of all QPM employees must display in the Evaluation Form and the Evaluation Request.   |
| 3.1.1.5.5      | The ACI will have access to all information in the database via the report screens provided in the PGA.   |
| 3.1.1.5.6      | Regular system users (Evaluators and Evaluees) will have access to evaluations they have been requested to submit and evaluations submitted for them (after they have been cleansed by the ACI).  |
| 3.1.1.5.7      | <p>The data contained in the database will consist of:</p> <ul style="list-style-type: none"> <li>▪ Evaluation questions</li> <li>▪ Evaluation answers</li> <li>▪ Dates of evaluations (Due Date and Submission Date)</li> <li>▪ Evaluator names</li> <li>▪ Evaluee names</li> </ul>  |

| ID                  | Description  |
|---------------------|--|
| <b><u>3.1.1</u></b> | <b><u>Phase 1 Functional Requirements</u></b>  |
| 3.1.1.5.7           | <ul style="list-style-type: none"> <li>▪ Evaluation iteration identifiers</li> <li>▪ ACI Notes field (seen only by the ACI)</li> </ul> <p>The Database must be designed in a way to support flexibility of changing questions, adding additional evaluation forms, and expanding the functionality to support more complex evaluation scenarios.</p>   |
| <b>3.1.1.6</b>      | <b><i>ACI Reports</i></b>  |
| 3.1.1.6.1           | <p>The PGA System must provide the following three types of reports to the ACI user:</p> <ul style="list-style-type: none"> <li>▪ Submitted Evaluations</li> <li>▪ Modified Evaluations</li> <li>▪ Request History</li> </ul>  |
| 3.1.1.6.2           | <p>The Submitted Evaluations report must display a list of all submitted evaluations that have not been yet modified by the ACI, filtered by date (most recent at the top). This list must also show the date of submission. This screen also provides functionality for the ACI to view each evaluation (without modifying it) and to modify it. When ACI selects to modify the evaluation, he can either Save it, which just saves the changes to the database or Submit it, which sends it to the Evaluee. Saved, but not sent evaluations remain in the Submitted Evaluations list.</p>  |
| 3.1.1.6.3           | <p>The Modified Evaluations report must display a list of all modified (and sent) evaluations, filtered by date (most recent at the top). This report is meant to serve as a historical record of all evaluations sent to the Evaluee.</p>   |
| 3.1.1.6.4           | <p>Request History provides a list of all submitted requests. This report must list the date of the request, the Evaluator submitted to, the Evaluee name, the evaluation iteration number, and the due date. This list will also identify whether the request is outstanding or not. This is determined by whether there has been an evaluation submitted corresponding to the request. The Request History report will also provide functionality for the ACI to view each submitted request via InfoPath. The top of the Request History screen will contain Submit New Request link that allows the ACI to submit a new evaluation request. Once the request has been submitted, it appears in the Request History report.</p> |
| 3.1.1.6.5           | <p>The ACI main screen will also include the Manage Iterations screen that allows the ACI to create new evaluation iteration by specifying iteration start and end dates. The iteration numbers themselves are system generated and are incremented by one every iteration.</p>  |

| ID | Description |
|----|-------------|
|----|-------------|

**3.1.1 Phase 1 Functional Requirements**

|                |  |
|----------------|--|
| <b>3.1.1.7</b> | <b><i>Regular User Reports</i></b>   |
| 3.1.1.7.1      | <p>Regular PGA System users must have access to one type of report:</p> <ul style="list-style-type: none"> <li>▪ "Evaluations to Do"</li> <li>▪ "My Evaluations"</li> </ul>  |
| 3.1.1.7.2      | <p>The My Evaluations report allows users to view evaluations submitted for them. This screen lists the submission date and the evaluation mode fields.</p>  |
| 3.1.1.7.3      | <p>The My Evaluations report hides the Evaluator's name to allow for anonymous comment process. It also hides the ACI notes field.</p>   |
| 3.1.1.7.4      | <p>Evaluations to Do screen lists all outstanding requests the Evaluator has not submitted evaluations for. When the users select Save Changes button on the Evaluation, the request remains listed on the Request screen. When the user submits the evaluation, the request is removed from the Request list.</p> |
| <b>3.1.1.8</b> | <b><i>Requested Evaluation Iterations</i></b>  |
| 3.1.1.8.1      | <p>The PGA system also supports a concept of Iterations. Iterations are meant to distinguish the evaluation period and allow for better tracking of Evaluations and Evaluation Requests.</p>   |
| 3.1.1.8.2      | <p>The ACI has ability to add new iteration by specifying Iteration Start and End Dates. This action is performed on the Iterations page of the system.</p>  |
| 3.1.1.8.3      | <p>Iteration numbers are system generated and are automatically displayed when the ACI adds a new iteration.</p>   |
| 3.1.1.8.4      | <p>The PGA system also provides a table of all existing iterations. This table allows the ACI to de-activate iterations. Once an iteration is deactivated, all requests associated with that iteration are removed from the user screens (both ACI and regular users).</p>   |

**3.2 NON-FUNCTIONAL REQUIREMENTS**

Non-functional requirements describe what the system must be and include user interface and performance requirements.

| ID | Description |
|----|-------------|
|----|-------------|

**3.2.0    User Interface Guidelines**

|              |   |
|--------------|---|
| <b>3.2.1</b> | <b><i>Network-accessible interface</i></b>  |
| 3.2.1.1      | System must be web-based and must be accessed from inside the network only.   |
| <b>3.2.2</b> | <b><i>Usability and Appearance are User-friendly</i></b>  |
| 3.2.2.1      | The system must provide an aesthetic display of the information.  |
| 3.2.2.2      | Same screens appear each time system is displayed.  |
| 3.2.2.3      | Uses standard GUI features (e.g. browser navigation). This interface is familiar to most users and allows quick comprehension of the data being displayed with minimal training.  |
| 3.2.2.4      | Application screens have consistent look and feel.  |
| 3.2.2.5      | Data formats are consistent throughout screens.   |
| 3.2.2.6      | Information and error messages are useful, accurate, meaningful, and correctly spelled.   |
| 3.2.2.7      | Unnecessary warnings do not appear. For example, two dialog boxes in succession do not appear regarding the same error.   |
| 3.2.2.8      | Input required fields checking will be performed for entered data.  |
| 3.2.2.9      | Operation: Runs (coexists) with other applications.   |
| 3.2.10       | Dialogs, Commands, and Options: No clipped dialogs or misspelled words; labels consistent with figures in user help.  |
| 3.2.11       | <p>Online Help is available:</p> <ul style="list-style-type: none"> <li>▪ Easy to use</li> <li>▪ Hypertext links jump to proper subject</li> <li>▪ Extensive and understandable</li> <li>▪ Spelling and grammar correct</li> <li>▪ Messages displayed only when needed (without being annoying).</li> </ul> |

| ID | Description |
|----|-------------|
|----|-------------|

**3.3    Performance Guidelines**

|       |   |
|-------|---|
| 3.3.1 | <b><i>The system must be accessible from a remote connection.</i></b> |
|-------|---|

| ID           | Description  |
|--------------|--|
| <b>3.3</b>   | <b><u>Performance Guidelines</u></b>   |
| 3.3.1.1      | This requirement is addressed by building the system that is installed on the QuantumPM internal network that provides authenticated access to all users.  |
| <b>3.3.2</b> | <b><i>The system must be easily accessible.</i></b>  |
| 3.3.2.1      | The PGA system will be available and accessible 24 hours, 7 days a week from any location that allows for connection to the network with the exception of scheduled maintenance and unexpected outage as outlined in the Service Level Agreement.  |
| <b>3.3.3</b> | <b><i>The system must process quickly.</i></b>   |
| 3.3.3.1      | Every attempt to reduce display time to the user will be made. However, it must be noted that the system display speed will vary depending on access media and current network performance.  |
| <b>3.3.4</b> | <b><i>The system must contain a limited number of defects in each severity.</i></b>  |
| 3.3.4.1      | <p>The system must contain no more than the following number of defects in each severity:</p> <ul style="list-style-type: none"> <li>▪ Critical – None. Critical severity is defined as a complete system failure.</li> <li>▪ High – None. High severity is defined as a major function of the system experiences a failure.</li> <li>▪ Medium – 1-5. A functionality non-threatening problem that does not cause a serious system dysfunction.</li> <li>▪ Low – 5-10. Low severity is defined a small part of the functionality (generally a cosmetic problem) is not as intended.</li> </ul> |



---

## 4.0 Phase 2.0 Requirements

### 4.1 Functional Requirements

| ID | Description |
|----|-------------|
|----|-------------|

#### **4.1.1 Phase 2 Functional Requirements**

|                |  |
|----------------|--|
| <b>4.1.1.1</b> | <b><i>Evaluation Mode</i></b>  |
| 4.1.1.1.1      | The PGA System supports two types of evaluation modes: <ul style="list-style-type: none"><li>▪ Requested Evaluation</li><li>▪ Ad hoc Evaluation</li></ul>                            |
| 4.1.1.1.2      | Ad hoc Evaluation follows the Ad Hoc Evaluation Process as shown in Figure 2.  |
| <b>4.1.1.2</b> | <b><i>Evaluation Form</i></b>  |
| 4.1.1.2.1      | A different Evaluation Form is displayed for each evaluation mode.   |
| 4.1.1.2.2      | The Ad Hoc Evaluation Form provides a drop with a dynamic list of all QPM employees to select for evaluation.  |
| 4.1.1.2.3      | The Ad Hoc Evaluation Form must include a short list of questions that will be provided by QuantumPM.  |
| 4.1.1.3.4      | The Evaluation Form provides a Submit button. Upon clicking the Submit button a confirmation is displayed, allowing user to click the Edit or Continue buttons.                      |
| 4.1.1.3.5      | Upon clicking the Edit button, user is returned to the previous screen, allowing further editing of the information.   |
| 4.1.1.3.6      | Upon clicking the Continue button, the evaluation is submitted to the database, and a confirmation for successful submission is displayed to the user.                               |
| 4.1.1.3.7      | A summary of Evaluator answers is provided at the top of each completed evaluation form to indicate the number of answers in each range, separated by the section of the evaluation. |
| <b>4.1.1.3</b> | <b><i>Edit Mechanism</i></b>   |
| 4.1.1.3.1      | The Evaluation Edit screen allows ACI to modify any information displayed in the form and submit it to the database upon clicking the Submit button.                                 |

| ID             | Description  |
|----------------|--|
| <b>4.1.1</b>   | <b><u>Phase 2 Functional Requirements</u></b>  |
| 4.1.1.3.2      | Upon clicking the Submit button a confirmation is displayed, allowing user to click the Edit or Continue buttons.  |
| 4.1.1.3.3      | Upon clicking the Edit button, user is returned to the previous screen, allowing further editing of the information.   |
| 4.1.1.3.4      | Upon clicking the Continue button, the evaluation is submitted to the database, and a confirmation for successful submission is displayed to the user.   |
| <b>4.1.1.6</b> | <b><i>ACI Reports</i></b>  |
| 4.1.1.6.1      | <p>The PGA System must provide an additional, 4<sup>th</sup> ACI report:</p> <ul style="list-style-type: none"> <li>▪ Evaluations Out of Bounds</li> </ul>   |
| 4.1.1.6.2      | The Evaluations Out of Bounds report must calculate on a formula the number of questions in each section of the evaluation form that are out of normal bounds (the bounds to be determined by the SME at a later time). This report is also included as a summary at the top of each evaluation form, allowing the ACI to view at a glance the evaluation results. |

---

## 5.0 Phase 1.0 PGA System Design

The PGA System Design section provides some high level system design information such as the project schedule, database design, user interface design, encryption design, and system interaction flow.

---

### Note

This section has been modified to incorporate the latest design changes.

---

### 5.1 PGA System Project Schedule

The PGA System Project Schedule was created to outline the tasks involved in full system design, starting with requirements and continuing to testing. The System Project Schedule is meant to not only set out the amount of work and specific dates for each task within the project, but to also demonstrate the constraints and relationships between tasks which dictate the order in which the activities have to be carried out.

