Spring 2008

The Automation of Software Development Metrics

Anthony J. Traino
Regis University

Regis University

College for Professional Studies Graduate Programs

Final Project/Thesis

The Automation of Software Development Metrics

Anthony J. Traino

Regis University

School for Professional Studies

Master's of Science in Computer Information Technology

Front Matter

Regis University

School for Professional Studies Graduate Programs

MSc in Computer Information Technology Program

Graduate Programs Final Project/Thesis

Certification of Authorship of Professional Project Work

Print Student's Name Anthony Joseph Traino

Telephone (585) 739-7037

Email atraino@gmail.com

Date of Submission 4/23/2008 Degree Program **MSc in Computer Information Technology**

Title of Submission The Automation of Software Development Metrics

Advisor/Faculty Name Greg Wells

Certification of Authorship:

I hereby certify that I am the author of this document and that any assistance I received in its preparation is fully acknowledged and disclosed in the document. I have also cited all sources from which I obtained data, ideas or words that are copied directly or paraphrased in the document. Sources are properly credited according to accepted standards for professional publications. I also certify that this paper was prepared by me for the purpose of partial fulfillment of requirements for the Master of Science in Computer Information Technology Degree Program.

Anthony Joseph Traino 4/23/2008

Student Signature *Date*

School for Professional Studies Graduate Programs

MSc in Computer Information Technology Program

Graduate Programs Final Project/Thesis
Certification of Authorship of Professional Project Work

Print Student's Name _____

Telephone _____

Email _____

Date of Submission _____ Degree Program **MSc in Computer Information Technology**

Title of Submission _____

Advisor/Faculty Name _____

Certification of Authorship:

I hereby certify that I am the author of this document and that any assistance I received in its preparation is fully acknowledged and disclosed in the document. I have also cited all sources from which I obtained data, ideas or words that are copied directly or paraphrased in the document. Sources are properly credited according to accepted standards for professional publications. I also certify that this paper was prepared by me for the purpose of partial fulfillment of requirements for the Master of Science in Computer Information Technology Degree Program.

_____ _____

Student Signature *Date*

Regis University

School for Professional Studies Graduate Programs

MSc in Computer Information Technology Program

Graduate Programs Final Project/Thesis
Authorization to Publish Student Work

I, Anthony Joseph Traino, the undersigned student, in the Master of Science in Computer Information Technology Degree Program hereby authorize Regis University to publish through a Regis University owned and maintained web server, the document described below ("Work"). I acknowledge and understand that the Work will be freely available to all users of the World Wide Web under the condition that it can only be used for legitimate, non-commercial academic research and study. I understand that this restriction on use will be contained in a header note on the Regis University web site but will not be otherwise policed or enforced. I understand and acknowledge that under the Family Educational Rights and Privacy Act I have no obligation to release the Work to any party for any purpose. I am authorizing the release of the Work as a voluntary act without any coercion or restraint. On behalf of myself, my heirs, personal representatives and beneficiaries, I do hereby release Regis University, its officers, employees and agents from any claims, causes, causes of action, law suits, claims for injury, defamation, or other damage to me or my family arising out of or resulting from good faith compliance with the provisions of this authorization. This authorization shall be valid and in force until rescinded in writing.

Print Title of Document(s) to be published: The Automation of Software Development Metrics

Anthony Joseph Traino — *Student Signature*

4/23/2008 — *Date*

Check appropriate statement:

X The Work does not contain private or proprietary information.

_____ The Work contains private or proprietary information of the following parties and their attached permission is required as well: _____

Name of Organization and/or Authorized Personnel

Regis University

School for Professional Studies Graduate Programs

MSc in Computer Information Technology Program

Graduate Programs Final Project/Thesis
Advisor/Professional Project Faculty Approval Form

Student's Name: Anthony Traino Program: MS in Computer Information Technology
PLEASE PRINT

Professional Project Title: The Automation of Software Development Metrics
PLEASE PRINT

Content Advisor Name Gregory S. Wells
PLEASE PRINT

Project Faculty Name _____
PLEASE PRINT

Advisor/Faculty Declaration:

I have advised this student through the Professional Project Process and approve of the final document as acceptable to be submitted as fulfillment of partial completion of requirements for the MSc in Computer Information Technology Degree Program.

Gregory S. Wells — 4/22/2008
Original Content Advisor Signature — *Date*

Donald Kline — 4-30-2008
Original Module/Class Facilitator Signature — *Date*

Degree Chair Approval if:

The student has received project approval from Faculty and has followed due process in the completion of the project and subsequent documentation.

Sharri A. Hartz-Masters — 5-1-2008
Original MScCIT Degree Chair/Designee Signature — Date

Project Paper Revision History

Version	Date	Notes
First Draft	1/15/2008	Submitted to Advisor – Greg Wells
Second Draft	3/19/2008	Submitted to Advisor – Greg Wells
Final Draft	3/31/2008	Approved by Advisor – Greg Wells
Final Draft	4/12/2008	Submitted to Advisor – Greg

Table of Contents

Figures Table

Index of Tables

Abstract

This project examines an organization that lacks adequate metrics for the evaluation of quality software maintenance of a product. A database and software solution will be implemented in order to organize data and report significant information to management. Prior to the implementation of this database, inaccurate statistics have been reported and no relevant metrics have been established. The statistics that have been reported are superficial and unfiltered. The database and software solution to be implemented will be populated by an automated feed from three separate databases. Each of the databases contains information that is relevant to the metrics that will be defined. Once the database is populated, management will be able to run a report, called the Payroll Report Card.

The establishment of this system will define the company's metrics and will measure the true quality of the organization's software changes. This will serve as a case study for software development organizations who seek to define metrics and improve quality with a limited budget.

Acknowledgements

Working on this project, and the Masters' degree as a whole, has been a long journey. Finding a good project to work on was a challenge, and I would like to thank all the co-workers and my boss for allowing me the opportunity to lead such a wonderful initiative. I would not have been able to complete this project without all of your dedication and hard work.

Because of the long nights and heavy workloads, my family also felt the impact. I would like to thank my parents for the encouragement and patience as I finished this project/degree. You always emphasized the importance of education and you were right.

Finally, I would like to thank the love of my life, Heather. Thank you for your continued support and encouragement, even when I did not want to go any further. There is very little I could do without you.

Executive Summary

A national payroll provider has experienced large growth in its Information Technology department. Many new projects have been scheduled and completed, due to an expanding product line. This has caused the quality of software projects to decrease in quality. The company has no process in place for evaluating metrics and improving deliverables.

This paper outlines the company's challenges, and proposes an automated metrics system that provides a quarterly report. The initiative faces many challenges due to restrictive company security policies and no budget. The only resources that are available for the project include five employees (developers and QAs) and any licensed applications that are used as part of the job.

The project requires that an analysis is conducted to evaluate the company's goals, locate key decision-makers and then define metrics based on available data. The data must then be made available by centralizing the information so that a report can be created in Microsoft Excel. Centralizing the data includes creating export programs, import programs and parsing programs where necessary. Within the initiative, it is required that all resources and solutions abide by company security policies. Furthermore, it is essential that key decision-makers agree on the established metrics and be prepared to make changes based on the findings.

The schedule for the project follows a 10 week plan where requirements are derived, technical solutions are identified, deliverables are made and then the report is implemented. Throughout the timeline, the key metrics are established and included in the technical solution. This leads to the project's completion.

The project concludes as successful. The automated metrics report is proposed and implemented as defined in the early stages of the project. The reception and action of management is not documented in this paper, but recommendations have been made to enhance the report. The enhancements include refining metrics as reports are reviewed and increasing the report's accessibility across multiple departments.

Chapter 1: Introduction

Introduction

As computer technology continues to grow across the globe, the software development industry has become more complex to manage and measure. Software development has become one of the fastest growing occupations in the United States (U.S. Department of Labor, 2004). Software projects are often completed through outsourcing efforts or by in-house development teams. If a project is not run smoothly, costs for such efforts can quickly rise. Furthermore, the end product may not fit the requirements or the intentions of the sponsor. This can result in incomplete projects that require re-work or future efforts to correct major issues. To control the quality and the cost of a project, every organization should have a working process for collecting and analyzing adequate metrics.

An organization can drastically improve its ability to measure success of software development projects through metrics identification and reporting. Metrics are a collection of data, from a process or system, which are collected and compared in a measurement scale with organizational goals. Implementing a reporting database with measurable data can provide the ability to consistently gauge project success. It provides a baseline for measurement and allows an organization to identify where improvements have been made, and where they can be made (Wiegers, 1999). The most productive way of achieving this is through identifying adequate organizational metrics, simplifying the reporting mechanism through automation and tying the data to process improvement committees (Wiegers, 1999).

Problem Definition

The company that is used in this case study will be referred to as *Superior Payroll Solutions*. Superior Payroll Solutions is a national payroll service that provides payroll and human resource services to its clients. The company has three major software products that allow clients to input their payroll information. The product that this case study will focus on will be referred to as *Superior Payroll*. Superior Payroll has rapidly grown throughout the past three years. It was originally intended to be retired, being replaced by more graphically driven software. When the organization found that the behind-the-scenes calculations were difficult to replicate, the product's lifespan was extended. Because of Superior Payroll's marketable features, the company began to schedule more software development projects. Throughout the course of three years, the software development team expanded by over 300%. With this expansion, quality became much more difficult to maintain.

In order to improve quality, the product's development team will develop an automated tool that gathers data and presents pre-defined measurements/metrics. It will import this data into a formatted document that will serve as a quarterly metrics report for management review. This tool will not be completely automated, as the Development Manager will have to make some slight modifications on a quarterly basis. However, the format, graphs and data must be readily available whenever needed. The measurements will be identified through extracting valuable measurements, evaluating the data and executing reports to meet the goals of the product (Ebert, Dumke, Bundschuh, and Schmietendorf, 2005). Reviewing organizational goals will help identify these statistics so that success criteria can be in line with development metrics (Syntelligence, 2005).

Relevance, Significance or Need for the Project

Superior Payroll Solutions is quickly expanding its use of the Superior Payroll product across its customer base. Because of this growth, product-related defects and system limitations are becoming a priority for the business unit. An increase in software projects has caused the team to expand its resources for both development and quality assurance specialists. The goal of the team is to increase project bandwidth while maintaining a strong level of quality. This, however, can only be done through collecting adequate metrics on a quarterly basis to see where quality efforts are failing.

Superior Payroll supports a wide range of features for payroll processing and human resource services. Based on priority and impact, metrics must be selected to rate the overall health of the product in production. Furthermore, metrics must be collected that can evaluate quality efforts (quality assurance) as well as work/project prioritization effectiveness (operations work requests). This means that the metrics reporting tool will focus on various areas of the software development life cycle.

In order to gather the required data, access to information in multiple databases is going to be required. A data import/export will be required, which will consolidate the information into one database. This will be referred to as the *central database*. The quarterly metrics report will then be built using the data stored in the central database. This report will be used by management to report product-relevant successes and failures to senior management.

Barriers and Issues

This project will require that data is exported from multiple databases, because company security policies do not allow for direct access from third-party applications.

More specifically, data exports will be needed for the change management system, the defect-tracking tool and the production application databases.

Financial limitations will disallow any major purchases and will require that more in-house development is done to fill in any technical gaps. This will most likely apply to a data parser, a data importer and a data exporter.

There is also potential for inaccessibility to specific data. This will be identified throughout the project, but will cause occasional limitations in regard to collecting ideal metrics.

Elements, Hypotheses, Theories, or Questions to be Discussed / Answered

This thesis claims that automated metrics reporting can be achieved through in-house efforts. Many organizations suffer from scattered data/statistics, and struggle to implement a metrics reporting tool that will analyze quality software successes and failures.

This thesis also claims that metrics reporting can be achieved, even when data is secured and direct database access is denied.

There are two main questions that must be answered within this thesis:

1. Can a metrics reporting tool be implemented through in-house efforts with a limited budget?
2. Can the metrics reporting tool truly be automated to contain up-to-date information without direct database access?

Scope of the Project

This project is concerned with the gathering and automation of metrics within a software development organization. The data must be identified by working with

management. The data must then be located and migrated to a central database, where the report can access relevant information. Finally, the report will be organized and displayed for management review.

Figure 1 illustrates the users interacting with the relevant applications. These applications are currently isolated from one another. Each application has a separate database structure with no relation to the others.

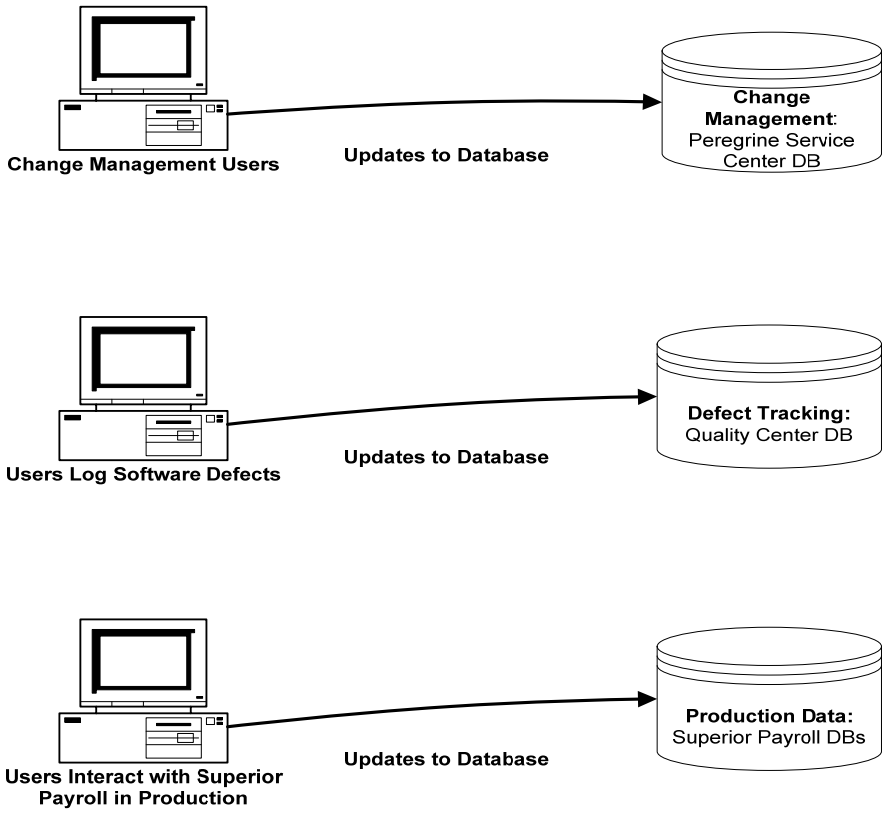

Figure 1: Product updates are made to databases

Figure 2 represents the proposed solution to gaining access to the data in the three separate databases. Because direct access to the data is not allowed, each data owner would provide a daily export of data. The data would then be parsed and imported into the central database.

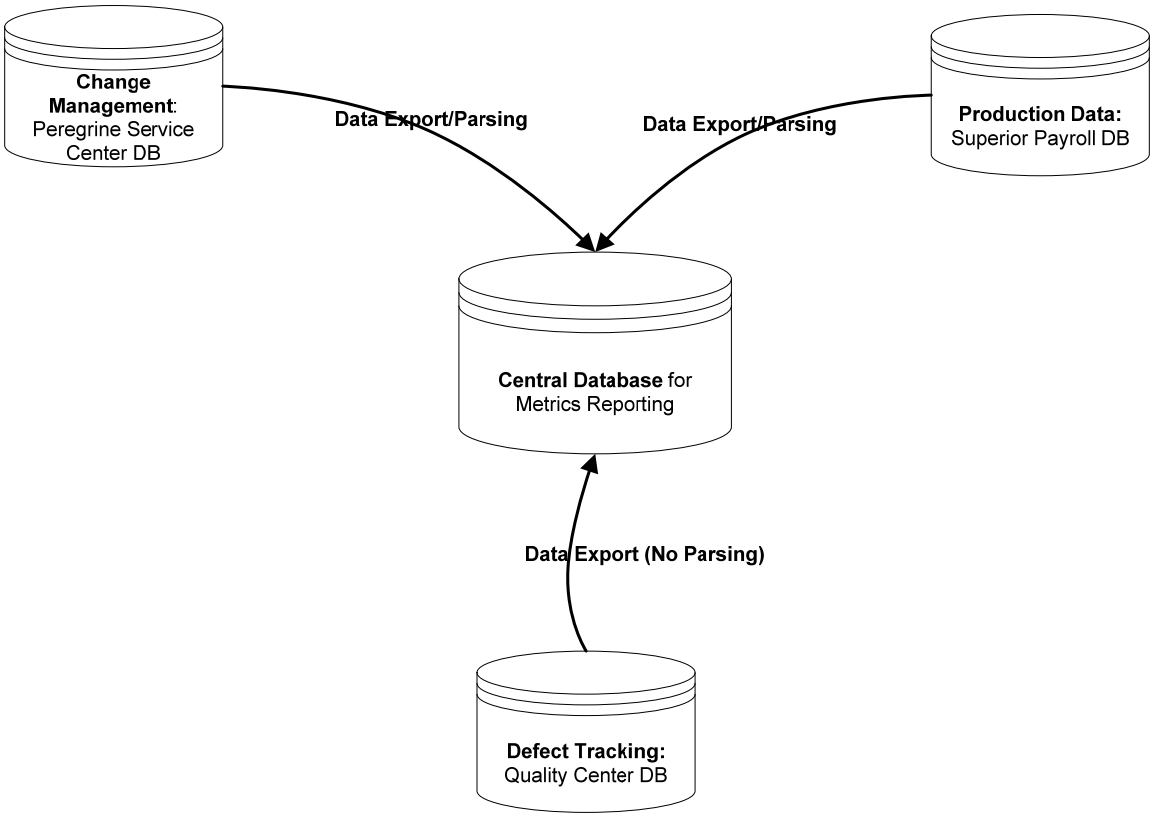

Figure 2: Database Migration

The level of automation of this tool cannot yet be determined. Using tools that are already available within the organization, technical solutions will be implemented to bridge the gap between the Payroll Report Card and where the data currently exists. A format will be identified, including graphs, charts and additional statistics. Figure 3 represents the link between the Central Database, the customization of the Payroll Report Card and the production of the report itself.

It is assumed that future iterations of the metrics report will be completed; however, the initial implementation of the report is what this project will focus on.