---

### Legend

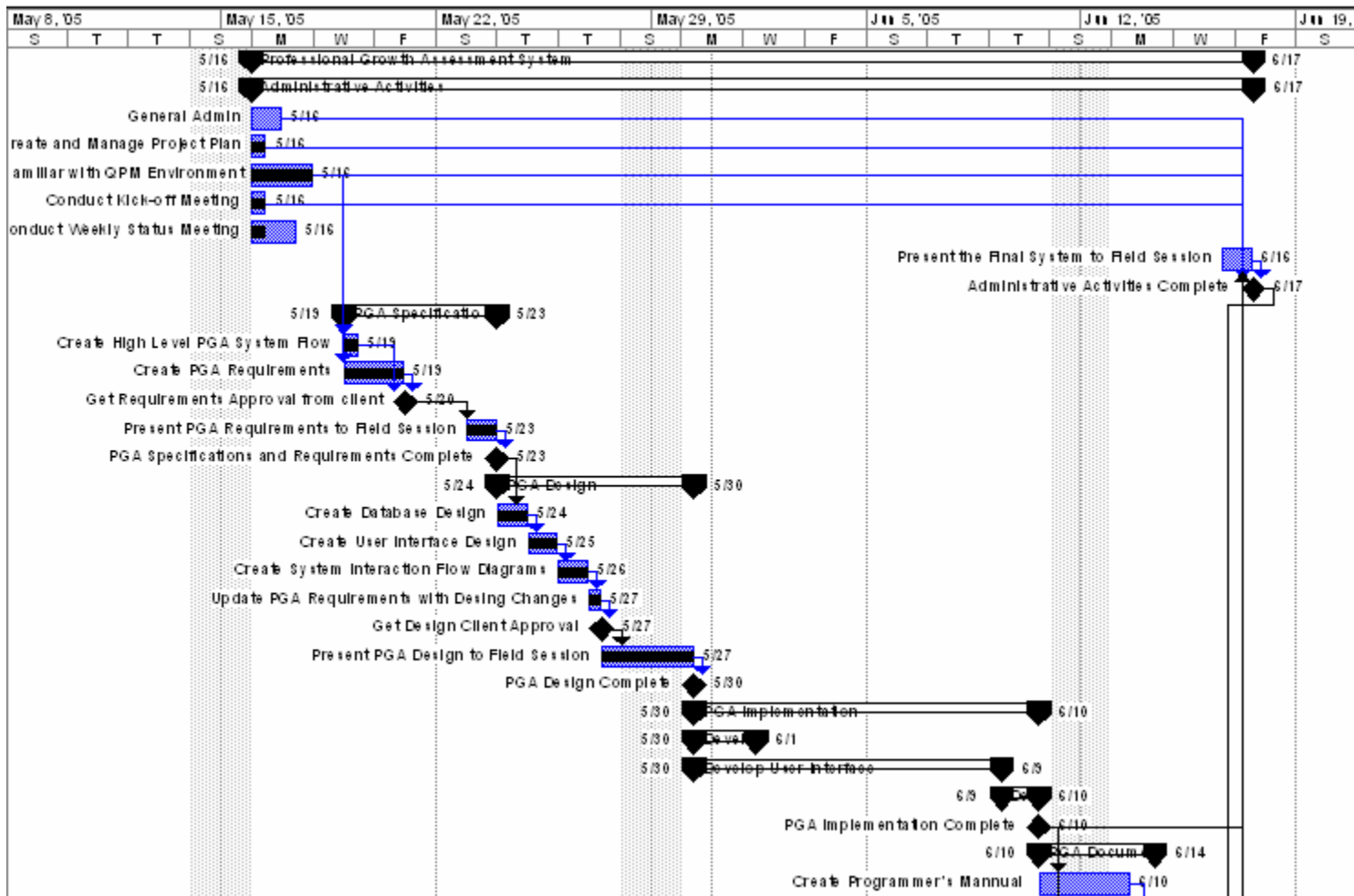
- Summary Tasks – Black Bold font
  - Critical Path Tasks – Indigo font
  - Milestone Tasks – Green Italics font
-

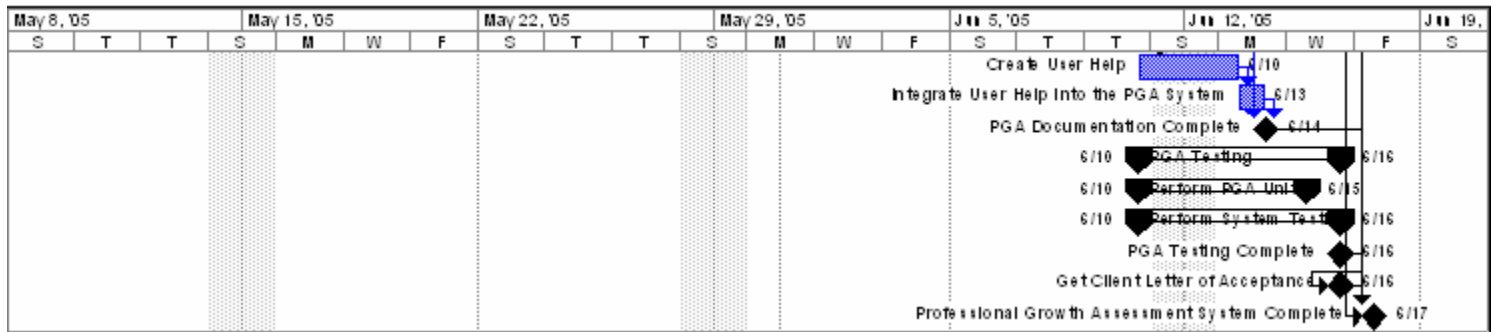
### 5.1.1 PGA Work Breakdown Structure

| ID | Task Name   | Duration          | Start              | Finish             | Predecessors        |
|----|---|-------------------|--------------------|--------------------|---------------------|
| 1  | <b>Professional Growth Assessment System</b>          | <b>24.75 days</b> | <b>Mon 5/16/05</b> | <b>Fri 6/17/05</b> |                     |
| 2  | <b>Administrative Activities</b>                      | <b>24.75 days</b> | <b>Mon 5/16/05</b> | <b>Fri 6/17/05</b> |                     |
| 3  | General Admin   | 1 day             | Mon 5/16/05        | Mon 5/16/05        |                     |
| 4  | ✓ Create and Manage Project Plan                      | 4 hrs             | Mon 5/16/05        | Mon 5/16/05        |                     |
| 5  | ✓ Get Familiar with QPM Environment                   | 2 days            | Mon 5/16/05        | Tue 5/17/05        |                     |
| 6  | ✓ Conduct Kick-off Meeting                            | 4 hrs             | Mon 5/16/05        | Mon 5/16/05        |                     |
| 7  | Conduct Weekly Status Meeting                         | 12 hrs            | Mon 5/16/05        | Tue 5/17/05        |                     |
| 8  | Present the Final System to Field Session             | 1 day             | Thu 6/16/05        | Fri 6/17/05        | 55                  |
| 9  | <i>Administrative Activities Complete</i>             | <i>0 days</i>     | <i>Fri 6/17/05</i> | <i>Fri 6/17/05</i> | <i>3,4,5,6,8</i>    |
| 10 | ✓ <b>PGA Specifications and Requirements</b>          | <b>3 days</b>     | <b>Thu 5/19/05</b> | <b>Mon 5/23/05</b> |                     |
| 11 | ✓ Create High Level PGA System Flow                   | 0.5 days          | Thu 5/19/05        | Thu 5/19/05        | 5FS+1 day,8FS+1 day |
| 12 | ✓ Create PGA Requirements                             | 2 days            | Thu 5/19/05        | Fri 5/20/05        | 5FS+1 day,8FS+1 day |
| 13 | ✓ <i>Get Requirements Approval from client</i>        | <i>0 days</i>     | <i>Fri 5/20/05</i> | <i>Fri 5/20/05</i> | <i>12,11</i>        |
| 14 | ✓ Present PGA Requirements to Field Session           | 1 day             | Mon 5/23/05        | Mon 5/23/05        | 13                  |
| 15 | ✓ <i>PGA Specifications and Requirements Complete</i> | <i>0 days</i>     | <i>Mon 5/23/05</i> | <i>Mon 5/23/05</i> | <i>14</i>           |
| 16 | <b>PGA Design</b>                                     | <b>4.25 days</b>  | <b>Tue 5/24/05</b> | <b>Mon 5/30/05</b> |                     |
| 17 | ✓ Create Database Design                              | 1 day             | Tue 5/24/05        | Tue 5/24/05        | 15                  |
| 18 | ✓ Create User Interface Design                        | 1 day             | Wed 5/25/05        | Wed 5/25/05        | 17                  |
| 19 | ✓ Create System Interaction Flow Diagrams             | 1 day             | Thu 5/26/05        | Thu 5/26/05        | 18                  |
| 20 | ✓ Update PGA Requirements with Design Changes         | 2 hrs             | Fri 5/27/05        | Fri 5/27/05        | 19                  |
| 21 | <i>Get Design Client Approval</i>                     | <i>0 days</i>     | <i>Fri 5/27/05</i> | <i>Fri 5/27/05</i> | <i>20</i>           |
| 22 | ✓ Present PGA Design to Field Session                 | 1 day             | Fri 5/27/05        | Mon 5/30/05        | 21                  |
| 23 | ✓ <i>PGA Design Complete</i>                          | <i>0 days</i>     | <i>Mon 5/30/05</i> | <i>Mon 5/30/05</i> | <i>22</i>           |
| 24 | <b>PGA Implementation</b>                             | <b>9.5 days</b>   | <b>Mon 5/30/05</b> | <b>Fri 6/10/05</b> |                     |
| 25 | <b>Develop Database</b>                               | <b>2 days</b>     | <b>Mon 5/30/05</b> | <b>Wed 6/1/05</b>  |                     |
| 26 | Develop Database Tables                               | 1 day             | Mon 5/30/05        | Tue 5/31/05        | 23                  |
| 27 | Develop Database Queries                              | 1 day             | Tue 5/31/05        | Wed 6/1/05         | 26                  |
| 28 | <i>Develop Database Complete</i>                      | <i>0 days</i>     | <i>Wed 6/1/05</i>  | <i>Wed 6/1/05</i>  | <i>27</i>           |

| ID | Task Name   | Duration        | Start              | Finish             | Predecessors  |
|----|---|-----------------|--------------------|--------------------|---------------|
| 29 | <b>Develop User Interface</b>                         | <b>8 days</b>   | <b>Mon 5/30/05</b> | <b>Thu 6/9/05</b>  |               |
| 30 | Develop InfoPath Evaluation Forms                     | 0.5 days        | Mon 5/30/05        | Mon 5/30/05        | 23            |
| 31 | Develop ASP.NET User Interface Screen Layout          | 4 days          | Wed 6/1/05         | Tue 6/7/05         | 28            |
| 32 | Develop Windows NT User Authentication                | 1 day           | Tue 6/7/05         | Wed 6/8/05         | 30,31         |
| 33 | Connect User Interface to Database                    | 1 day           | Wed 6/8/05         | Thu 6/9/05         | 28,32         |
| 34 | <i>Develop User Interface Complete</i>                | <i>0 days</i>   | <i>Thu 6/9/05</i>  | <i>Thu 6/9/05</i>  | 33            |
| 35 | <b>Develop Encryption</b>                             | <b>1.5 days</b> | <b>Thu 6/9/05</b>  | <b>Fri 6/10/05</b> |               |
| 36 | Develop Database Encryption                           | 1.5 days        | Thu 6/9/05         | Fri 6/10/05        | 34            |
| 37 | <i>Develop Encryption Complete</i>                    | <i>0 days</i>   | <i>Fri 6/10/05</i> | <i>Fri 6/10/05</i> | 36            |
| 38 | <i>PGA Implementation Complete</i>                    | <i>0 days</i>   | <i>Fri 6/10/05</i> | <i>Fri 6/10/05</i> | 28,34,37      |
| 39 | <b>PGA Documentation</b>                              | <b>1.5 days</b> | <b>Fri 6/10/05</b> | <b>Tue 6/14/05</b> |               |
| 40 | Create Programmer's Manual                            | 1 day           | Fri 6/10/05        | Mon 6/13/05        | 38            |
| 41 | Create User Help                                      | 1 day           | Fri 6/10/05        | Mon 6/13/05        | 38            |
| 42 | Integrate User Help into the PGA System               | 0.5 days        | Mon 6/13/05        | Tue 6/14/05        | 41            |
| 43 | <i>PGA Documentation Complete</i>                     | <i>0 days</i>   | <i>Tue 6/14/05</i> | <i>Tue 6/14/05</i> | 40,41,42      |
| 44 | <b>PGA Testing</b>                                    | <b>4 days</b>   | <b>Fri 6/10/05</b> | <b>Thu 6/16/05</b> |               |
| 45 | <b>Perform PGA Unit Tests</b>                         | <b>3 days</b>   | <b>Fri 6/10/05</b> | <b>Wed 6/15/05</b> |               |
| 46 | Perform Database Unit Testing                         | 1 day           | Fri 6/10/05        | Mon 6/13/05        | 38            |
| 47 | Perform User Interface Unit Testing                   | 1 day           | Mon 6/13/05        | Tue 6/14/05        | 46            |
| 48 | Perform Encryption Unit Testing                       | 1 day           | Tue 6/14/05        | Wed 6/15/05        | 47            |
| 49 | <i>Perform PGA Unit Tests Complete</i>                | <i>0 days</i>   | <i>Wed 6/15/05</i> | <i>Wed 6/15/05</i> | 46,47,48      |
| 50 | <b>Perform System Testing</b>                         | <b>4 days</b>   | <b>Fri 6/10/05</b> | <b>Thu 6/16/05</b> |               |
| 51 | Create System Test Cases                              | 1 day           | Fri 6/10/05        | Mon 6/13/05        | 38            |
| 52 | Perform Full PGA System Test                          | 3 days          | Mon 6/13/05        | Thu 6/16/05        | 51            |
| 53 | <i>Perform System Testing Complete</i>                | <i>0 days</i>   | <i>Thu 6/16/05</i> | <i>Thu 6/16/05</i> | 52            |
| 54 | <i>PGA Testing Complete</i>                           | <i>0 days</i>   | <i>Thu 6/16/05</i> | <i>Thu 6/16/05</i> | 49,53         |
| 55 | <i>Get Client Letter of Acceptance</i>                | <i>0 days</i>   | <i>Thu 6/16/05</i> | <i>Thu 6/16/05</i> | 54            |
| 56 | <i>Professional Growth Assessment System Complete</i> | <i>0 days</i>   | <i>Fri 6/17/05</i> | <i>Fri 6/17/05</i> | 55,9,43,54,38 |