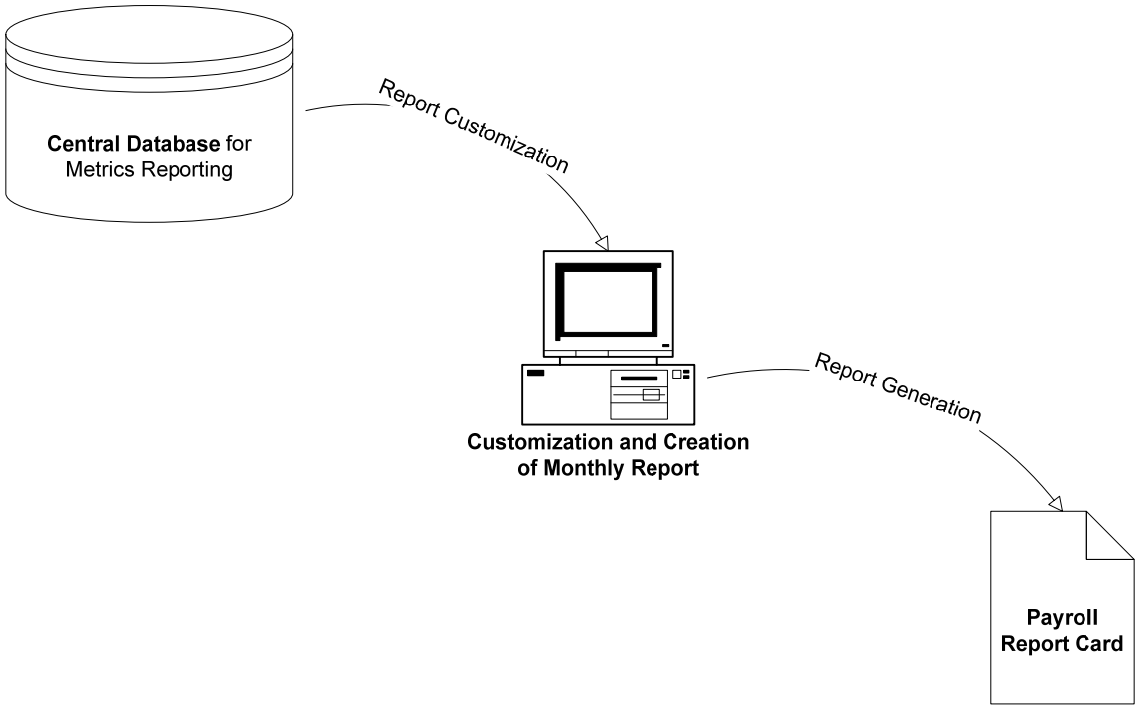

Figure 3: Automation with Customization of Payroll Report Card

Definition of Terms

Central Database – A new database that will be created with this project. Its purpose is to house data from multiple databases that contain relevant information for metrics reporting.

Change Management – A company policy/procedure for meeting Sarbanes-Oxley regulations. All company work is scheduled through change management as *change requests*.

Change Requests (CR) – The collection of issues that exist in change management (Afora International, 2007). Each piece of work that is scheduled must be done so in the form of a change request.

Freeware – Software that can be downloaded from the internet for free usage. There is no fee for the license.

Payroll Report Card – The automated metrics report that is being implemented with this project.

Peregrine Service Center – The software tool that Superior Payroll Solutions implemented as the solution to change management and Sarbanes-Oxley legislation. Change requests are logged in this software and serve as the enabler to software work.

Production Application Databases – A large collection of databases that house client and employee-related data from the company's production systems. This can also be referred to as live data.

Quality Center – Industry-leading software that is licensed by HP and manages defect tracking, requirements and test scripts. Superior Payroll Solutions uses this software for defect-tracking only.

Acronyms

CM – Change Management

CR – Change Request

OS – Operating System

QA – Quality Assurance

Summary

This project will implement an automated metrics reporting tool, called the *Payroll Report Card*,. It will serve management through providing a metrics report on a quarterly basis. Data will be exported from multiple databases and housed in the central database. A format will be identified and displayed each time the report is run. Only minor customizations will be required by management, as to keep preparation time for the report to a minimum. A very limited budget will require that in-house development

be employed. This means that freeware and already available applications will be researched for use in this report. This project is expected to serve as an example to other companies who lack metrics reporting and an ideal budget to implement it.

Chapter 2: Review of Literature and Research

Research Sources

Management Experience

This project requires management's involvement, because they will be presenting the quarterly report to senior management. Each of the managers that are involved in the project have over ten years of experience in software development. The three managers are as follows: The Application Development Manager oversees the Development and Quality Assurance team; The Support Manager oversees the Support and Implementation teams; and, the Operations Manager oversees the Business Representatives for the company.

All three managers know the product well and have experience in the roles they manage. Therefore, their knowledge is a source for the project's research and decision-making.

Internet Vendor Documentation

Internet software vendors were researched, using white papers or other documentation, if available. The team was considering the use of available online products for components of the reporting tool or as an all-inclusive solution. This source refers to online vendors who produce freeware, shareware or products by license.

Books/Publications

Two specific publications have been used to implement a methodology for analyzing and identifying good metrics for the organization. *Best Practices in Software Measurement* by Ebert, Dumke, Bundschuh and Schmientendorf served as a strong

source for adopting good metrics. *Software Metrics: A Guide to Planning, Analysis, and Application* by C. Ravindranath Pandian was also used as a guide to collecting and analyzing measurements in order to assure they were meeting the goals of the organization. Object Oriented and Classical Software Engineering, by Schach, was also used, but a lesser extent.

Literature and Research that is Specific and Relevant to the Project

In order to identify effective metrics for an organization, there must be an analysis that connects each measurement to a specific corporate goal (Ebert et al, 2005). The goals of an organization must be declared before being able to establish effective metrics.

The first step to implementing an effective metrics system is extracting the necessary data (Ebert et al, 2005). Once corporate goals are identified, data that supports those goals must be located and extracted. Extracting the data is important, as the information should be isolated from its original location (Pandian, 2004). This allows collected information to be free from production influence. In other words, the data is frozen to assure that the timeframe and data is accurate.

Once data has been extracted, it must then be evaluated (Ebert et al, 2005). Evaluating the data requires that the information be displayed and compared with organizational goals and benchmarks (Ebert et al, 2005). This is the comparison between actual data and projected goals. Reports should be generated and detailed analyses of the results should be discussed.

After the data has been extracted and evaluated, change must be executed to influence improvement in the organization (Ebert et al, 2005). If an organization is not prepared to enable change based on metrics results, then the value of the process is lost

(Ebert et al, 2005). Although the execution lies at the end of the metrics process, this is something that should be discussed and agreed upon early in the process. It is also important to note that an adequate timeframe must exist after change is implemented so progress can take effect. If there is not sufficient time between analyses, employees can become discouraged and progress can stall.

Aside from methodology research, specific charts were researched and used during the project to assure that the goals of the organization were aligned with the data that was going to be collected. Examples of these include a Goal Tree and a Decision-Making Hierarchy. These should be created and reviewed prior to the extraction of data.

Goal trees can be used in order to trace organizational goals to sub-systems and specific processes (Pandian, 2004). Goal trees list corporate goals and show how they connect to decision centers and systems within the organization.

Decision-Making Hierarchies can be used to display the organization's layers of decision-making (Pandian, 2004). It links those decisions to levels of metrics to show where change will be needed based on results (Pandian, 2004). This sets the expectation that specific decision-makers will be responsible for change on specific levels.

Research was also done to examine any pre-packaged software metrics systems. The internet was used for this source of research. Measure Foundry is license-based software that provides the ability to customize a test or measurement system without development (Measure Foundry, 2007). The system supports data consolidation and measurement capabilities. The cost of the basic software package is $1995, and the professional package is $2595 (Measure Foundry, 2007).

Microsoft Dynamics is a software system that integrates data for customer service and supply-chain management (Microsoft Dynamics, 2007). The software can be utilized for service companies, such as payroll companies, and can serve as metrics-based software to improve business processes. Although the software is not driven toward defect management, it could be used in collaboration with Quality Center reporting. The pricing options for Dynamics can be determined through one payment or on a "per user" basis, using a 36-month payment process (Microsoft Dynamics, 2007). For one server, the software cost ranges from $1,244 to $1,761. When financed over a 36-month period, the cost can range from $20 to $28 per user per month.

Research was also done on software that the organization already had licenses for. Microsoft Word, Excel and Access were examined to see if data reporting features would be sufficient. No source was used for this, as the team had experience with each product.

Summary of What is Known and Unknown

The three primary sources of metrics data exist in the Peregrine Service Center database (change management), the Quality Center database (defect tracking tool), and the production databases that serve as the information systems for the Superior Payroll product. Due to company security policies, the reporting tool will be unable to directly access data in a read-only fashion. Sarbanes-Oxley company policies protect these databases from being directly accessed by groups that do not exclusively own the data. Therefore, the data owners from each group have agreed to export data from their respective databases so that the metrics team can import all data into one central database.

Availability of technical resources is unknown for the duration of the project. Although management has designated a metrics team, unforeseen priorities could impact the timeline. Furthermore, unforeseen technical expertise could require the request for new resources if needed.

Due to budget restrictions, management has only provided resources in the form of team members who may have higher priority work throughout the duration of the project. It is not known if management will be able to approve software and other technical expenditures. This will be determined during research sessions based on cost and alternative approaches.

Project Contribution to the Software Industry

The value of metrics is known by many companies worldwide. Despite proven success with metrics in the software industry, there are still many companies that have not taken the steps to implement adequate reporting. This project will demonstrate that metrics reporting can be implemented, even with a limited budget and resources.

Large and small companies can use this project as an example of an organization that had a very limited budget and partially dedicated resources, and in eight weeks, was able to implement a metrics solution. Although the data and company-specific details will push this project in a definitive direction, the approach can be used within any organization. This will be achieved through the adoption of a specific metrics methodology and technical/in-house customization.

Chapter 3: Methodology

Research Methods to be Used

There were three phases of research methodology: Metrics Gathering Methodology; Decision-Making Methodology; and Metrics Data Management and Systems Configuration. Text books were primarily used to support the methodology of research in each phase.

Phase 1: Metrics Gathering Methodology

The goals of the metrics project had to be identified upfront. Once the goals of the organization were established, further analyses were conducted to identify decision-makers and relevant data needs. This was achieved through declaring a Goal Tree (Pandian, 2004).

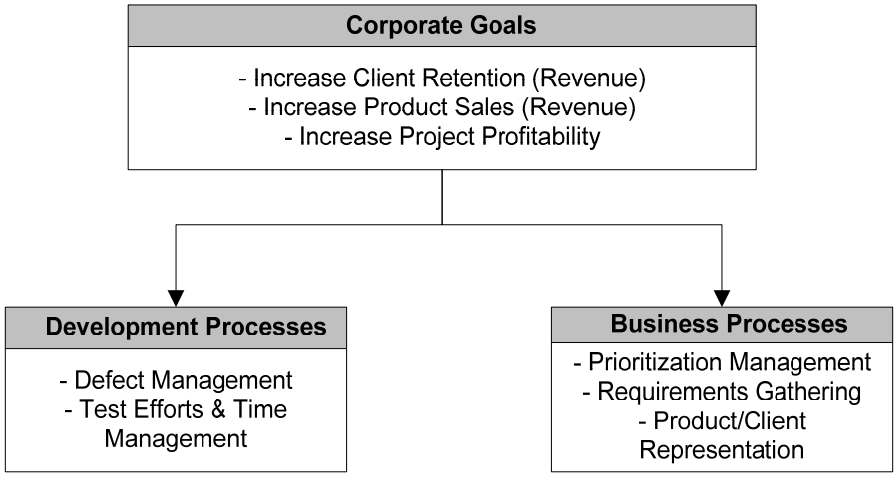

Figure 4: High-Level Goal Tree

Meetings with management determined that increasing client retention, product sales and project profitability were the goals of the organization. These corporate goals were listed and linked to specific processes on either the Development side or the Business side of the organization.

It was determined that client retention and product sales could be directly linked to product enhancements and defect remediation. Superior Payroll relies heavily on accurate calculations and rules that follow tax laws. Defects that cause late or insufficient payroll transactions often cause fines or fees for the client. Furthermore, product limitations disallow sales from pursuing sports teams and international companies. Therefore, defect-related measurements provide project-level quality analyses for the organization. Business representatives provide change requests, and therefore, are responsible for prioritization management. Since change requests are ranked based on priority, an analysis can be provided on change requests completed versus change requests that are outstanding. This will exhibit the prioritization efforts of the business representatives.

Increasing project profitability also focuses on prioritization, but it is representative of strong time management, good client representation/requirements and coordinated testing efforts as well. In order for a project to be successful, good system requirements are needed. This means that the business representatives need to provide detailed requirements. The company's Analysis and Requirements group documents details exactly as they are conveyed. Therefore, the burden of good requirements falls on the business units. Coordinated test efforts also provide strong project profitability. Since testing is done between two separate groups, Quality Assurance and Implementation, test efforts should be coordinated. Each group should know what kind of defects will be logged, as they represent the types of testing that are being performed.

Phase 2: Decision-Making Methodology

Before metrics are gathered, research needed to be done on who has what decision-making power. This research would identify the decision makers so that metrics could be organized and linked for future execution of change. Therefore, the decision-makers for each category of metrics needed to be identified and correlated. The research of this effort is displayed in the following figure (Pandian, 2004).

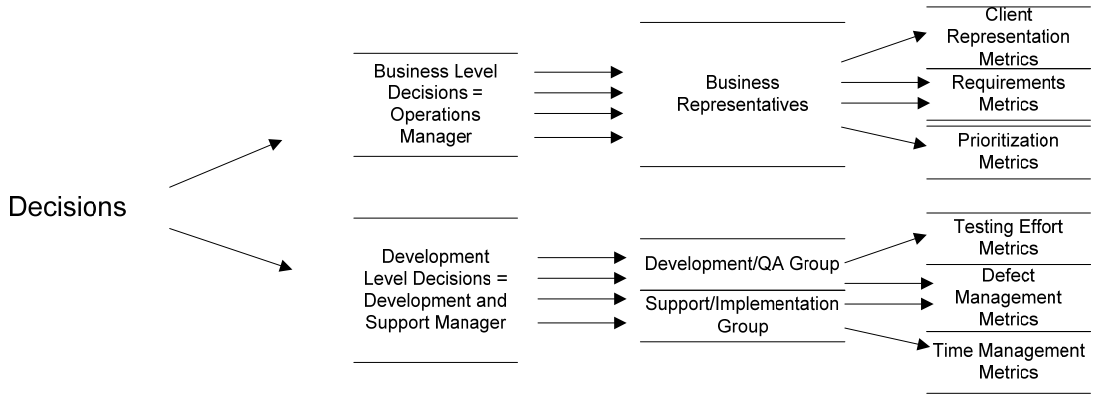

Figure 5: Decision-Making Hierarchy

Figure 5 shows that Business decisions are directly linked to the Operations manager, who is responsible for the Business Representatives. Client representation, requirements and prioritization metrics represent data that is a result of the choices and actions of the Business Representatives.

The Development decisions are made by the Development and the Support Manager. Because there are two groups that provide testing, the decision-making responsibilities fall under two separate groups. Both groups are responsible for testing, defect and time management metrics that are a result of the efforts of Development and Implementation groups (Implementation falls under the Support structure).

Phase 3: Metrics Data Management and Systems Configuration

Now that metrics have been linked to the organization's goals and the decision-makers have been identified, the methodology for capturing and organizing the data in a software system must be outlined.

Due to budget constraints, senior management will not allocate funds to support the purchase of the researched software solutions from *Chapter 2* (Measure Foundry and Microsoft Dynamics). Therefore, the metrics system must be developed in-house, using available software and equipment.

To achieve data extraction from multiple systems, a set of algorithms must be implemented into a healthy export and store procedure. The model below details how the metrics system will be structured:

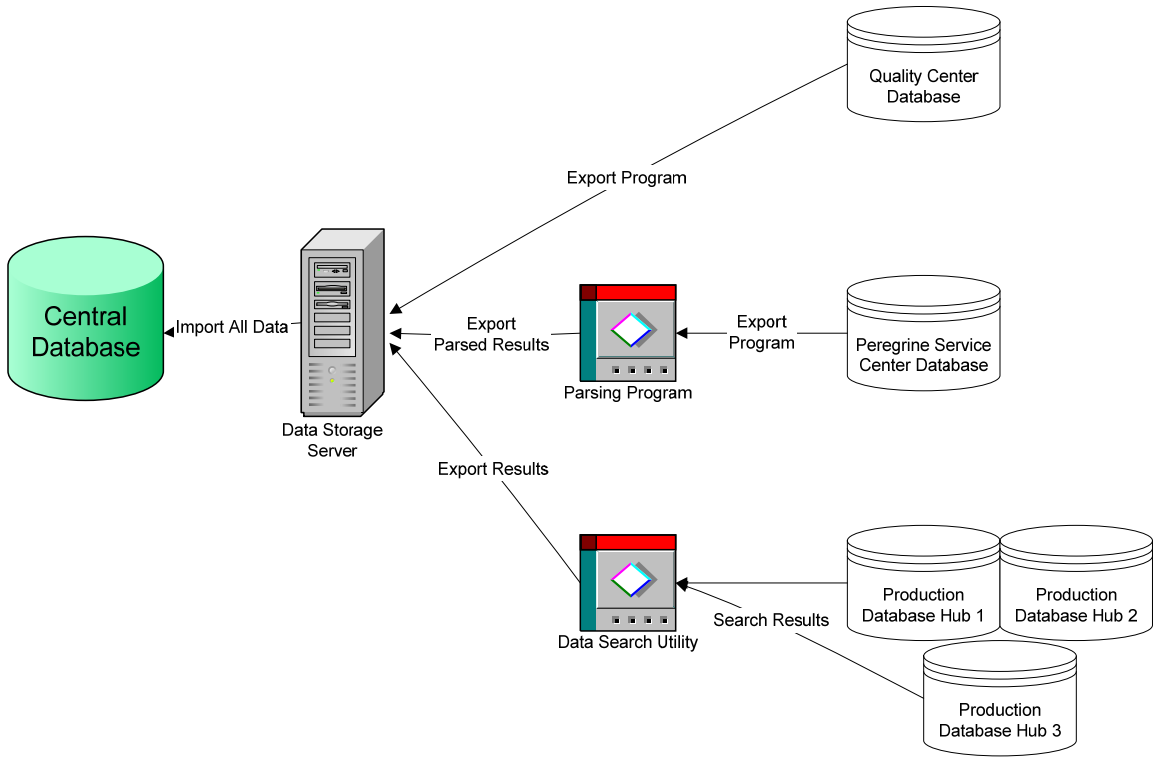

Figure 6: Metrics Dataflow Model

Figure 6 displays the databases on the right side of the diagram. Quality Center data can be directly exported because the data is stored in an Oracle database by User Interface (UI) field. This is why no parsing program is needed for this application. Peregrine Service Center contains non-relational data in a proprietary database. The database contains fields of database strings that are not organized by comprehensive naming conventions or UI fields. Therefore, a parsing program is needed to separate data based on metrics needs. The production databases contain data from the Superior Payroll product. Because the product is a nation-wide product, the data is organized by hub, or processing center. A tool called the Data Search Utility was created in 2005 due to Support needs. The tool meets security policy requirements, and therefore, will be used to gather data from the production hubs. This data must still be exported after being gathered. No parsing is needed, because data is stored in database fields according to UI field.

All of the data that is exported must be placed on one secure server, in a specific directory. The Metrics team will create an import tool that takes all exported data and moves it into the Central Database. The Central Database is where the Payroll Report Card will pull its data from.

Microsoft Excel is the program of choice to cut and carve the metrics data for graphical configuration. This is because Excel grants more flexibility and ease with importing and displaying information. Microsoft Access will not be used because the features of the program are more structured and not as versatile. Graphs, charts and calculations will be made with Excel, and the results will be linked to a Microsoft Word document to display the full report. These programs have been chosen because they

contain all of the functionality and simplicity that is needed to tailor the metrics data for reporting.

Life-Cycle Model to be Used

Upon evaluating the life-cycle model to be used for the project, there are two potential models that fit the organization's common practices. One is the Waterfall Model. This model is commonly used within the organization for development projects because of dependencies within individual units of code. If code is deployed to an environment and there are dependencies, the Superior Payroll product could become dysfunctional.