### 5.1.2 PGA Gantt Chart View






### 5.1.3 PGA Critical Tasks List

| ID | Task Name  | Duration   | Start       | Finish      |
|----|--|------------|-------------|-------------|
| 1  | Professional Growth Assessment System                              | 24.75 days | Mon 5/16/05 | Fri 6/17/05 |
| 2  | Administrative Activities  | 24.75 days | Mon 5/16/05 | Fri 6/17/05 |
| 8  | Present the Final System to Field Session                          | 1 day      | Thu 6/16/05 | Fri 6/17/05 |
|    | <i>ID Successor Name Type Lag</i>                                  |            |             |             |
|    | <i>8 Administrative Activities Complete FS 0 days</i>              |            |             |             |
| 9  | Administrative Activities Complete                                 | 0 days     | Fri 6/17/05 | Fri 6/17/05 |
|    | <i>ID Successor Name Type Lag</i>                                  |            |             |             |
|    | <i>88 Professional Growth Assessment System Complete FS 0 days</i> |            |             |             |
| 24 | PGA Implementation   | 9.5 days   | Mon 5/30/05 | Fri 6/10/05 |
| 25 | Develop Database   | 2 days     | Mon 5/30/05 | Wed 6/1/05  |
| 29 | Develop User Interface   | 8 days     | Mon 5/30/05 | Thu 6/9/05  |
| 35 | Develop Encryption   | 1.5 days   | Thu 6/9/05  | Fri 6/10/05 |
| 38 | PGA Implementation Complete  | 0 days     | Fri 6/10/05 | Fri 6/10/05 |
|    | <i>ID Successor Name Type Lag</i>                                  |            |             |             |
|    | <i>40 Create Programmer's Manual FS 0 days</i>                     |            |             |             |
|    | <i>41 Create User Help FS 0 days</i>                               |            |             |             |
|    | <i>46 Perform Database Unit Testing FS 0 days</i>                  |            |             |             |
|    | <i>61 Create System Test Cases FS 0 days</i>                       |            |             |             |
|    | <i>66 Professional Growth Assessment System Complete FS 0 days</i> |            |             |             |
| 44 | PGA Testing  | 4 days     | Fri 6/10/05 | Thu 6/16/05 |
| 50 | Perform System Testing   | 4 days     | Fri 6/10/05 | Thu 6/16/05 |
| 54 | PGA Testing Complete   | 0 days     | Thu 6/16/05 | Thu 6/16/05 |
|    | <i>ID Successor Name Type Lag</i>                                  |            |             |             |
|    | <i>68 Get Client Letter of Acceptance FS 0 days</i>                |            |             |             |
|    | <i>66 Professional Growth Assessment System Complete FS 0 days</i> |            |             |             |
| 55 | Get Client Letter of Acceptance                                    | 0 days     | Thu 6/16/05 | Thu 6/16/05 |
|    | <i>ID Successor Name Type Lag</i>                                  |            |             |             |
|    | <i>8 Present the Final System to Field Session FS 0 days</i>       |            |             |             |
|    | <i>66 Professional Growth Assessment System Complete FS 0 days</i> |            |             |             |
| 56 | Professional Growth Assessment System Complete                     | 0 days     | Fri 6/17/05 | Fri 6/17/05 |



### 5.1.4 PGA System Milestones

| ID |  | Task Name                                    | Duration   | Start       | Finish      |
|----|---|--|------------|-------------|-------------|
| 1  |   | Professional Growth Assessment System        | 24.75 days | Mon 5/16/05 | Fri 6/17/05 |
| 2  |   | Administrative Activities                    | 24.75 days | Mon 5/16/05 | Fri 6/17/05 |
| 10 | ✓   | PGA Specifications and Requirements          | 3 days     | Thu 5/19/05 | Mon 5/23/05 |
| 13 | ✓   | Get Requirements Approval from client        | 0 days     | Fri 5/20/05 | Fri 5/20/05 |
| 15 | ✓   | PGA Specifications and Requirements Complete | 0 days     | Mon 5/23/05 | Mon 5/23/05 |
| 16 |   | PGA Design                                   | 4.25 days  | Tue 5/24/05 | Mon 5/30/05 |
| 21 |   | Get Design Client Approval                   | 0 days     | Fri 5/27/05 | Fri 5/27/05 |
| 23 | ✓   | PGA Design Complete                          | 0 days     | Mon 5/30/05 | Mon 5/30/05 |
| 24 |   | PGA Implementation                           | 9.5 days   | Mon 5/30/05 | Fri 6/10/05 |
| 25 |   | Develop Database                             | 2 days     | Mon 5/30/05 | Wed 6/1/05  |
| 29 |   | Develop User Interface                       | 8 days     | Mon 5/30/05 | Thu 6/9/05  |
| 35 |   | Develop Encryption                           | 1.5 days   | Thu 6/9/05  | Fri 6/10/05 |
| 38 |   | PGA Implementation Complete                  | 0 days     | Fri 6/10/05 | Fri 6/10/05 |
| 39 |   | PGA Documentation                            | 1.5 days   | Fri 6/10/05 | Tue 6/14/05 |
| 44 |   | PGA Testing                                  | 4 days     | Fri 6/10/05 | Thu 6/16/05 |
| 45 |   | Perform PGA Unit Tests                       | 3 days     | Fri 6/10/05 | Wed 6/15/05 |
| 50 |   | Perform System Testing                       | 4 days     | Fri 6/10/05 | Thu 6/16/05 |
| 43 |   | PGA Documentation Complete                   | 0 days     | Tue 6/14/05 | Tue 6/14/05 |
| 54 |   | PGA Testing Complete                         | 0 days     | Thu 6/16/05 | Thu 6/16/05 |
| 55 |   | Get Client Letter of Acceptance              | 0 days     | Thu 6/16/05 | Thu 6/16/05 |
| 9  |   | Administrative Activities Complete           | 0 days     | Fri 6/17/05 | Fri 6/17/05 |
| 56 |   | Professional Growth Assessment System Comple | 0 days     | Fri 6/17/05 | Fri 6/17/05 |

## 5.2 Database Design

This section outlines in detail the database model for the PGA System. These include:

- Database Schema – the overall layout of the database tables, dependencies, and constraints.
- Database Stored Procedures – lists all stored procedures used for inserting, updating, and querying information stored in the database.

### 5.2.1 Database Details

This section provides a schema of the PGA System Database.

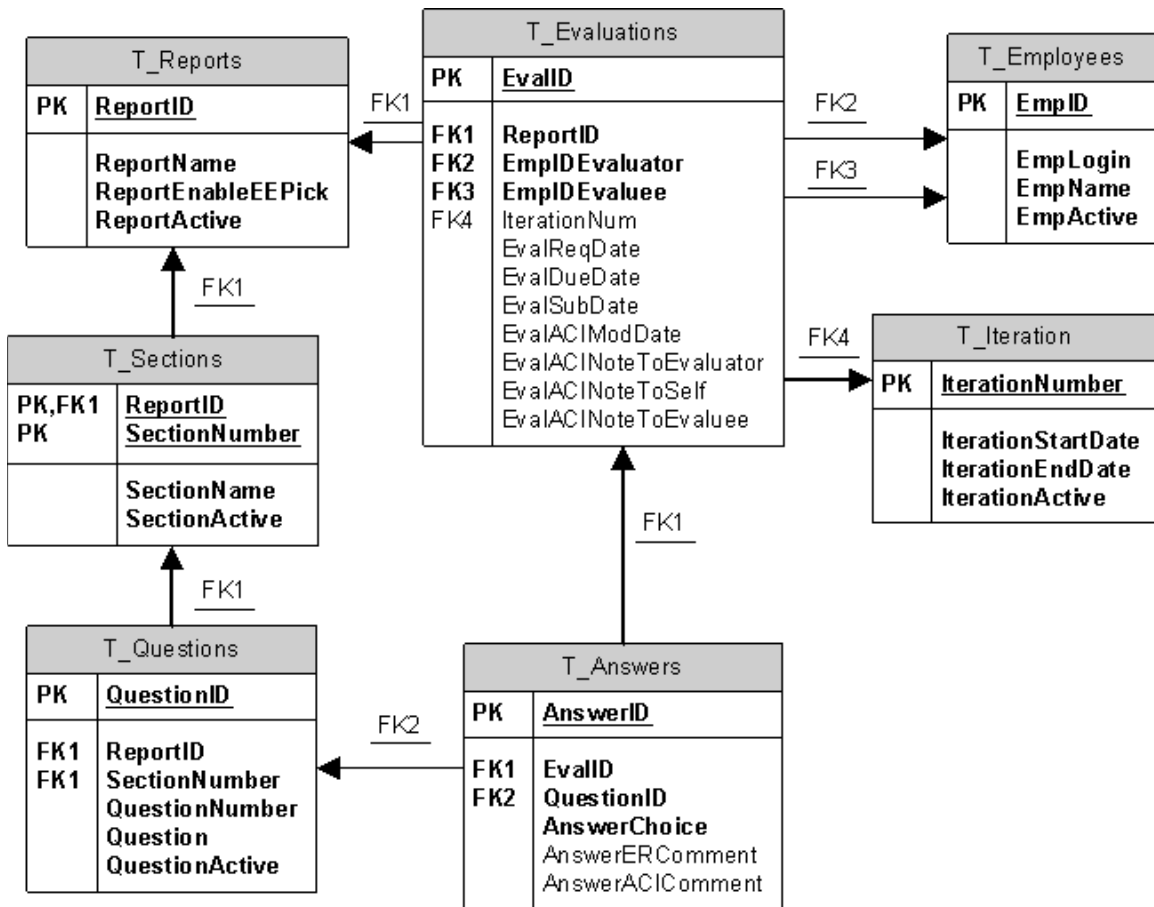


Figure 3 Database ER Diagram

The PGA system database includes 7 tables:

- **T\_Reports** – the Reports table is designed to specify the type of evaluation in the PGA system. In Phase 1.0, the PGA system supports only one type of evaluations – Requested. This table was designed specifically to support the future growth of the system by providing ability to add new evaluation types that are linked to the rest of the tables in the database. The Report table contains the following fields:

- ReportID – (integer) Unique incrementing identifier.
- ReportName – (varchar) Name/Type of report.
- ReportEnableEEPick – (bit) 1 = Yes, the report allows the Evaluator to select the Evaluatee from a drop down menu.
- ReportActive – (bit) 1 = Yes, employee is a current employee; 0 = No, the employee is no longer employed.
- **T\_Evaluations** – the Evaluations table is designed to contain information about each request submitted by the ACI and the evaluation information associated with it, such as evaluation ID, Evaluatee ID, Evaluator ID, and so on. The Evaluations table was combined with the Requests table to save space and keep track of evaluation information more efficiently. The Evaluations table contains the following fields:
  - EvalID – (integer) Unique incrementing identifier.
  - ReportID – (integer) ID of report this evaluation uses.
  - EmpIDEvaluator – (integer) ID of evaluation's Evaluator.
  - EmpIDEvaluatee – (integer) ID of evaluation's Evaluatee.
  - IterationNumber – (integer) Evaluation iteration number.
  - EvalReqDate – (smalldatetime) Date evaluation requested by ACI.
  - EvalDueDate – (smalldatetime) Date evaluation due from Evaluator.
  - EvalSubDate – (smalldatetime) Date evaluation submitted to ACI by Evaluator.
  - EvalACIModDate – (smalldatetime) Date evaluation submitted to Evaluatee by ACI.
  - EvalACIComment – (varchar) Text instructions/comments from ACI to Evaluators.
- **T\_Employees** – the Employees table is designed to contain the list of all QuantumPM employees, including their Windows Authentication login and full name. Within the database, each employee is referenced by employee ID, which is unique to the PGA system. The Employees table contains the following fields:
  - EmpID – (integer) Unique incrementing identifier.
  - EmpLogin – (varchar) Windows Authentication login.
  - EmpName – (varchar) Employee's full name.
  - EmpActive – (bit) 1 = Yes, employee is a current employee; 0 = No, the employee is no longer employed.
- **T\_Sections** – the Section tables is designed to contain the different sections of the evaluations. Each evaluation is broken down into areas of interest, e.g. Communication Skills. The sections are meant to distinguish the areas of interest and store the information related to each. The Sections table contains the following fields:
  - SectionID – (integer) Unique incrementing identifier.
  - ReportID – (integer) ID of report this evaluation uses.

- SectionNumber – (integer) Number of section.
- SectionName – (varchar) Name of section.
- SectionActive – (bit) 1 = Yes, section will be shown on new evaluations; 0 = No, the section will not be shown on new evaluations.
- **T\_Iterations** – the Iterations table is designed to contain all scheduled iterations used for Requested Evaluations. Iterations represent time frames between which each evaluation period takes place. The Iterations table contains the following fields:
  - IterationNumber – (integer) Unique value of iteration.
  - IterationBeginDate – (smalldatetime) Beginning date of iteration.
  - IterationEndDate – (smalldatetime) Ending date of iteration.
  - IterationActive - (bit) 1 = Yes, specifies whether the iteration is active or has been deactivated by the ACI.
- **T\_Questions** – the Questions table is designed to contain all questions used for evaluations. This table is not specific to any evaluation type and allows specification of the evaluation type that the question belongs to. The Questions table contains the following fields:
  - QuestionID – (integer) Unique incrementing identifier.
  - SectionID – (varchar) ID of section this question belongs to.
  - QuestionNumber – (integer) Number of question.
  - Question – (varchar) Question text.
  - QuestionActive – (bit) 1 = Yes, question will be shown on new evaluations; 0 = No, the question will not be shown on new evaluations.
- **T\_Answers** – the Answers table is designed to contain answers to all evaluations submitted by users. The Answers table contains the following fields:
  - AnswerID – (integer) Unique incrementing identifier.
  - EvalID – (integer) ID of evaluation this answer belongs to.
  - QuestionID – (integer) ID of question this answer answers.
  - AnswerChoice – (integer) Answer to multiple choice Evaluatee rating.
  - AnswerERComment – (varchar) Text answer from Evaluator.
  - AnswerACIComment – (varchar) Modified text answer from ACI.

## 5.2.2 Database Stored Procedures

The Database Stored Procedures section contains the list and description of all stored procedures used to insert, update, and query data contained in the database.

Below is the list of all stored procedures used in the system:

- P\_ACIEvalsToModify - selects all evaluations that do not have an ACIModDate (Date ACI reviews the evaluation) but do have an EvalSubDate (Evaluation was completed by the Evaluator).