The other model that is occasionally used is the Iterative Model. This model is used for smaller projects, only when code can be deployed in increments. By delivering the code in pieces, the testing effort can progress in parallel to the development effort. This model also requires strong analytical skills and planning from an architectural standpoint (Odhinn, 2007).

Upon evaluating the code delivery for the project, the developer recommended that the Waterfall Model be used. This is because QA would not be able to test the project in smaller iterations. Therefore, all members of the team agreed to employ the use of the Waterfall Model throughout the Software Development Life Cycle (SDLC). The Waterfall Model allows for requirements gathering and documentation before design and implementation begin. When requirements change, the project team will loop back and document any changes before re-development takes place (Schach, 2005). The model for this project is illustrated in *Figure 7*.

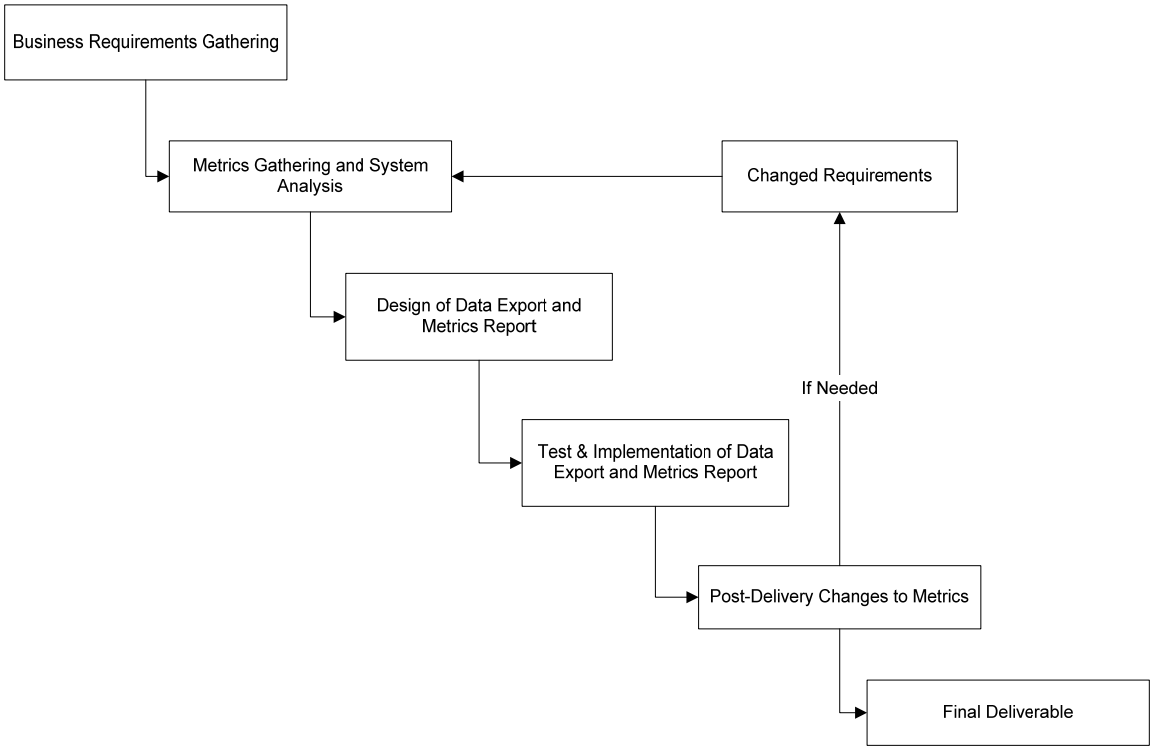

Figure 7: Waterfall Model Details

Project-Specific Procedures

Using the detailed research from *Section 3.1*, the project will require a design for the Payroll Report Card. Business requirements for the functionality of the report will be gathered from management. The requested features and usage of the Payroll Report Card will also be documented.

The metrics for the report will be gathered in the Analysis phase. These metrics will be collected by collaborating with management and product specialists. After metrics are gathered, the team will work to identify metrics data. It will be necessary to determine how that data is going to be migrated and displayed. This also requires that the project team will need to meet with the data owners of the individual systems that data is needed from.

After requirements are gathered and the analysis is complete, development will begin coding and the QA team will start creating the needed SQL statements in order to display the data from the central database. Because the data is going to be organized and displayed with Excel, the QAs will work on formatting the graphs with temporary data. Once development turns the code over to QA, Excel will be pointed at the external database and the data will be pulled to form charts, graphs and calculations for display on the report. QA will then need to test the data import and parsing programs in the Central Database. Tests will also be needed to assure that the data in Excel is accurate. Test and remediation efforts will need to be completed before the system can reach the implementation phase.

Implementation of the project will be achieved once the daily data exports and parsing programs are feeding data into the central database, and all logged defects have been resolved. At this point, all programs will be officially placed in "production".

Although the team will make an attempt to gather requirements in full during the Analysis phase, management is expected to occasionally request an additional or altered metric that the system does not display. Therefore, it may be necessary to change requirements, provide an analysis, re-document requirements, re-develop the product, and then re-implement the report and/or data export programs.

The final product will be delivered during the Final Deliverable phase. Any future changes will be done as another iteration/phase of the project.

Project-wide procedures will include weekly team meetings to discuss status, milestones and to review the project's timeline. Work for this project will be conducted

whenever team members have time available. The project timeline and milestones will be created based on projected availability.

Format for Presenting Deliverables

Because the Waterfall Model was used, documentation served as the main source of deliverables (Schach, 2005). Documentation was created to reflect company standards, meaning that the formats and content were consistent with other company projects. Additionally, one prototype was delivered to management to assure that the project team was heading in the right direction. The final deliverable was the Payroll Report Card.

Review of Deliverables

Throughout the project, specific documentation was delivered for each phase. These details are in the following table:

Project Phase	***Documents***	***Purpose***
Business Requirements Gathering	*Payroll Report Card Requirements Document*	This document identified the functionality and basic design of the report. The focus was placed on how much work would need to be done each time the report was run. Also, the look and feel of the report was documented. Included in this document was the Decision-Making Hierarchy and the Goal Tree.
Metrics Gathering and System Analysis	*Metrics Analysis Document*	This document detailed the specific metrics that were to be used for the report. For example, how many payroll failures did the product suffer during Q1? What is the purpose of displaying this metric? This document detailed how these metrics would be calculated and why.
	Software Design Document	Although the project would not deliver a full software product, documentation was needed to display where data was being exported from, what data was being parsed and how it was going to be migrated into the central database. The document was fairly simplistic, including the Metrics Data Flow Diagram.
	Payroll Report Card Prototype	This prototype displayed what the format would look like. Management needed to approve of this design before the project continued.
Design of Data Export and Metrics Report	*SQL Index*	This was a document that housed all SQL statements that would be used to cut and carve the data in the central database. This is exhibited in *Appendix A.*
Post-Delivery Changes to Metrics	*Metrics Analysis Document*	All Post-Delivery Changes were made to either expand metrics or discard specific metrics. Therefore,

		changes would simply be made to the Metrics Analysis Document. This is represented in the Project-Specific section, where each metric is discussed.
Final Deliverable (Simultaneous Deliverables)	*Payroll Report Card*	The Payroll Report Card was delivered. This included the report and the spreadsheets with the calculations and SQL statements. This is what management creates custom reports with.
	Central Database and Data Exports	This is the underlying structure that is separate from the Payroll Report Card. This includes the central database, the data exports and the parsing programs.

Table 1: Review of Deliverables

Required Resources

This project has a very limited budget. A team of five members (including myself) will be partially dedicated to the project. The team members on the project are as follows:

Position	***Proficiencies***	***Major Deliverables***
Project Manager and Quality Assurance Lead (Myself)	Experienced project lead. Experience with SQL, Excel, and product-specific data.	The direction and management of the Payroll Report Card. Report formatting, including charts and graphs design.
Quality Assurance Specialist	Expert Excel user. Strong skills in SQL and reporting capabilities. Some product knowledge.	Initial Excel setup. Early prototypes of the report.
Database Specialist	Expert in SQL and database technologies.	Central Database Administration.
Developer	Experienced Developer.	Import and parsing program. Small changes to export tool for production data.
Product Specialist	The most advanced expert on Superior Payroll. Over 10 years of experience with the product and it's functionality.	Help in identification of data that would represent the required metrics.

Table 2: Team Resources

Because of financial constraints, there is no budget for the purchase of software or equipment. However, existing software tools and licenses are available. This will result in a large amount of in-house development to reach goals.

Project Outcomes

After the Payroll Report Card is delivered, management will be able to view organized metrics that relate to the Superior Payroll product. Management will be able to customize the display of the data through Excel's functionality, while having the latest

data refreshed automatically. The immediate outcome will enable tangible metrics that are linked to process and product health.

The desired long-term outcome for the project (outside the scope of this project) is to enable procedural changes to improve quality. Reviews of the metrics will indicate a number of things that could include a lack of software quality, inadequate business prioritization, poor requirements or the duplication of efforts. This will be discussed in more detail in *Section 5.4*.

Summary

This project will employ the use of the Waterfall Model for its Software Development Life Cycle. Initial research has been done to examine the goals of the organization, as well as to identify the decision-makers for the metrics results. This will enable the proper organization of data for the Payroll Report Card. Furthermore, research has been provided to outline the data flow for specific data. Three systems will require an export and the Central Database will contain all needed data. Excel will be used to graphically display the data, and the results will be linked to Microsoft Word for customization and printing needs. Although resources are fairly limited, most work will be completed through the use of available software and partially dedicated resources.

Chapter 4: Project History

Project Beginning

This project began as an initiative from the Development, Support and Business Operations groups for the Superior Payroll product. As project bandwidth was expanded, both groups began experiencing a decrease in quality requirements and a lack of properly prioritized Change Requests. This was causing escalated work to become more common. As a result, the QA and Implementation groups were becoming less effective due to changing requirements and an overlap of testing efforts.

The Payroll Report Card concept was discussed with Senior Management, and was determined to be the first of a number of successive metrics reporting initiatives. The purpose of the first initiative, which this project represents, was to create the Payroll Report card based on the necessary metrics data.

Project Management

The author served as the Project Manager for this project. The author worked directly with management for overall direction of the Payroll Report Card. Meetings with management led to the identification of organizational goals and high-level metrics. Resource allocations were provided by management, and directives were given to the author.

A kickoff meeting and weekly status meetings were chaired by the author and attended by each of the other four resources. A high-level list of milestones and dates were established to help manage the project's progression. Meetings with non-team resources were conducted with the data owners of Peregrine Service Center. A close working relationship was needed to facilitate the export of confidential data.

The author followed the Waterfall Model's process of requirements gathering, analyses, design and test, product implementation and the final deliverable. Some requirements changes were needed, as expected.

Project Milestones

The milestones for this project will be presented in phases according to the Waterfall Model.

Milestone Name	Description	Owner	Due Date
Pre-Project Discussions	Author met with management to discuss scope, deliverables and details.	Author	4/2007
Project Kick-Off	Kick-off meeting with team members to discuss project scope.	Team	4/2007
Payroll Report Card Requirements Document	Requirements needed to be gathered regarding the format and functionality of the report.	Author	5/1/2007
Database Research	The team knew that data would need to be retrieved from three databases. Discussions needed to take place between the project team and external resources.	Team	5/2007
Security Guidelines	The author and developer met with the security organization to discuss data exchange constraints.	Author and Developer	5/2007
Metrics Analysis	Deliver the document that outlines the metrics and data that are needed. The team worked on this with management approval.	Team (namely the Product Specialist)	5/18/2007
Software Design Doc	Basic document showing the design of the import, exports and parsing program.	Developer	5/24/2007
Initial Data Export	This represented the export programs created by the project team and the expected delivery of the export for the Peregrine Service Center system.	Developer and External Resource	6/6/2007
Payroll Report Card Prototype	Completion and delivery of the Payroll Report Card to management for approval.	Author and Quality Assurance Specialist	6/6/2007
Parsing Program	This was delivered simultaneously with the import program and Central Database creation. This was development's last large effort.	Developer and Database Administrator	6/25/2007
Import Program	Import program to take the data export files and import them into the Central Database.	Developer and Database Administrator	6/25/2007
Central Database Creation	Central Database was created and ready for data import.	Developer and Database Administrator	6/25/2007
SQL Index	SQL queries were created so they could	Author, Quality	6/25/2007

	just be entered into Excel.	Assurance Specialist and Database Administrator	
QA Testing	Data was tested to assure the imports, exports and parsing program were working correctly.	Author, Quality Assurance Specialist, Database Administrator	7/16/2007
Implementation	The product was done with testing and management was notified that they could access it.	Team	7/16/2007
Post-Delivery Changes	One change was requested, and some changes needed to be made.	Author and Developer	7/19/2007
Final Deliverable	Payroll Report Card was approved by management.	Team	7/23/2007

Table 3: Milestones Overview

Business Requirements Gathering

Upon meeting with management, there were two types of business requirements that were requested. The first type of business requirement was the functionality of the report. The Payroll Report Card needed to have an automatic refresh of data each time it was run. Parameters would be needed to customize the report, but the data in the Central Database and Excel should always be updated. A daily update of the information was required. Also, the ability to enter parameters for specific time periods was required. Management wanted to be able to run for any quarterly release or year. Only one release period needed to be included per report, so that each release could be focused on exclusively.

The second type of business requirement that was requested was specifications related to the report's format. Management wanted a report card look to the Payroll Report Card. Goals needed to be included on the report and there needed to be a designated location for comments on each metric. This would allow for report customization for each release. This would also allow for management to provide a more in-depth analysis of the data, including any additional research that was completed.

Furthermore, there needed to be built-in functionality to control the goals for each metric. This would allow for goal inclusion based on management direction.

Metrics Gathering and System Analysis

Before Development could begin creating exports, imports and databases, metrics needed to be gathered in order to identify data needs. With management direction and approval, metrics were gathered based on the organization's goals. The metrics that were identified were related to the company's Change Management System, Defect Tracking System and Superior Payroll Production Systems. The metrics will now be discussed individually.

Number of Open Change Requests by Ranking:

This metric displays the number of Change Requests that exist on the product's list for maintenance. It also displays the number of changes that exist for each level of *Rank*. Rank is used to indicate the priority of each planned Change Request. This is indicative of both the level of quality development and quality prioritization, based on other metrics in the report. The goal is to reduce the number of high ranking Change Requests on the maintenance list. This metric was generated using a count of all Change Requests that did not have a *ProductStatus* of *Released*.

Number of Completed Change Requests by Release and Ranking:

This metric collaborates with the previous metric to show how many high ranking changes are being scheduled each quarter. In order to decrease the number of high ranking Change Requests on the maintenance list, the Business Representatives must be scheduling as many highly ranked changes as possible. By improving prioritization, a

better quality product exists in production and client retention is improved. This metric was generated by counting the number of Change Requests by *Plan* and organizing the numbers by *Rank* where *ProductStatus* equals *Released.*

Number of Completed Change Requests by Release and Cause:

This metric focuses on the number of Change Requests that are being completed on a quarterly basis and what the cause of each change is. This is another metric that could potentially identify Development or Business quality. If there are a several Change Requests on each plan that have a cause equal to "Defect", then too many defects being missed by Development, QA and Implementation. If a majority of Change Requests are being completed with a cause equal to "Requirements", then it shows that Business Representatives are not delivering accurate and/or completed requirements. This metric is displayed using a count of Change Requests by *Plan* and organizing the numbers by *DriverOfChange* where *ProductStatus* equals *Released*.

Number of Open Change Requests by Priority:

The priority of a Change Request shows whether the change can wait to be "Planned", which would then receive a *Rank*, or if it must be delivered immediately. Escalated Change Requests must be delivered within five business days, whereas, Emergency Change Requests must be delivered within 24 hours. Because these types of changes take time for Support to troubleshoot, they will exist on the maintenance list until assigned to Development. Therefore, this metric displays the number of Planned, Escalated and Emergency Change Requests that exist on the maintenance list. This is directly indicative of quality. This is displayed by counting the Change Requests and organizing the numbers by *Priority* where *ProductStatus* does not equal *Released*.

Number of Completed Change Requests by Release and Priority:

This metric is used as a comparative analysis of the number of open Change Requests by priority. This shows how much of Development's bandwidth is being used for high priority work. Although this is a reflection of quality development, it can also affect the Business Representatives' ability to prioritize higher ranking work for planned quarterly release. This is displayed by counting the Change Requests and organizing the number by *Priority* and *Plan* where *ProductStatus* equals *Released.*

Number of Completed High Priority Change Requests by Release and Application Feature:

This metric identifies either Escalated or Emergency Change Requests, and what application feature they are related to. The reason for this metric is to identify quality and time management. If a lot of simplistic features are being completed through high priority work, then Support may be contributing to poor time management. If a majority of high priority work is related to more complex processes, then quality may not be meeting expectations. It is expected that management will examine specific Change Requests if needed, in order to support this metric. This is displayed by counting the Change Requests and organizing the number by type of *Priority*, *Plan* and *SubApplication* where *ProductStatus* equals *Released*.

Number of Requirements Updates by Release and Date:

This metric evaluates the number of times requirements were updated in Peregrine Service Center based on a date range. The reason for this metric is to identify poor requirements gathering by evaluating the number of times they were updated beyond the expected requirements lock-down date. When requirements are updated

closer to the release date, it is done so because of an unexpected change or a missing detail. This negatively affects quality and represents poor client representation and insufficient requirements gathering. This is displayed by counting the number of records that have a *RecordType* equal to *RqmtUpd* and *LogDate* equal to a specified date range.

Number of Defects Detected by Release and Group:

This displays the number of defects that were logged in Quality Center by quarterly release and group (e.g. Development, QA or Implementation). This indicates the level of quality within specific testing groups. It was determined by the company that Development and Quality Assurance is accountable for 80% of all defects. The Implementation group, which acts as a safety net, is accountable for 17% of defects. The other 3% of defects are considered an acceptable rate of failure based on priority and ranking, which is displayed using other metrics. This metric is displayed by counting the number of Quality Center defects and organizing the number by *DetectedByType* and *Quarter* where *LogType* does not equal *Build Request*.

Number of Defects Detected by Release, Group and Defect Type:

This metric is similar to the previous metric, except that the defect's type is focused on. Upon discovering defects, each tester must determine if the defect is pre-existing, a valid defect or an enhancement. If one particular testing group is logging too many types of enhancements or pre-existing issues, then this is considered a time management issue. A defect is only supposed to be logged if it needs to be fixed on the current project. A small portion of enhancements will be implemented and all pre-existing issues must be fixed in the future. This is to control the scope of the project. This is displayed by counting the number of Quality Center defects and organizing the

number by *DetectedByType*, *Quarter*, and *LogType* where *LogType* does not equal *Build Request*.

Number of Defects Detected by Release, Group and Application Feature:

This metric displays the number of defects detected by application feature and group. This is indicative of Development's quality. If there are an excessive number of defects logged for *Screens*, then basic GUI functionality is being missed during Unit Testing. Because there are multiple levels of testing, it is important to identify what level of testing is responsible for what types of defects. This facilitates time management and quality accountability. This is displayed by counting the number of Quality Center defects and organizing the number by *DetectedByType*, *Quarter* and *SubApplication* where *LogType* does not equal *Build Request*.