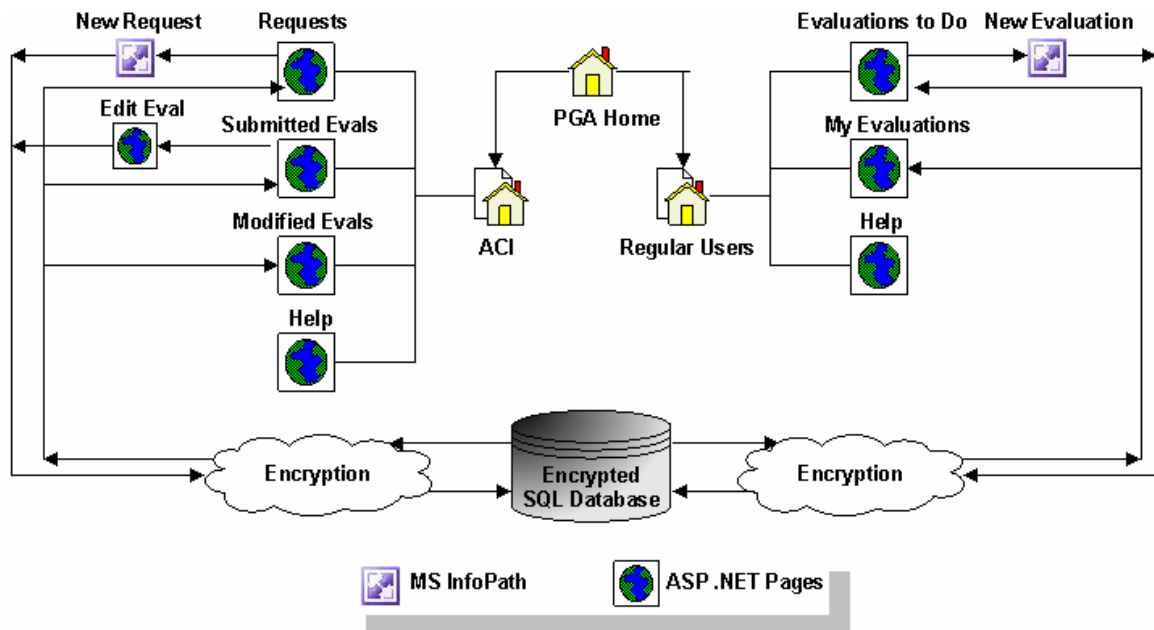
- P\_ActivateIteration - sets the activation status of an iteration to Active.
- P\_AddEmployee - takes a username and employee name and inserts it into the T\_Employees table.
- P\_AllEmployees – returns all employees currently registered in the system from the T\_Employee table.
- P\_CheckEmployee - takes a user name and checks to see if the user name is already in the T\_Employees table.
- P\_CheckEvaluation - takes an EvalID and ReportID and checks to see if all the evaluations questions have been answered by returning a list of the evaluations' QuestionIDs that have not been answered.
- P\_CheckSection - takes an EvalID, ReportID, and SectionNumber and checks to see if all the section's questions have been answered by returning a list of questions that do not have an answer.
- P\_EvalsToDo - takes a username of an evaluator and returns all the EvalID's of evaluations in an active iteration that have not been completed by checking if the EvalSubDate is NULL.
- P\_GetCurrentIteration - gets the latest iteration by selecting the greatest IterationNumber.
- P\_GetEvalType - takes a ReportName and returns the ReportID.
- P\_GetEvaluationInfo - takes an EvalID and returns related fields of the evaluation.
- P\_GetEvaluatorID - retrieves the evaluator ID from the T\_Employees table based on the Evaluator selected in the Evaluation Request InfoPath form. The Evaluator ID is required in order to create a new evaluation request.
- P\_GetEvaluateeID - retrieves the Evaluatee ID from the T\_Employees table based on the Evaluatee selected in the Evaluation Request InfoPath form. The Evaluatee ID is required in order to create a new evaluation request.
- P\_GetEvaluateeName - takes an EvalID and returns the Evaluatee's name.
- P\_GetHistory - returns all EvalID's where the evaluation has been completed by the evaluator and reviewed by the ACI.
- P\_GetIterationActive - takes an IterationNumber and returns 1 if the iteration is active, 0 if the iteration is not active.
- P\_GetIterations - returns all the iterations in descending order.
- P\_GetQuestionsAndAnswersBySection - takes an EvalID, ReportID, and SectionNumber and returns all questions and corresponding answers for this section.
- P\_GetRecentRequests - returns the last 25 evaluation requests with active Iteration numbers.
- P\_GetSectionName - takes a ReportID and SectionNumber and returns the Section Name.
- P\_InsertRequest - takes the Iteration Number, Request Date, Due Date, ACI Comment to Evaluator, Evaluator ID, Evaluatee ID, and Report ID values entered in

the InfoPath Evaluation Request and creates a new evaluation entry in the T\_Evaluations table.

- P\_MyEvaluations - takes a username and selects all EvalIDs with an active IterationNumber.
- P\_NewIteration - takes a Start and End dates and creates a new iteration in T\_Iterations table.
- P\_SaveACIComment - saves and encrypts ACI comment in the T\_Answers table.
- P\_SetSubDate - enters the SubDate into the EvalID after an evaluator has completed the evaluation.
- P\_SubmitAnswer - takes an EvalID, QuestionID, Answer, and Comment and either inserts or updates the T\_Answers table.
- P\_SubmitRequest - takes the request fields and creates a new evaluation in T\_Evaluation.
- P\_SubmitToEvaluatee - stored procedures takes the evaluation modified by the ACI and submits it to the Evaluatee by updating the ACI Modified Date in the T\_Evaluations table.
- P\_UpdateACINoteToSelf - takes an EvalID and an ACIComment and updates the ACIComment field in the evaluation.
- P\_UpdateIterationActive - takes an IteratioNumber and flips the IterationActive field from true to false, or from false to true.

### 5.3 System Interaction

The System Interaction section includes the diagram that depicts interaction between different parts of the system. This includes the communication between the MS SQL Server Database, ASP .NET User Interface, InfoPath form construction, and encryption.



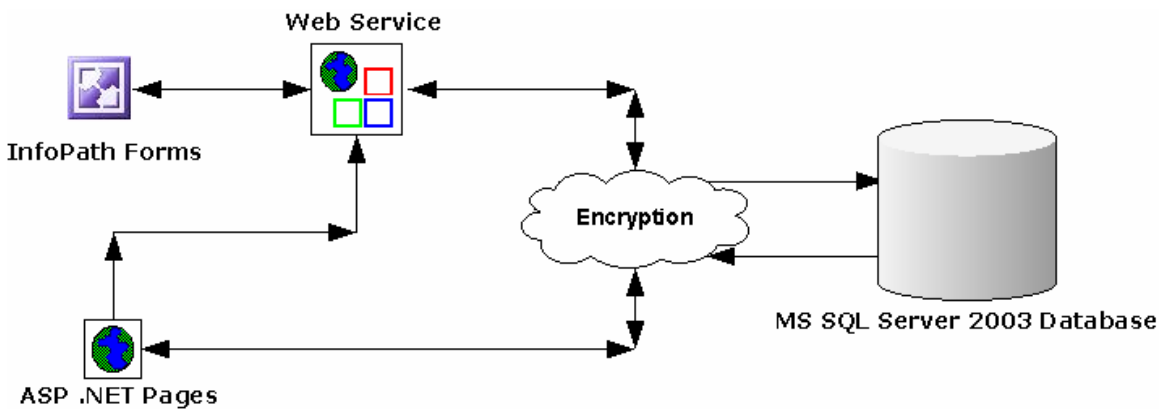
**Figure 4 Full System Interaction Diagram**

## 5.4 Encryption Design

The encryption design for the PGA System is a data level encryption based on an Encrypt() function.

When the user submits answers to the data to the database, it is added to the appropriate tables using the Encrypt() function. The same function is used again when the data is retrieved by the user.

Figure 5 below presents a high-level encryption design.



**Figure 5 Encryption Design Diagram**

Since the function is available to everyone and does not use the private/public key concept for encrypting data, the function itself needs to be encrypted. SQL offers a tag "WITH ENCRYPTION" that will be used to hide details of the function on how it encrypts the data. In addition, to increase the security, the function is passed a password so that it may not be used by an unauthorized user. This password will only be visible in the code before it is compiled. This will allow programmers that will be extending the system in the future to have access to the encryption, but it is protected from everyone else who decides to look at the system.

## 5.5 User Interface Design

The User Interface is a new requirement that became known during the Design Phase of the project. The Field Session Team and QPM have discovered that InfoPath technology will be unable to support the user interface that the system functionality calls for. In order to adapt to this technological constraint, QPM has requested that a stand-alone user interface to be implemented to support that functionality.

Client's newest requirement for the User Interface implementation is to use ASP .NET, which is a new technology that none of the team members are familiar with. Due to the lack of expertise in ASP .NET, QPM is going to waive some of the requirements (marked with asterisk in the Phase 1.0 Requirements section) in order to allow for the new programming language learning curve.

This section provides screen mockups and an example of the evaluation form to demonstrate the design of the user interface.

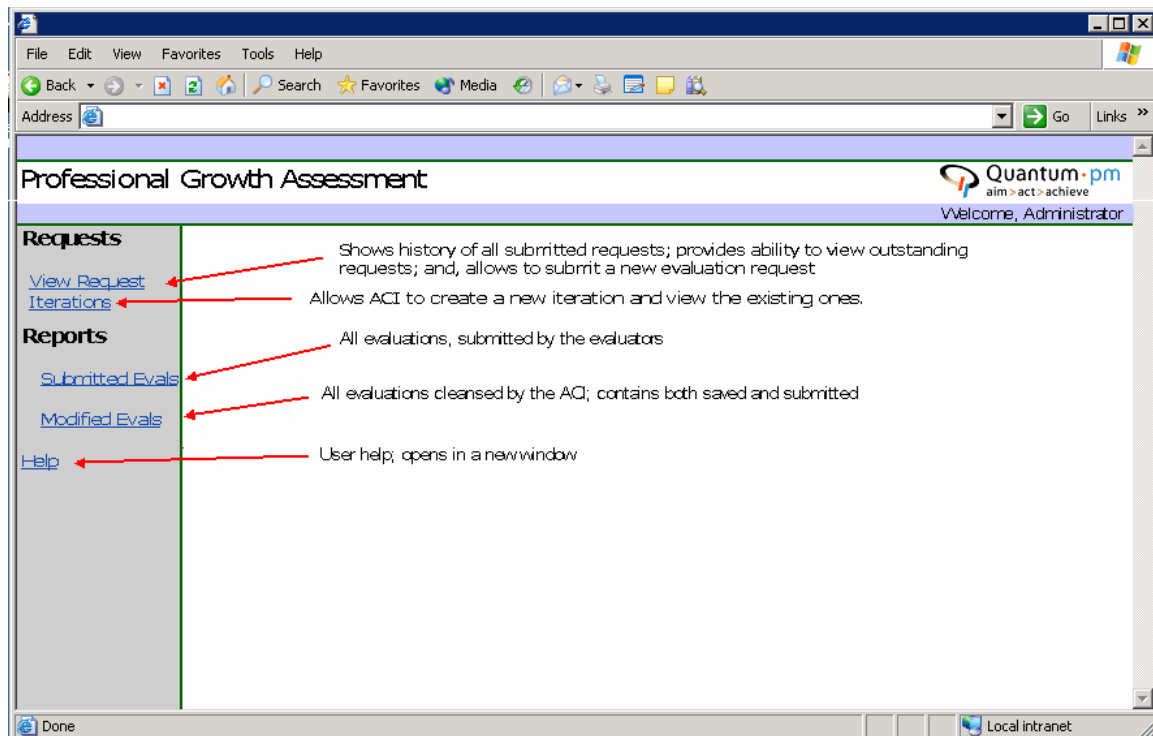
### Note

Screen mockups are meant to provide an example of system UI and may not be implemented exactly as pictured.

Figure 6 below shows an example of an ACI main screen. The left side of the screen provides a list of links to reports and other functionality that ACI is allowed to view (for more details, see Phase 1.0 Requirements section).

### 5.5.1 ACI UI Screens

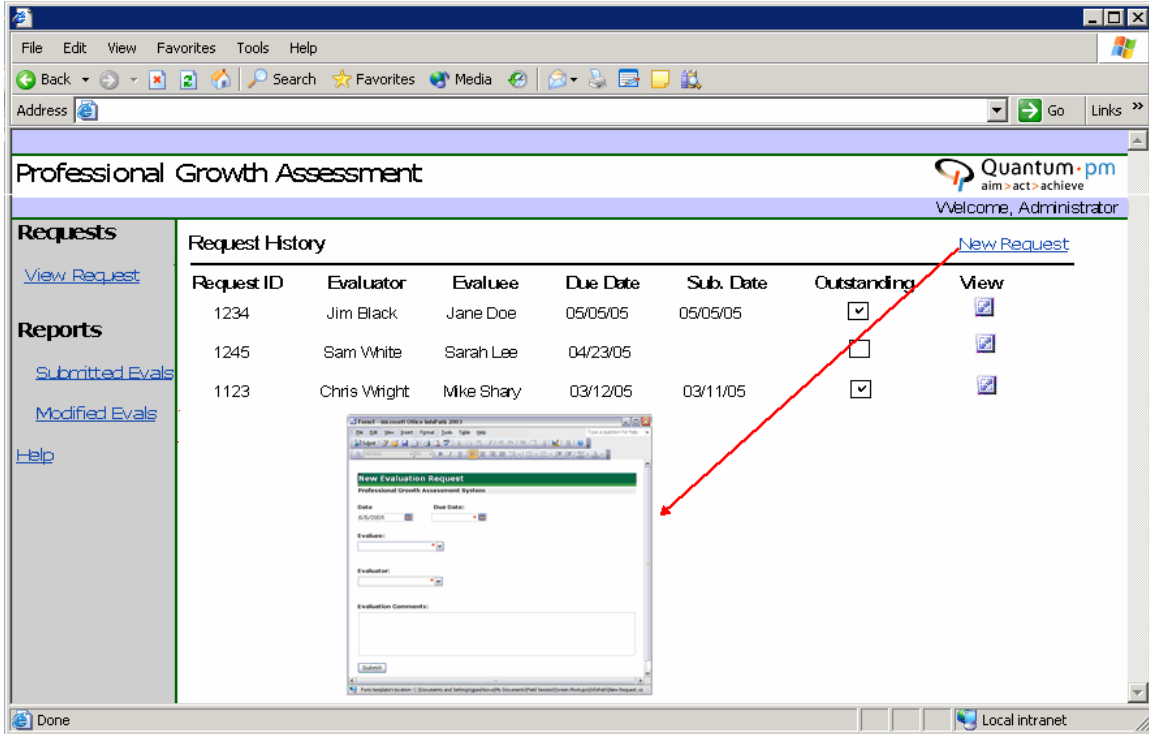
This section contains the mockup screens the ACI user will see, when accessing the PGA system.