Average Process Times by Date:

This metric examines production data to identify the average process times for specific product-related processes. One example of this is the amount of time it takes to process the average payroll. If process times are increasing, what is the acceptable rate of increase? Although this standard has not been established yet, it is expected to be declared with the Payroll Report Card. This is another indicator of quality and good requirements. The performance expectations should be established by the Business Representatives through assuming client expectations. Development should also be involved in determining this expectation/requirement. It is then Development's responsibility to meet those expectations as a requirement of the project. This is displayed by importing all the process-related times and categorizing them. Excel is then used to create an average of all the process times for the respective processes. This

metric requires that the *ProcessTime*, *ProcessType* are imported where *Jdate* equals a specific date range.

Design of the Data Export and Metrics Report

After determining the high level metrics for the Payroll Report Card, the project team needed to work together to identify specific data needs. This meant that the team would take the previously identified metrics categories (both Business and Development metrics) and derive where the data would come from, and what calculations/comparisons would be required.

Peregrine Service Center, the company's Change Management Software, contains the details of all work that gets scheduled within the organization. This detail allowed the project team to track what, why and when Change Requests are scheduled. The specific fields that were required are listed in *Table 4*.

Data Field	***Description***
Plan	Specifies what quarterly/project plan (e.g. Q1, P1, Q2 etc.) the Change Request would be scheduled for.
Priority	Determines if the Change Request can wait to be "Planned" or if it is considered a higher priority such as: Emergency or Escalated.
Ranking	If the Change Request is Planned, a rank is assigned to it to indicate its urgency. The Ranking is determined through a set of Justification questions.
ProductStatus	This indicates if a Change Request has been completed, closed or is still open.
DriverOfChange	This indicates "why" the Change Request is needed (e.g. normal defect, requirements change, etc.).
SubProductArea	The dominant feature of the product to be worked on. This could indicate reports, processes, screens, etc.
LogDate	This will be used to determine when a change was made to a specific field in the application.
RecordType='RqmtUpd'	The only time RecordType was needed was to identify audit history for Requirements updates.

Table 4: Peregrine Service Center Data fields

The required fields for Quality Center were identified next. Quality Center contains all data related to defect logging throughout the Software Development Life

Cycle. Development, QA and Implementation log issues to the software to report issues during the test phase of all projects. The specific fields that were required are listed in *Table 5*.

Data Field	*Description*
Project	A project falls under either Business As Usual (BAU) or a specific Project Name.
Quarter	If Project = BAU, then the quarter is required. If not, then the Quarter is not available.
SubApplication	This indicates the area of the application that the defect applies to (e.g. Reports, processing, etc.).
DetectedByType	Indicates what group entered the defect. This could be Dev, QA or Imp.
LogType	Identifies whether the reported issue is a normal defect, a pre-existing issue, a requirements change, etc.

Table 5: Quality Center Data Fields

The last source for data extraction was the production databases for Superior Payroll. Because the application contains a large set of databases, only the tables that contained the necessary data would be exported. The specific fields that were required are in *Table 6*.

Data Field	*Description*
ProcessTime	This is a log that is populated each time a payroll is run to indicate the length of time it took to process.
Jdate	This is a field that stores the date by Julian standards. This works in collaboration with ProcessTime.
Perror	If processing errors for daily payrolls, quarter end transmissions or receivers occur, this stores and exception in a log file. This contains what kind of error occurred
Pprocess	This indicates what process failed in collaboration with Perror.
Edate	This logs the error date.

Table 6: Superior Payroll Production Data Fields

After all the fields were documented, the project team began preparing for data extraction. The project's developer contacted the Change Management team and requested that an export of data (only from the needed tables) be transmitted to the SPAPPDEV31 server. A specific directory on this server is where the programs were planned to be picked up, in order to be imported into the Central Database. A naming convention was used for the file to determine if it was a file that was already processed or

not. The format used for the naming convention is as follows: DBIDBSearchDetailsPSCMMDDYYYY.TXT. The frequency for the export file was determined to be daily to allow for flexibility with running the report.

Because Development is the owner of Quality Center, the Metrics Project Team was able to complete a data export to the SPAPPDEV31 server. Quality Center works off of an Oracle Database, so the team's Developer was able to create a simple export program that consolidated all of the application's data into one export file. This was a scheduled service that would run once daily, and it uses the following format: DBIDBSearchDetailsQCMMDDYYYY.TXT

The Data Search Utility was an existing program that allowed Support to gather production-level data from different hubs for Superior Payroll. This allowed the team to avoid more Development efforts, with the exception of a small modification. The utility needed to be altered to allow for its settings to be saved. A change was also needed so that a user could specify the format and naming convention of the output file. The following format for the file must be used and saved procedurally: DBIDBSearchDetailsSPPMDDYYYY.TXT.

Unfortunately, the development was unable to schedule Data Search Utility to run automatically, because the data is located over a secured network that requires manual sign-on authentication each time. Therefore, the author had to report the issue to management. Management agreed that a larger effort was not feasible. This meant that management would have to run the utility each time before running the Payroll Report Card.

Once the export programs were routinely populating files on the server, it became apparent that Peregrine Service Center was going to require a parsing program to separate large strings of data into individual fields that could be mapped back to the UI. To save time, only the required fields were parsed. The import program was also created. It is called ImportDBISAMtoMySQL.exe.

Once the Peregrine Service Center data parser was completed, the team's Developer and Database expert created the Central Database. The MySQL database structure was chosen because it could facilitate data requirements and the expected data volume. MySQL was also chosen because it had no cost associated with it.

Test/Implementation of the Data Export and Metrics Report

Once the Central Database was created, QA assured that data was being populated using the import program. MySQL's interface was used to query the different tables so that QA could compare the data import files to the database's contents. Any issues were corrected by Development.

With the data export programs working, and the import programs picking up the data and parsing any required fields, the project team began assembling charts and graphs to display the established metrics. Excel was used to export data from an external source. One spreadsheet was used as the data warehouse for the report. Each SQL query, located in *Appendix A,* was used to import the correct columns of data. After the data existed in one of the spreadsheets, the team worked to generate PivotTables and PivotCharts for the display of metrics. One PivotTable/PivotChart would be populated per spreadsheet. The tables and charts were then linked to the Spreadsheet that was reserved for the Payroll

Report Card. This assembled all the charts and tables that would be linked to a Word document used for display and quarterly customization.

Once all data was imported into Microsoft Word, the report was placed on the server, and Implementation was officially completed.

Quarterly Procedures Phase

With the necessary metrics data automatically imported and graphically displayed, there will be two required processes that will need to be completed on a quarterly basis. First, management must identify goals for future data. These goals cannot be automatically calculated due to the unpredictable nature of diversely scheduled projects. Therefore, these goals must be determined and entered into the Payroll Report Card so that the goals can display next to the actual data each quarter. Furthermore, previous goals will need to remain so that accomplishments and failures can be identified. These will be inputs for the user.

Secondly, management will need to enter comments and manual analyses for each section. The presentation of the Payroll Report Card is expected to have a management assessment of the automated metrics, so that the correct action can be taken. This is to assure that the correct *execution* of change is enabled.

Post-Delivery Changes to Metrics

Once management was able to use the Payroll Report Card, they determined that there was one more metric they wanted to have added to the report. They wanted to the ability to view the number of failures for four different processes. This would further support prioritization and quality issues.

The author worked with the team to add process failures to the documentation, and then evaluate whether more data was going to be needed from the Superior Payroll production databases. After evaluating, the required data was already in the logging table that was being used to detail process times. The additional metric is detailed below.

Fatal Failures by Process Type:

This metric identifies the number of fatal errors that have occurred in the production environment for Superior Payroll. This examines the product's health, much like the previous metric, but focuses more on evaluating quality and prioritization efforts. The Business Representatives are responsible for prioritizing the most critical product defects/limitations. If lower priority work is being scheduled, but there is an abundance of fatal process failures, this may be an indication of poor prioritization. However, this may also be an indication of poor testing efforts. It is important to note that this metric will also need further investigation into the details, as the numbers do not necessarily provide a clear and consistent answer to the problem. This metric is displayed by counting the number of process failures, by process. This is done through counting instances of *Perror* and organizing it by *Pprocess* where *Edate* equals a specific date range. Once the metric was added to the report, the changes were tested and implemented onto the correct server.

Project Plan Alterations

The project team was dependent on the Peregrine Service Center export program. The export program was delivered two weeks later than originally planned. Furthermore, complications with the format of the export program caused additional development on a parsing program. The parsing program separated large strings of data and inserted a

delimiter so that it could be stored in the Central Database according to UI field. This also expanded the amount of QA testing that was needed. Because the project was scheduled to be completed over a month before quarter-end, the report's delay had no impact on management's request.

Were Project Goals Met?

The goals for the project were to identify metrics according to organizational needs, and then to create an automated reporting tool that would give management the required data. Although Superior Payroll's production data cannot be scheduled to automatically import data into the Central Database, management is still not required to manually move data. The Data Search Utility can have its settings saved so that management only needs to login and click "Run". From there, the Central Database will automatically import the data at noon each day.

The Payroll Report Card contains metrics based on organizational goals and management direction. The report contains up-to-date data, which was not directly accessible due to security policies. Through import, export and parsing programs, the data is readily available in the report. The entire project was completed through in-house efforts using existing software. The only resources that were used throughout the project's lifecycle was the extra time that the team dedicated to this effort. It can be stated that the goals for this project were successfully met.

What Went Right?

The project team was able to obtain all required data for the report. Although direct access to the data in the respective databases was unattainable, export programs were able to grant access to updates on a daily basis.

Microsoft Excel had all of the functionality that the project team needed in order to graphically display the data. Data could be imported and refreshed automatically. Furthermore, the graphs could be linked to a Word document that is able to open and display on any computer.

What Went Wrong?

The project team ended up having to rely on another data owner for the Peregrine Service Center export. Once the team received the export, extra development was needed to isolate the necessary data. This delayed the development and QA efforts. This also means that if extra data is needed from Peregrine Service Center in the future, the parsing program will need to be altered each time.