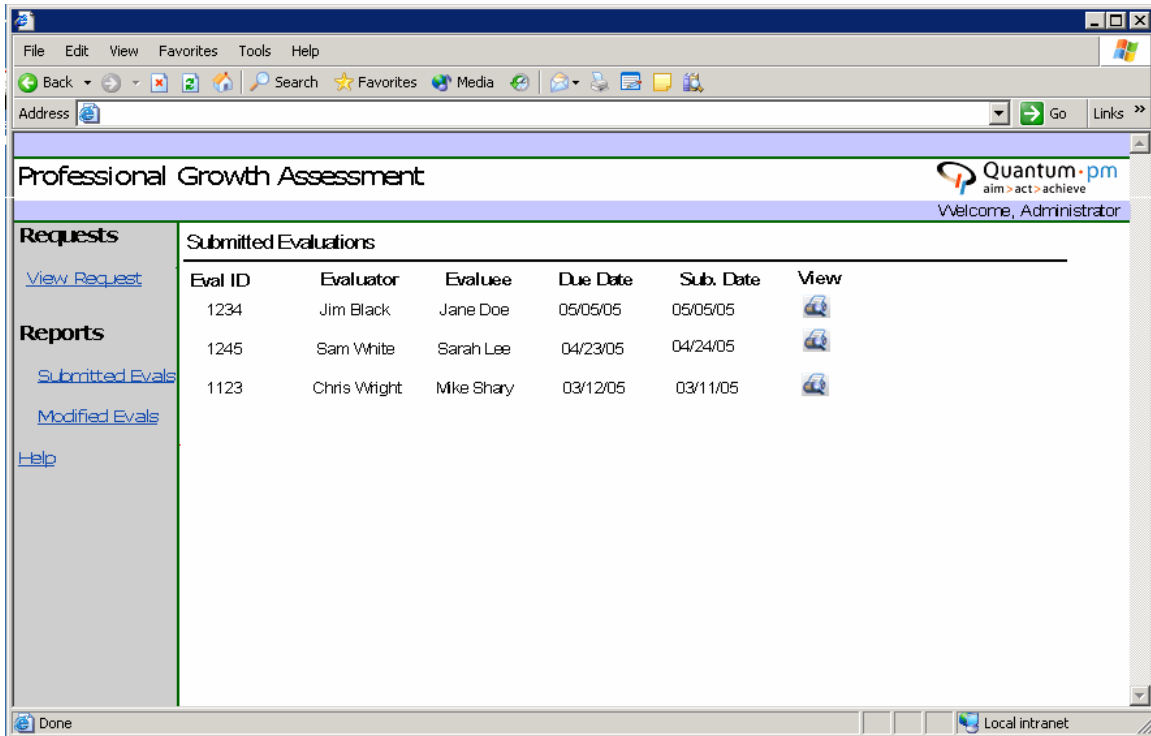
**Figure 6 PGA ACI Main Screen**

Figure 6 shows the PGA system, when ACI first logs in. The links on the left hand side represent various actions that the ACI user is able to perform inside the PGA System.



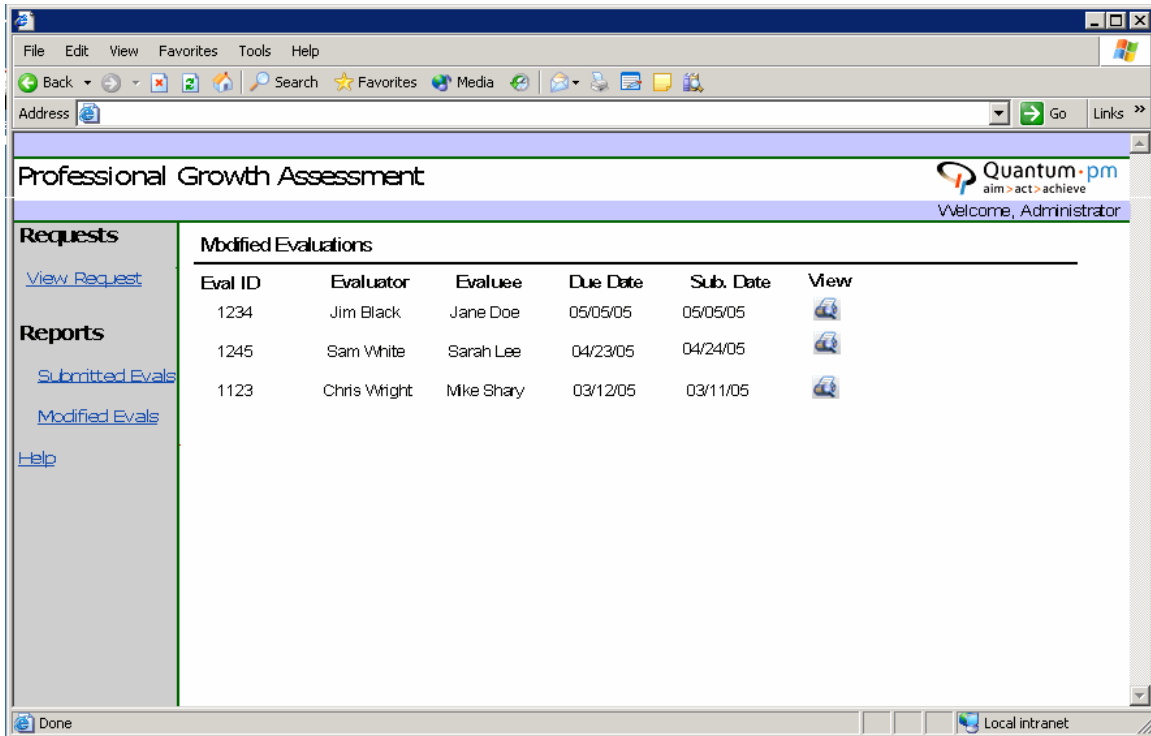
**Figure 7 PGA ACI Request Screen**

Figure 7 shows the Submit New Request page, which can be accessed by the ACI by selecting the View Requests link in the left side pane. Once the ACI user clicks the InfoPath icon to submit a new request, a new InfoPath form opens, allowing him/her to fill out the new request information. This screen also shows a table with all submitted requests. The requests appearing on this page are all of the requests from the current iteration. As soon as the iteration Finish Date expires, all requests associated with that iteration are automatically removed from this screen. The Request History table lists the Request ID, Evaluator, Evalued, Due Date, Submission Date, checkmark whether the request is outstanding or not, and ability to view the original request sent to the Evaluator. Once the ACI user clicks on the request under View, an evaluation request opens as read-only, allowing the ACI user to view the original request.



**Figure 8 PGA ACI Submitted Evaluations Screen**

Figure 8 shows a mockup of the Submitted Evaluations screen. The ACI user can access the Submitted Evaluations page by selecting Submitted Evals in the left side pane. The Submitted Evaluations page lists the Evaluation ID, Evaluator, Evaluatee, Due Date, Submission Date, and also provides ability to View each evaluation. All evaluations listed on this screen have been submitted by the Evaluators, but have not yet been sent to the Evaluatees. This screen also shows evaluations that the ACI user started to modify, but have not yet submitted to the database. All evaluations are filtered by the Submission Date (most recent at the top).

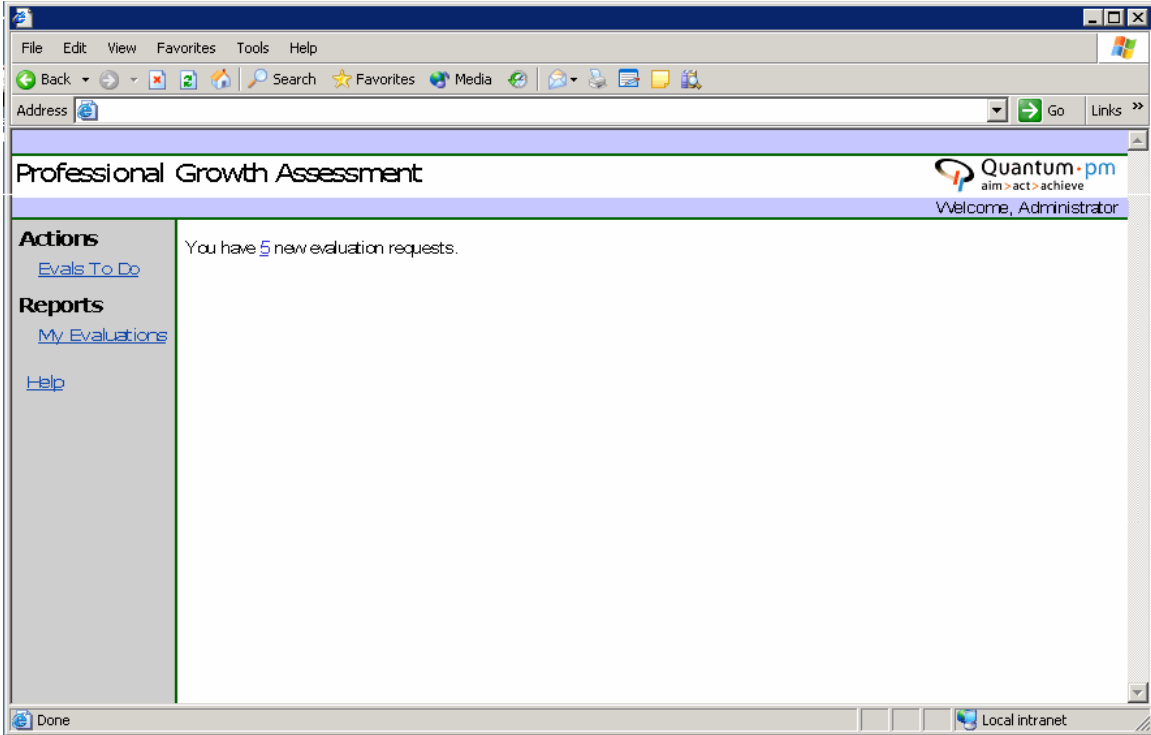


**Figure 9 PGA ACI Modified Evaluations Screen**

Figure 9 shows a mockup of the Modified Evaluations page. The Modified Evaluations page can be accessed by the ACI user by selecting Modified Evals in the left side pane. The Modified Evaluations page lists the Evaluation ID, Evaluator, Evaluatee, Due Date, Submission Date, and also provides ability to view the evaluation. Once the evaluations have been submitted to the Evaluatee, the system does not allow any more changes. When the ACI selects the View icon, a read-only InfoPath evaluation form is displayed. The list of evaluations is filtered by the submission date, with the most recent displayed at the top.

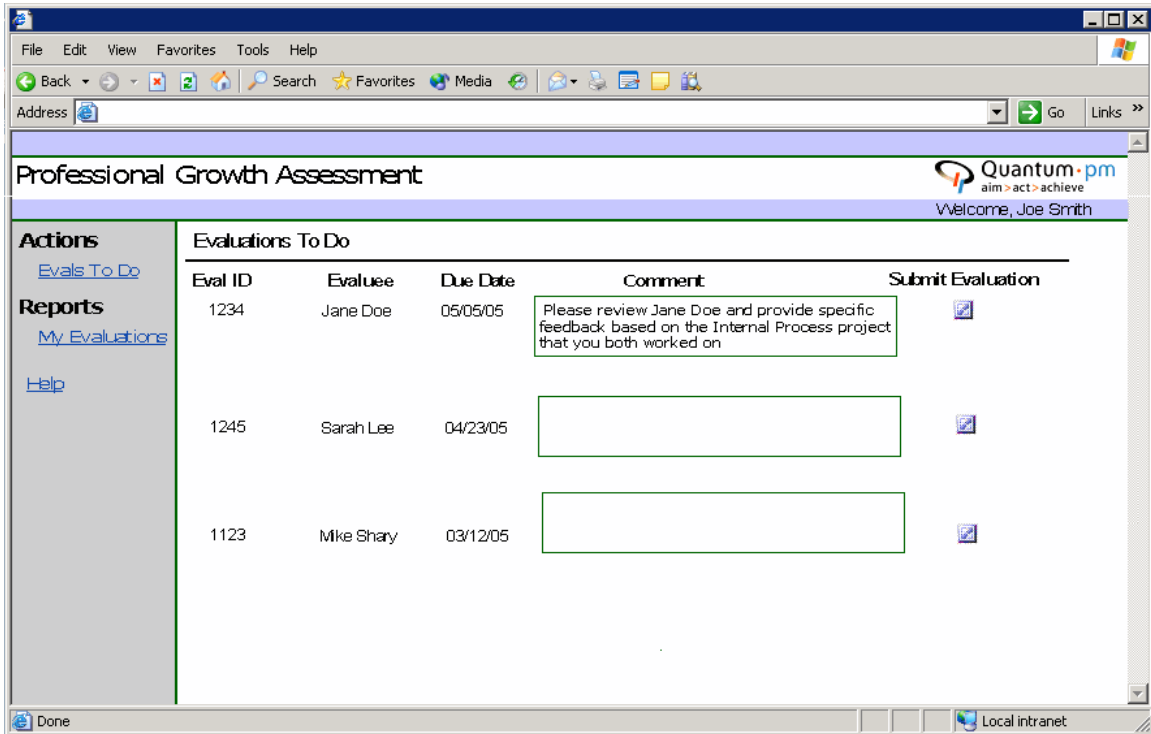
### 5.5.2 Regular User UI Screens

This section contains mockups of the UI screens that a regular user will have access to in the system.



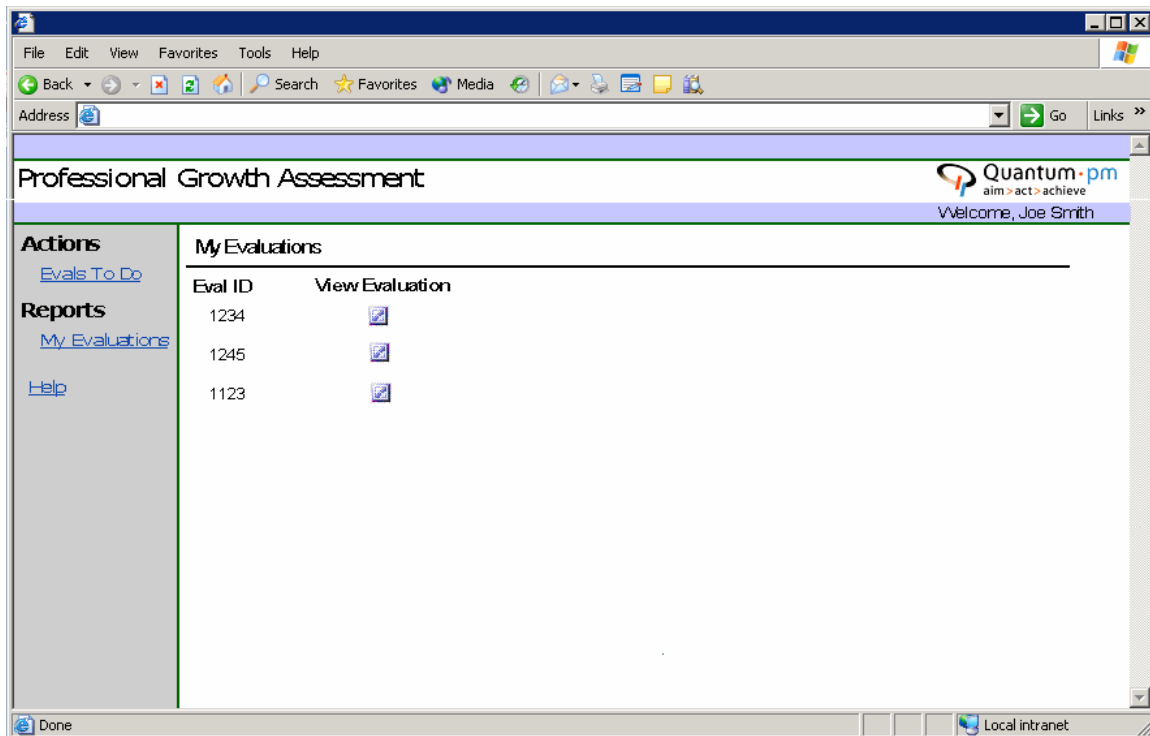
**Figure 10 PGA Regular User Main Screen**

Figure 10 demonstrates a mockup of the regular user main screen. The main screen is displayed when the user first logs into the system and displays the number of new requests to submit evaluations the user has. Clicking the hyperlink of the message, the user is taken to the Evaluations To Do screen.



**Figure 11 PGA Regular User Evaluations To Do Screen**

Figure 11 displays a mockup of the Evaluations To Do screen for regular users. The Evaluations To Do screen can be accessed by selecting Evals To Do link in the left side pane. The Evaluations To Do screen displays the Evaluation ID, Evaluee, Due Date, and ability to View Request or Submit Evaluation. When the user clicks the View Request icon, a read-only request InfoPath form opens, allowing the user to view the details of the request. When the user clicks the Submit Evaluation icon, an InfoPath form opens for that particular evaluation. Since the user has ability to either Save Changes or Submit the evaluation, when the changes are saved, the form remains on the Evaluations To Do page. Once the user submits the evaluation, that evaluation record is permanently removed from the Evaluations To Do page. The Evaluations To Do page is filtered by the Due Date, with the oldest at the top.



**Figure 12 PGA Regular User My Evaluations Screen**

Figure 12 shows a mockup screen for the My Evaluations report for regular users. The My Evaluations page can be accessed by selecting My Evaluations in the left side pane. The My Evaluations page displays the Evaluation ID and provides ability to view the evaluation. Once the user clicks the View Evaluation icon, a read-only ACI-cleansed InfoPath evaluation form is displayed. The My Evaluations page displays the evaluations in the order they were received by the Evaluee and hides the Evaluator, ACI Notes, Due Date, and Submission Date fields.

### 5.5.3 PGA Evaluation Form

Figure 13 shows an example of an Evaluation Form, developed using InfoPath. A similar (more complex) form will be developed for the final system and will integrate with the ASP .NET user interface to display the form.

## PGA Evaluation Form

### Applicant Information

Evaluee Name: 
 Due Date:

---

### Ratings Instructions

On a scale of 1-5 (1=poor, 2 = fair, 3 = average, 4 = good, 5 = outstanding, and n/a=not applicable), rate the applicant on the following skills:

### Problem-Solving and Decision Making Skills

|                                 | 1                     | 2                     | 3                     | 4                     | 5                     | n/a                              |
|---------------------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|----------------------------------|
| Technical Skills and Knowledge  | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> |
| Explanation of Rating:          |                       |                       |                       |                       |                       |                                  |
|                                 |                       |                       |                       |                       |                       |                                  |
| Consulting Skills and Knowledge | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> |
| Explanation of Rating:          |                       |                       |                       |                       |                       |                                  |
|                                 |                       |                       |                       |                       |                       |                                  |
| Additional Comments:            |                       |                       |                       |                       |                       |                                  |
|                                 |                       |                       |                       |                       |                       |                                  |

**Figure 13 Evaluation Form Example (InfoPath)**

#### 5.5.4 PGA Evaluation Request

Figure 14 shows a mockup of the evaluation request submitted by the ACI to the Evaluator.

# Evaluation Request



**Iteration Number:**

**Date:**

**Due Date:**

**Evaluee:**

**Evaluators:**

Add Evaluator

**Evaluation Comments:**

**Figure 14 PGA New Evaluation Request InfoPath Form**

The user is able to click the arrow next to the Evaluators box to insert additional row and add more Evaluators to the request. In the database, each request creates as many records as there are Evaluators to ensure that each request is associated with only one evaluation.

---

## 6.0 Phase 1.0 Implementation Details

The Implementation Details section provides additional, more detailed information about specific design decisions, implementation issues that arose, and design details that were not included in the 5.0 Phase 1.0 Design section of this document. This section is organized to provide an implementation overview of the entire system, specific implementation components, communication between the different components.

---

### Cross-Reference

Please see the Programmer's Manual for more technical details pertaining specifically to the code implemented for the PGA system in Phase 1.

---

## 6.1 PGA System Implementation Overview

As part of the requirement for the PGA system, QuantumPM has requested that the following technologies are used in the system implementation:

- Active Server Pages (ASP) .NET – used for web site creation and allows implementation of other .NET languages, such as Visual Basic .NET.
- Visual Basic (VB) .NET – used to create the forms and layouts displayed using the ASP .NET pages.
- Active X Data Objects (ADO) .NET – is an object model used to access data in the database. ADO .NET is a layer between the .NET programming languages and the database and it allows for easy access to data stored in the database. ADO .NET is used to populate the VB .NET lists with data queried from the database.
- Microsoft InfoPath – InfoPath is a Microsoft application that is part of the Microsoft Office 2003 package. InfoPath is XML based and allows for easy information collection and management. InfoPath provides many useful features such as user controls, script editing, and integration with other technologies, such as Windows SharePoint Services, Web Services, other XML documents, and Access or MS SQL Server databases. InfoPath is used in PGA to create the original Evaluation Request and Requested Evaluation forms.
- MS SQL Server 2003 Database – used to store the data collected by the PGA System.

## 6.2 ASP .NET Implementation

ASP.Net is a scripting language that allows the creation of dynamic websites with the use of Visual Studio .Net. ASP.Net allows incorporation of .NET languages, such as Visual Basic .Net and C# into web pages and web applications.

ASP.Net was chosen for this application because there are many powerful, easy, and useful tools in the .NET framework that would allow adding functionality to the application in a fast and secure way. Also, it was chosen because ASP .NET and the .NET framework are highly used and common right now. This is important because it is a lot easier to find a lot of information on the web, which speeds up the learning process and allows more time for design and coding of the application. Another



important reason is that QPM already uses the .NET framework and it would allow for the PGA system software to be very maintainable. The maintainability factor might have been the deciding one, as QPM is very pleased that we are using languages their current Development Shop specializes in and will be able to easily understand, support, and expand.

## 6.3 VB .NET Implementation

Visual Basic .NET (VB.NET) is a programming language used to create Windows applications and was chosen because QuantumPM has a lot of developers who know and use VB.NET and can build on the PGA system, when it needs to be updated. However, VB.NET was not a requirement, it was just a suggestion.

## 6.4 Website Implementation

The Website Implementation section describes the system flow and technical details that allow to system to perform the required functions.

### 6.4.1 The Login Page

The application consists of two main branches, the Anonymous Collector of Information (ACI) side and the Regular Users side. A Login page directs every user to the side they need to go to, based on Windows NT authentication. Since all users must log into QuantumPM's network, it will be easy to identify the user and direct them to their respective site. When the user logs into the network, an environment variable is set to hold their user name. The Login page will use that variable to identify the user as an employee or the ACI and redirect them to their respective sites.

### 6.4.2 The Home Page

When the users open their Home page, they will be greeted by the application using their name. The Home page has three frames:

- Title frame – the name of the application, a welcome message to the user, and the company logo.
- Side menu – contains links to the different options of the application.
- Main frame – contains the main content pages for all of the system functions.

All three frames were developed using ASP.NET with VB.NET.

The Title frame is mostly HTML, but does include a small amount of VB.NET to query the database for the username and authentication. The Side menu is HTML with a small amount of Jscript to pass information between frames. The Main frame is mostly implemented with ASP .NET and VB .NET because all of the functionality will be contained in those pages.

### 6.4.3 ACI Functions

In the ACI Side Menu, the ACI will have a Requests link that updates and shows the Requests page in the Main frame. The Requests page provides an option to fill out an Evaluation Request form (ask an employee to evaluate another employee). This option will be a link or a button that will launch InfoPath, which contains the form. The InfoPath form will use web services and SQL stored procedures to insert and get data from the database. The requests page will also show a list of recent evaluation

requests submitted and identify any outstanding evaluation requests for which the evaluations have not been submitted.

In addition, the ACI will have the option to view submitted evaluations by clicking on the Submitted Evaluations link. When the ACI navigates to this page, he will see a list of submitted evaluations that need to be edited for submission to the Evaluatee. This list is populated from the database using VB.NET and ADO.NET.

When the ACI selects an evaluation to review, the application will launch InfoPath. The ACI will be able to use InfoPath to delete or change any unacceptable feedback to constructive feedback. InfoPath will then use web services to submit and access data in the database.

Finally, The ACI will have the option to see past evaluations through the Modified Evaluations page. This will use the same logic that the Submitted evaluations page uses, except that when the ACI selects an evaluation, he/she is redirected to another ASP.NET page that shows the answers for the selected evaluation independent of InfoPath. This page is developed using VB.NET and ADO.NET. When the ACI selects evaluations, the Evaluations number will be saved in the session state, and a query using this number will be performed. The query will extract all the data for that evaluation, such as Evaluator, Evaluatee, date of submission, questions, and answers. A decision has been made to not use InfoPath in this case because there will be no data submitted into the database. Therefore all the functionality of InfoPath is not needed.

#### **6.4.4 The User Functions**

The regular system user has two options:

- The option to fill requested evaluations, which is accessed through the Evals To Do page.
- The option to see the evaluations that have been submitted for him/her, which is accessed using the My Evaluations page.

The system structure and the database have been designed so that the system is easily expandable in the future and allows for an addition of other evaluation types, such as an Ad-Hoc evaluation which can be performed at any time and is not associated with an Evaluation Request.

When the Evals To Do page is accessed, the application queries the database for all of the uncompleted evaluations by that user and displays a list of all of them. Upon clicking on the evaluations, InfoPath is launched so that the user can fill out and submit the evaluation. The user will not be able to edit or see any of the evaluations, once they have been submitted to the ACI. The evaluation will also allow the user to perform two actions:

- Save Changes – saves the user changes to the database and allows the user to come back and complete the evaluation at a later time.
- Submit – submits the entire form to the database, which requires all of the required fields (all questions) to be completed and does not allow further modifications. Once the evaluation has been submitted, it is removed from the list and the user can no longer see or access it.

The My Evaluations link takes the user to an ASP .NET page that queries the database for all of the evaluations that have been submitted for them. Upon selecting an evaluation, an ASP.NET page will appear containing all of the questions

with their respective answers, but not the Evaluators name or any information that may be useful to identify the Evaluator.

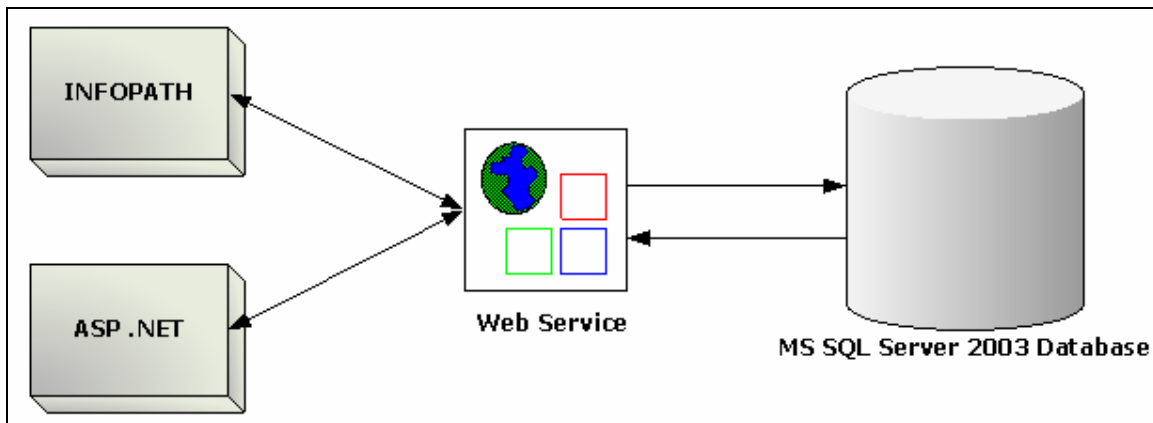
## 6.5 InfoPath Implementation

InfoPath application was one of the original guidelines and suggestions provided by QuantumPM to allow for easy development of the form interface to create a slick and consistent appearance.

InfoPath has created several risks since the beginning of the project. InfoPath is a relatively new technology and the knowledgebase for it is not quite as large as some other applications and languages. In addition, the original assumptions of InfoPath capability, such as being able to create a full user interface, were proven to be wrong with more research. As a solution, an ASP .NET and VB .NET based interface was developed for the system in order to accommodate the web-based user interface. One of the constraints of InfoPath is also that it is a desktop application and requires all users using the system to have it installed on their local machines. In order to accommodate this requirement, the PGA system will be accessed by all users from within the QuantumPM network that already provides access to the InfoPath application.

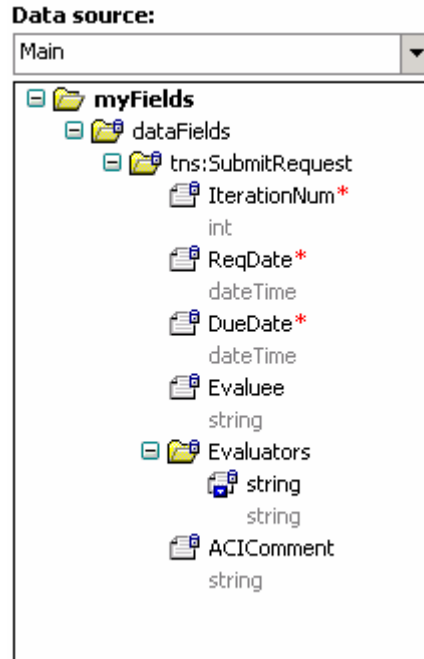
### 6.5.1 InfoPath Operation

Figure 15 below depicts the communication flow between InfoPath, Web Service, and MS SQL Server 2003 database.



**Figure 15 InfoPath Web Service Communication**

InfoPath provides the developer with ability to specify the fields used by the application, which can be then accessed and referenced by other methods accessing the InfoPath form, such as web services. The field hierarchy defines the dependencies, characteristics, data types, and other properties each field posses. Figure 16 demonstrates the field hierarchy used for the Evaluation Request InfoPath form.



**Figure 16 Evaluation Request InfoPath Data Fields**

Another constraint that was discovered during the implementation was that when creating an InfoPath connected to a web service, it has been determined that creating a web service first and then creating a form based on the parameters required by the web service is a more appropriate and error-free approach. The reason for that is a web service accepts the specified variables of specified types, but some of the types provided by InfoPath, such as Text (string), Integer, etc. that are available through drop down boxes may not be 100% compatible to data types used by the web service. Creating a Data Fields hierarchy based on the web service requested parameters allows for 100% match of the field values, which does not attempt to truncate the data and therefore does not create errors when the form is submitted.

InfoPath is connected to the web service (or any other data source, such as an XML file, Windows SharePoint Services library, or database) using a data connection that can either submit or receive data.

The submit data connections were used to submit the Evaluation Request data to the web service, which calls stored procedure to insert or check the data against the database entries.

The retrieve data connections are used to pull in required data directly from a database or a web service to provide access to the information already available in the database, such as the list of all QPM employees used to select the Evaluee and Evaluators for the Evaluation Request. This option is used to ensure that the user does not have to enter values that are global to the system and to prevent duplication of values in the database.

The main constraint with InfoPath discovered during the implementation was that in order to dynamically generate the questions and answers for the evaluation form, SQL-side XML parsing is required and additional code in Jscript needs to be written

for InfoPath in order to properly assign all objects. This implementation, including the learning curve, would have required a lot more time than the Field Session allows for, so a solution to use the already built VB .NET pages was accepted as an alternative.

As a result, Phase 1 provides a partial use of InfoPath only for the original Evaluation Request form, but makes it easier to continue to integrate InfoPath into the system design in the future iterations and system expansion.

## 6.6 Web Services Implementation

Web services were implemented in order to accommodate InfoPath access to the SQL Server 2003 database. For the Evaluation Request InfoPath form, Submit Request web service was implemented. The Submit Request web service can be accessed on the local machine by its Web Service Description Language (WSDL) at <http://localhost/InfoPathWebService/SubmitRequest.asmx?WSDL>.

The Submit Request web service has one main function SubmitRequest that performs the following:

- Takes in the parameters obtained from the InfoPath form that include:
  - IterationNum – (integer) indicates the iteration number to which the request belongs to. This is used to track which requests appear on the ACI Requests screen.
  - ReqDate – (date) indicates the date of the request (within InfoPath automatically defaults to today's date).
  - DueDate – (date) indicates the date by which the ACI needs the Evaluators to submit evaluations associated with this Evaluation Request.
  - Evaluee – (string) indicates the Evaluee name.
  - Evaluators – (array of strings) indicates all of the Evaluators' names.
  - ACIComment – (string) indicates the evaluation comment that ACI has provided as additional instructions in filling out the evaluation.
- Creates a connection to the SQL server
- Calls the P\_CheckEvaluee stored procedure that retrieves the Evaluee employee ID from the database, based on the Evaluee name passed in by the InfoPath form.
- For each Evaluator, it iteratively calls:
  - The P\_CheckEvaluator stored procedure that retrieves the Evaluator employee ID from the database, based on the Evaluator name passed in by the InfoPath form.
  - The P\_InsertRequest stored procedure that inserts a new row into the T\_Evaluations table with all of the request information passed in by the InfoPath form.

## 6.8 PGA Encryption Implementation

For encryption, an Encrypt() function is going to be used. Originally, this function was developed to encrypt passwords and it was found at <http://www.planet-source-code.com/vb/scripts/ShowCode.asp?txtCodeId=847&lngWId=5>. This function was chosen to be used for its ease of implementation as it does not require writing of any additional classes using the .NET libraries, which can get relatively complex. The Encrypt() function takes a string input and creates a key value depending on the length of the string. The function then cycles through each character in the string, converting it to unreadable, encrypted text. This action is performed in a while loop, so that it iterates through all characters in the string, hence the substring of answer at position iPOS for 1 character. The Encrypt() function is easy to use and it scrambles the letters in the database, so anyone looking at the tables directly in the database will be unable to read the answers.

One concern with this function is that it may not be as secure as originally intended as it does not make use of the public and private keys. The encryption is performed on the strings coming in and going out of the database and is not based on the user credentials. This is a serious concern, but after further considerations and discussions with the client, a decision was made to use the Encrypt() function.

In addition, to increase security even more, the function itself is encrypted and can be unlocked only by a password that will be available to the system administrator and the developers that will perform further work on the system.

---

### Cross-Reference

Please see the Programmer's Manual for details on the method used by Encrypt() function to encrypt the data.

---

## 6.9 Future Recommendations and Lessons Learned

The Future Recommendations and Lessons Learned section provides a high-level overview of the recommended future system enhancement. It also includes the valuable lessons that were learned during the Phase 1 implementation.

### 6.9.1 Lessons Learned

As part of the software development process of the PGA system, the team has performed lessons learned to discover what parts of the process and system implementation worked and which did not.

The lessons learned during the Phase 1 implementation are:

- Additional time needs to be allotted to ensure that the learning curve does not threaten the success of the project.
- Faster agreement upon the system requirements for the selected phase and a better knowledge of the technologies to be used is necessary to avoid some of the lost time in redesigning and re-thinking the system when constraints arise.
- Team communication has helped tremendously to ensure that all team members are on the same page and participating in the important decisions for the final product.

- Efficient and wise division of tasks is important in order to provide for a successful completion of the project. It is important to divide work based on the skills of the team members.
- Creating two solutions at once sometimes is the best approach when it is uncertain which technology will work at the end. The simultaneous development of the forms filled out by system users was done in both InfoPath and VB .NET. As a result, both solutions were used and created a nice compromise to the system's original requirements.
- Careful and detailed documentation is important and provides the necessary information not only to the current team members, but also to any developers that may be expanding the system in the future.
- Thorough research of all available solutions, smarter alternatives were discovered that reduced the implementation time.

As an overall lessons learned, the team has had a great experience with the system as it provided for the true real-world exposure, where things do not always work the way they are planned and sometimes it takes several tries of doing things the wrong way in order to get something right.

## 6.9.2 Future Recommendations

Phase 2 requirements have been included as part of this report in order to provide a future direction for the system enhancements. In addition to the functionality listed in the Phase 2 requirements, it is recommended that several technological adjustments should be made to further extend the system. These adjustments include:

- Create further system integration with one chosen technology. It is possible that in order to simplify the future system maintenance, it will be important to use fewer technologies and to switch exclusively to either using VB .NET or InfoPath. In this case, it is also important to take into consideration the constraints associated with each of these solutions, and in the end, it is possible that a full development of both is the ultimate choice. PGA system will have a lot to offer if the user had a choice of how to view the data and ability to do both.
- Dynamic generation of InfoPath forms must be considered to increase the system efficiency. Currently, most of the InfoPath code is static and many fields and parameters are hard-coded as a solution to the tight time frame. If the InfoPath forms are generated dynamically, it will decrease the maintenance time as well as allow for faster generation of data.
- Additional functions as described in Phase 2 requirements of this report should be considered for the further development of the system.
- A user interface for additional system maintenance, such as purging of records, can be implemented to require less time from programmers to maintain and update the system. In addition, ability to add new questions to the forms using the user interface will also provide more flexibility to the system.
- Converting the remainder of the VB .NET and ASP .NET code into classes is recommended in order to simplify the code maintenance and increase readability.

---

## Questions and Concerns

| Date | Section Affected | Initials | Comment |
|------|------------------|----------|---------|
|      |                  |          |         |
|      |                  |          |         |
|      |                  |          |         |
|      |                  |          |         |
|      |                  |          |         |



---

## Appendix A – Evaluation Questions

This section contains the detailed questions that must be included in the evaluation, divided by their respective areas of evaluation.

The following instructions need to be displayed in the beginning of each survey:

### ***Evaluation Instructions***

You are being asked to rate your team peers on a number of management and leadership skills and attributes. Your observations will be a key component of a team development program for QPM. Please take time to make a careful, considered assessment. Be assured that your rating is totally confidential. *Remember, your ratings are anonymous and will simply be averaged with all other ratings received. None of your responses will be attributed to you as an individual. Your comments will be mixed with the comments of other raters, and all identifying information will be eliminated.*

Comments are the most important part of the survey. Please take time to explain your rating; particularly if you have rated the individual extremely critically or extremely favorably.

### **1. PROBLEM-SOLVING AND DECISION MAKING SKILLS**

**1a1.** Technical Skills and Knowledge – specific proficiency of expertise acquired through formal education, training, or experience; committed to own self-development.

POOR FAIR AVG GOOD OUTSTANDING (unable to rate)

Explanation of rating:

**1a 2.** Consulting Skills and Knowledge – the extent to which the individual accurately reads and responds astutely and diplomatically to clients organizational needs and effectively deals with organizational politics.

POOR FAIR AVG GOOD OUTSTANDING (unable to rate)

Explanation of rating:

**1a 3.** Training Skills and Knowledge – to what extent does the individual keep up-to-date in their discipline and are able to incorporate newly acquired skills in the work place as well as share knowledge with other in the company.

POOR FAIR AVG GOOD OUTSTANDING (unable to rate)

Explanation of rating:

**1a 4.** Interpersonal Ego Containment - To what degree does the individual promote the expertise of the company over individual self interest and promotion? The WE vs. Me.

POOR FAIR AVG GOOD OUTSTANDING (unable to rate)

Explanation of rating:

**1a 5.** Commitment to effective Communications – to what extent does individual use high-impact techniques for informing and communicating on a day-to-day basis. Sophistication of written and verbal communications.

POOR FAIR AVG GOOD OUTSTANDING (unable to rate)

Explanation of rating:

**1a 6.** Adaptability and Emotional Stability – when situations are volatile and high pressured is the individual poised and composed to the benefit of the team and project.

POOR FAIR AVG GOOD OUTSTANDING (unable to rate)

Explanation of rating:

**1B.** Analytical Ability – the ability to dissect and understand complex, multi-faceted problems, identify relevant information and get to the source of the problem.

POOR FAIR AVG GOOD OUTSTANDING (unable to rate)

Explanation of rating:

**1C.** Decisiveness – the ability to make clear-cut and timely decisions, with the appropriate amount of data, when they are needed, and is sensitive to the differing impacts among the various stakeholders; will take a stand, even when it's controversial.

POOR FAIR AVG GOOD OUTSTANDING (unable to rate)

Explanation of rating:

**1D.** Creativity – the ability to initiate original and innovative ideas, products, services, and approaches. Helps discover outside-the –box solutions.

POOR FAIR AVG GOOD OUTSTANDING (unable to rate)

Explanation of rating:

**1E.** Thoroughness – the ability to attend to detail and develop a comprehensive approach to problems.

POOR FAIR AVG GOOD OUTSTANDING (unable to rate)

Explanation of rating:

**1F.** Objectivity – the ability to maintain realistic and rational perspective, and keep personal bias and agenda to a minimum.

POOR FAIR AVG GOOD OUTSTANDING (unable to rate)

Explanation of rating:

**1G.** Risk Taking – the willingness to take sound, calculated risks, based on good judgment, in situations where the outcome is uncertain; comfortable challenging the status quo and operating in the midst of ambiguity.

POOR FAIR AVG GOOD OUTSTANDING (unable to rate)

Explanation of rating:

**1H.** Open-mindedness – the willingness to consider new ideas and approaches, as well as input from others; quick to recognize new opportunities and to pursue them with vigor; eager to explore issues from any number of perspectives.

POOR FAIR AVG GOOD OUTSTANDING (unable to rate)

Explanation of rating:

## 2. SOCIAL AND COMMUNICATIONS SKILLS

**2A.** First Impression – the ability to create a positive impact through physical presence, social confidence, sincerity, and fluency with verbal concepts; is immediately approachable – puts others at ease quickly.

POOR FAIR AVG GOOD OUTSTANDING (unable to rate)

Explanation of rating:

**2B.** Sociability – the ability to relate to others in an outgoing, friendly, warm and personable manner; is supportive and sincerely cares about others.

POOR FAIR AVG GOOD OUTSTANDING (unable to rate)

Explanation of rating:

**2C.** Interpersonal Skills – the ability to enter into and maintain, overtime, effective interpersonal relationships with others; trusted and sought out by others.

POOR FAIR AVG GOOD OUTSTANDING (unable to rate)

Explanation of rating:

**2D.** Empathy and consideration – the extent to which the individual is supportive, considerate, and sensitive, and truly cares about the needs, concerns, agenda, interests, and aspirations of others.

POOR FAIR AVG GOOD OUTSTANDING (unable to rate)

Explanation of rating:

**2E.** Organizational Awareness – the ability to accurately read and respond astutely and diplomatically to organizational trends and norms, and effectively deal with organizational politics.

POOR FAIR AVG GOOD OUTSTANDING (unable to rate)

Explanation of rating:

**2F.** Conflict Management – the ability to mediate and resolve conflicts and disagreements between others, and to resolve conflicts between him/herself and others, in a manner that is best for all parties involved; promotes an atmosphere that is conducive to openness and candor; seeks the win-win resolution, yet “tells it like it is”.

POOR FAIR AVG GOOD OUTSTANDING (unable to rate)

Explanation of rating:

**2G.** Multi-directional communication – keeping team and management informed about decisions, events, and developments that affect them; receptive to soliciting of feedback; always working to corroborate, expand, and clarify information to maximize accuracy, relevance, and objectivity.

POOR FAIR AVG GOOD OUTSTANDING (unable to rate)

Explanation of rating:

**2H.** Formal Presentation – the ability to deliver an interesting, informative, and organized presentation; the ability to explain complex topics in easy-to-understand terms.

POOR FAIR AVG GOOD OUTSTANDING (unable to rate)

Explanation of rating:

**2I.** Persuasiveness – the ability to influence others about ideas, approaches, innovation, and strategies; effective at changing others’ perspectives and gaining their commitment.

POOR FAIR AVG GOOD OUTSTANDING (unable to rate)

Explanation of rating:

**2J.** Negotiation – the ability to negotiate outcomes which further the interests of the organization and, when possible, also further the interests of opposing groups; trusted to find creative, mutually beneficial outcomes.

POOR FAIR AVG GOOD OUTSTANDING (unable to rate)

Explanation of rating:

**2K.** Listening skill – the ability to closely attend to the verbal communication of others, picking out the relevant information and conveying to the other person both

a receptivity to and an understanding of their intended message; skilled at surfacing underlying, unspoken issues and confronting them in a supportive way.

POOR FAIR AVG GOOD OUTSTANDING (unable to rate)

Explanation of rating:

### 3. PERSONAL CHARACTERISTICS

**3A.** Drive and Motivation – the energy and motivation to work hard, strive to be successful, attain ambitious goals and complete difficult tasks; eager to make a contribution to the greater good of the team/organization; eager to take on greater responsibility.

POOR FAIR AVG GOOD OUTSTANDING (unable to rate)

Explanation of rating:

**3B.** Self-Discipline – the ability to adapt to change and modify one's style and approach, in order to adjust to changing circumstances or to attain an objective; recognizes that change is a constant and embraces it as an opportunity.

POOR FAIR AVG GOOD OUTSTANDING (unable to rate)

Explanation of rating:

**3C.** Flexibility – the ability to adapt to change and modify one's style and approach, in order to adjust to changing circumstances or to attain an objective; recognizes that change is a constant and embraces it as an opportunity.

POOR FAIR AVG GOOD OUTSTANDING (unable to rate)

Explanation of rating:

**3D.** Independence – the ability to be self-starting and to work independently of others when necessary; demonstrates a "just do it " attitude.

POOR FAIR AVG GOOD OUTSTANDING (unable to rate)

Explanation of rating:

**3E.** Poise – a strong sense of self-worth and a quiet self-confidence; is tolerant, patient, and understanding.

POOR FAIR AVG GOOD OUTSTANDING (unable to rate)

Explanation of rating:

**3F.** Emotional Control and Balance – the ability to maintain perspective and personal composure during times of stress or pressure, when things are uncertain, or when faced with conflict or disagreement; flexible under changing or ambiguous circumstances.

POOR FAIR AVG GOOD OUTSTANDING (unable to rate)

Explanation of rating:

**3G.** Dependability – the ability to be counted on to meet commitments and deadlines; accountability for own actions; is consistent, predictable and reliable.

POOR FAIR AVG GOOD OUTSTANDING (unable to rate)

Explanation of rating:

**3H.** Integrity and Ethical Conduct – the strength of one’s values and the ability to resist compromising basic business and ethical principles in the face of pressure; forthright, no hidden agenda, and protective of confidential information.

POOR FAIR AVG GOOD OUTSTANDING (unable to rate)

Explanation of rating:

#### **4. LEADERSHIP ATTRIBUTES**

**4A.** General Leadership Effectiveness – the ability to influence and guide the behavior of others in a certain direction and to motivate, help and develop staff.

POOR FAIR AVG GOOD OUTSTANDING (unable to rate)

Explanation of rating:

**4B.** Vision – having a sense of personal purpose and company direction and providing a compelling vision to others, which they can believe in; thinks and acts from a strategic perspective, taking the long-term view.

POOR FAIR AVG GOOD OUTSTANDING (unable to rate)

Explanation of rating:

**4C.** Directiveness – letting staff know, in a clear and motivating way, what tasks should be done and what results are expected; explain reasons and purpose for a particular action.

POOR FAIR AVG GOOD OUTSTANDING (unable to rate)

Explanation of rating:

**4D.** Emphasizing Excellence In Performance – setting challenging goals and high quality standards and assisting staff to perform at their highest level.

POOR FAIR AVG GOOD OUTSTANDING (unable to rate)

Explanation of rating:

**4E.** Company Representation – serving as a figurehead and spokesperson for one's unit, and effectiveness in promoting and defending the interests of one's team members.

POOR FAIR AVG GOOD OUTSTANDING (unable to rate)

Explanation of rating:

**4F.** Fostering involvement and decision-making Participation – collaborating and consulting with staff, soliciting suggestions, and taking these suggestions seriously; creates commitment and ownership.

POOR FAIR AVG GOOD OUTSTANDING (unable to rate)

Explanation of rating:

**4G.** Facilitating Teamwork – ability to promote teamwork, cooperation and identification with the work group and across disciplines; inspires and sustains team's energy.

POOR FAIR AVG GOOD OUTSTANDING (unable to rate)

Explanation of rating:

**4H.** Inspirational Role Model – ability to set a positive and inspirational example for others to follow; viewed as a source of leadership and motivation; maintains alignment throughout the organization with a shared sense of purpose.

POOR FAIR AVG GOOD OUTSTANDING (unable to rate)

Explanation of rating:

## **5. LEADERSHIP SKILLS**

**5A.** Short-Term Planning – the ability both to establish goals and objectives with staff and the work unit and to develop the action steps to achieve them.

POOR FAIR AVG GOOD OUTSTANDING (unable to rate)

Explanation of rating:

**5B.** Strategic Planning – the ability to develop a long-range direction with a clear focus for the organization or unit.

POOR FAIR AVG GOOD OUTSTANDING (unable to rate)

Explanation of rating:

**5C.** Organizing The Work Of Others – clearly defining roles, responsibilities, and expectations for team members; creates a challenging and satisfying work environment.

POOR FAIR AVG GOOD OUTSTANDING (unable to rate)

Explanation of rating:

**5D.** Empowerment – delegating responsibility and authority to team members, and allowing others discretion in determining how to do their work; collaborative in overall approach to people and projects.

POOR FAIR AVG GOOD OUTSTANDING (unable to rate)

Explanation of rating:

**5E.** Monitoring -- tracking project performance with a focus on performance enhancement and continuous improvement; effective handling of multiple projects.

POOR FAIR AVG GOOD OUTSTANDING (unable to rate)

Explanation of rating:

**5F.** Motivating Others -- showing enthusiasm and providing encouragement, recognition, performance feedback, and coaching to others; is an advocate and catalyst for change.

POOR FAIR AVG GOOD OUTSTANDING (unable to rate)

Explanation of rating:

**5G.** Staffing -- a sound record for finding, selecting, hiring, developing, and retaining highly qualified staff.

POOR FAIR AVG GOOD OUTSTANDING (unable to rate)

Explanation of rating:

**5H.** Networking -- using informal channels and networks for information sharing, support, resource acquisition, and the effective execution of projects.



POOR FAIR AVG GOOD OUTSTANDING (unable to rate)

Explanation of rating:

**51.** Customer/Client Focus -- strongly committed to building strong relationships with customers/clients and to deliver satisfaction at the highest possible level; driven be a core value of customer satisfaction, as that relates both to internal and external stakeholders.

POOR FAIR AVG GOOD OUTSTANDING (unable to rate)

Explanation of rating:

---

## Appendix B - Definitions, Abbreviations, and Acronyms

### ***PGA***

PGA – Professional Growth Assessment System. The PGA System allows users to submit anonymous constructive feedback about other company employees for purpose of personal and professional growth and improvement.

### ***ACI***

ACI – Anonymous Collector of Information. ACI is a designated company employee that is responsible for filtering the evaluation feedback to allow for maximum constructive comments and to facilitate the evaluation process. ACI is the only person in the company that has privileges to view all evaluations.

### ***InfoPath***

InfoPath is a Microsoft application that is included in Microsoft Office 2003 and is used for information gathering and management. InfoPath is XML based and allows integration with many other technologies, such as Windows SharePoint Services, .NET, and so on. InfoPath provides an easy to use user interface and many pre-built user controls, such as date pickers, drop down lists, repeating tables, and so on. More information on InfoPath can be found at the Microsoft InfoPath website at <http://office.microsoft.com/en-us/FX010857921033.aspx>.

### ***Active Server Pages (ASP) .NET***

ASP .NET is the latest version of Microsoft's Active Server Pages technology (ASP). ASP is a server side scripting technology that enables scripts (embedded in web pages) to be executed by an Internet server. ASP is a Microsoft technology and is a program that runs inside Internet Information Services (IIS). ASP .NET is an entirely new paradigm for server-side ASP scripting. More information on ASP .NET can be found at Microsoft Office ASP .NET website at <http://www.asp.net/Default.aspx?tabindex=0&tabid=1>.

### ***Visual Basic (VB) .NET***

VB .NET is a version of Microsoft's Visual Basic language that was designed, as part of the company's .NET product group, to make Web services applications easier to develop. Visual Basic is a programming environment from Microsoft in which programmer uses a graphical user interface to choose and modify pre-selected sections of code written in the Basic programming language. VB .NET is first fully object-oriented programming (OOP) version of Visual Basic, and supports OOP concepts such as abstraction, inheritance, polymorphism, and aggregation. More information and resources on VB .NET can be found on Microsoft VB .NET website at <http://msdn.microsoft.com/vbasic/>.

### ***Web Services***

Web Services is a standardized way of integrating Web-based applications using XML, SOAP, WSDL, and UDDI open standards over the internet protocol backbone. XML is used to tag (a command inserted in a document that specifies how the document, or a portion of the document, should be formatted) the data, SOAP is used to transfer the data, WSDL is used for describing the services available and UDDI is used for listing what services are available. Web services are used primarily for businesses to communicate with each other and with clients.

### ***Extensible Markup Language (XML)***

XML is a standard for creating markup languages which describe the structure of data. XML is not a fixed set of elements like HTML, but rather, it is like SGML (Standard Generalized Markup Language) in that it is a metalanguage, or a language for describing languages. XML allows designers to create their own customized tags, enabling the definition, transition, validation, and interpretation of data between applications and between organizations.

### ***Simple Object Access Protocol (SOAP)***

SOAP is the standard for web services messages. SOAP is based on XML and defines an envelope format and various rules for describing its contents. Seen (with WSDL and UDDI) as one of the three foundation standards of web services, it is the preferred protocol for exchanging web services.

### ***Web Service Description Language (WSDL)***

WSDL is an XML-formatted language that is used to describe a Web service's capabilities as collections of communication endpoints capable of exchanging messages. To invoke a service, a consuming application must know the service's interface, including how to structure content and which transport protocol to use. WSDL explicitly describes this interface in a standardized, machine-readable format fit for consumption by tools. WSDL is an integral part of UDDI, an XML-based worldwide business registry.

### ***Universal Description Discovery Integration (UDDI)***

UDDI is an XML-based protocol that provides a distributed directory that enables businesses to list themselves on the Internet and discover other services. Similar to a telephone number, businesses can list themselves by name, product, location, or the web service they offer.