Approximately half way into the project, two specific team members became overwhelmed with higher priority work. Although this did not impact the project's timeline, the project team was fairly negative and anxious to reach completion. Because the milestones for the project were broken down by deliverable/task, it appeared that progress was moving slowly. This was because project progress was dependent on other groups or decisions from management. This perpetuated the frustration and overwhelming nature of the project at this time.

Project Variables and their Impact on the Project

Because the author had a lot of previous knowledge and foresight into the details of this project, the project variables did not have as much of an impact on the final deliverables.

The company's security policies created difficult hurdles to overcome regarding data access. Gathering data through imports and exports drastically increased the amount

of work that needed to be done. It would have been ideal to have direct data access to get more up-to-date statistics. The parsing program was time-consuming and frustrating to implement. Complex data strings caused confusion and became a large risk factor. Furthermore, the team was only able to parse the data that was going to be used in this initiative. This means that the parsing program will have to be modified each time new data is used in the Payroll Report Card.

Another variable was related to the format of the report. Even though direction had been given to the author at the beginning of the project, management disapproved the first prototype and requested a different design. This caused additional time to correct, and added to project frustration. The author proposed the prototype in phases to assure that the team's direction was accurate the second time.

Analysis of Project Results

Although the project was accomplished for the Superior Payroll product, it still only provides a working metrics solution for one of the company's product teams. The other two product teams still do not have any way of measuring the product's quality. This proposes a difficult question that is likely to complicate IT governance within the organization. An ideal effort would have required that a project team from each product line work to implement a metrics solution that is aligned with consistent management evaluation. This would have ensured that change is enabled to improve quality processes for all products.

Because middle management is in control of the data and the customizations, it is possible that the metrics will be looked at incorrectly. Metrics are comprised of statistics that can be interpreted in a number of different ways. It would have been ideal to have

generated a metrics report that is reviewed by management that is one level above the individuals who are directly responsible for creating the data.

Despite the concerns that have been discussed, the goals of the organization have been aligned with the metrics that are being reported through the Payroll Report Card. Management has agreed that the specific metrics have value for enabling change. Furthermore, analyses on the product's health can be substantiated instead of debated based on intangible perceptions.

Summary

The project was a success based on the goals and scope that were discussed in the first two chapters. The Payroll Report Card was successfully automated to be run on a quarterly basis. No budget was appropriated, but in-house efforts delivered the product using existing software and resources. The goals of the organization have been documented and aligned with the metrics that are included in the report. Despite some unexpected security and data issues, the project was completed on-time due to extraordinary efforts of the team members.

Chapter 5: Conclusions

Lessons Learned

When working with team members who do not have a vested interest in completing the requested deliverables, it can be difficult to manage a timeline. Had the author communicated the need for a resource to complete the Peregrine Service Center export, management may have been able to assist in resolving the matter. At times, this resource was difficult to contact. There were also times where the Peregrine resource was hesitant to provide data needs. This was partially the cause of the export being delivered late, which led to the team's frustration.

Another lesson that impacted the project was the gathering of metrics from statistical data. There were complications with getting management to agree on data to be included in the report. Statistical data is not necessarily explanatory nor is it ideal. Sometimes there is no clear answer as to how to provide the best measurement. Further analyses and discussions of the quarterly Payroll Report Card will be needed to dissect the data. Change may not be so easy to achieve, as disagreements on the data are expected to exist. This specific complication was prevalent during discussions on how to evaluate the Superior Payroll's production health. There was also a point of conflict when determining how to evaluate quality requirements.

The Superior Payroll's health was solved through evaluating critical processes' failure rates and process times. Quality requirements were solved through identifying late requirements updates and defects that were logged through requirements being re-worked.

It was also quite an experience to lead a group of resources who already have a full workload. Luckily, the developer, who had the largest workload, had the easiest schedule. It can take extra motivation to keep resources focused and on-track for success.

What Would Have Been Done Differently

Because of the explicit instructions by management, there was not much that could have been done differently throughout the project.

The difficulties with the Peregrine Service Center resource may have been minimized had the author discussed the situation with management. Also, the milestones could have been broken down differently to allow for more phasing and frequent deliverables. Documents were frequently delivered, and considered a milestone, but there was a long period of time before actual code was delivered. This caused the Developer to feel overwhelmed and unaccomplished. The development deliverables should have been spaced out more to give the developer a break in the timeline.

Were Project Expectations Met?

The expectations for the project were met. The Payroll Report Card was delivered to management with the metrics that were approved. The report was automated with the exception of having to run the Data Search Utility based on its saved settings. All other data is automatically exported, imported and parsed for display, as requested.

What is the Next Stage of Evolution for the Project?

The Payroll Report Card is sure to be a work in progress in the future. It is expected that management will want to optimize data and metrics after meeting with senior management. Specific data will possibly be altered, replaced or removed if senior management does not agree with the defined measurements.

The technical configuration is also a concern. Management would like to expand the functionality of the report to be able to be run by multiple users over a larger shared network. Capacity concerns, such as having more than one instance of the report open, will need to be addressed as the report becomes more popular.

Was the Project a Success?

The project was a success, as the Payroll Report Card was delivered with established metrics and the approved format. The project was completed with no budget for an organization that previously had no established metrics.

Conclusions and Recommendations

Although implementing a metrics reporting tool can improve process and product quality, many companies still struggle to incorporate such a system. A majority of organizations suffer from the same complications as the one in this study does. Data can be difficult or nearly impossible to gain access to. Security policies can create such roadblocks. This does not mean, however, that the data is unobtainable. Each organization that faces such a challenge, should work to examine security policies. Allowing data owners to create their own data deployment solutions could meet policy procedures.

If an organization uses its software development resources, it can be possible to create a metrics reporting tool with no additional budget. Through customizing an import, export and parsing program, Excel can be used to retrieve data out of a MySQL database and perform the necessary calculations, graphs, charts and logical formulas needed for such a tool. There are many other programs that provide similar functionality without high profile purchases.

When implementing a metrics reporting system, it is best to do so from a senior management level. By enabling such a system on a higher level of management, IT governance is more likely to align lower-level departments and products with consistency. It is essential that management execute the necessary change, based on the reported metrics. Without change, the extracted data will only serve as information with no means for improvement.

Summary

The project was a success, despite the small setbacks that were discussed. All goals and requirements were met. The true impact of the Payroll Report Card is yet to be realized. The project's investment will be seen in the future, which will be directly tied to any change that is made because of its findings.

Appendix A: SQL Queries

Count Open Change Requests by Ranking:

```
SELECT  Ranking, count(ChangeRequestNumber)
FROM changerequests c
WHERE c.productstatus <> 'Released'
AND c.productstatus <> 'CR Rejected'
group by ranking;
```

Count Completed Change Requests by Release and Ranking:

```
SELECT c.Plan, c.Ranking, Count(c.ChangeRequestNumber)
FROM statsdb.ChangeRequests c
WHERE (c.ProductStatus='Released')
GROUP BY c.Plan, c.Ranking
```

Count Completed Change Requests by Release and Cause:

```
SELECT Plan, DriverOfChange, count(*)
FROM changerequests c
WHERE (c.ProductStatus='Released')
GROUP BY c.driverofchange, c.plan;
```

Count Open Change Requests by Priority (Escalated/Emergency vs. Planned):

```
SELECT Plan, Priority, Count(*)
FROM ChangeRequests c
GROUP BY c.plan, c.priority;
```

Count Completed Change Requests by Release and Priority (Escalated/Emergency vs. Planned):

```
SELECT Plan, count(*)
FROM ChangeRequests c
WHERE c.ProductStatus = 'Released'
group by c.Plan;
```

Count of High Priority Change Requests by Release and Application Feature:

```
SELECT c.Plan, Count(*)
FROM statsdb.ChangeRequests c
WHERE (c.ProductStatus='Released')
GROUP BY c.Plan
```

Defects Detected by Release, Group (Development, QA or Implementation) and Type:

```
SELECT d.DetectedByType, d.Quarter, count(*)
FROM defects d
WHERE d.logtype <>'Build Request'
GROUP BY d.quarter, d.detectedbytype;
```

Defects Detected by Release, Group and Application Feature (Reports, Help Files, etc.):

```
SELECT d.DetectedByType, d.SubApplication, d.Quarter, Count(*)
FROM statsdb.defects d
WHERE (d.LogType<>'Build Request') AND ((D.detectedbytype) In ('DEV','QA','IMP'))
GROUP BY d.Quarter, d.DetectedByType, d.SubApplication;
```

Defects Detected by Release, Group and Defect Type:

```
SELECT d.Quarter, d.LogType, d.DetectedByType, Count(*)
FROM statsdb.defects d
WHERE (d.LogType<>'Build Request')
GROUP BY d.Quarter, d.LogType, d.DetectedByType
```

Average Process Times by Date:

```
SELECT p.ProcessTime, p.Jdate, p.ProcessName
FROM statsdb.ProdProcesses c
WHERE c.Jdate < ##
GROUP BY p.ProcessName, p.ProcessTime
```

Retrieve Process Fatal Errors by Date and Error Type:

```
SELECT p.Perror, p.Pprocess, p.Edate, Count(*)
FROM statsdb.ErrorTable p
WHERE p.Edate <> ##
GROUP BY p.Perror, p.Pprocess
```

Requirements Updates by Release and Date:

```
SELECT c.Plan, c.RecordType, c.LogDate
FROM statsdb.ChangeRequests c
WHERE c.RecordType=’RqmtUpd’
GROUP BY c.LogDate, c.Plan
```

Appendix B: Payroll Report Card Sample

Payroll Report Card

Metrics Detail Section

Metric Overview:	***Future Goal***	**Comments**
This chart shows the number of CRs on the identified BAU quarterly plan by Driver of Change. The Driver of Change can be defined as the type of change or the reason for the change. For example, legislative CRs would indicate that it was scheduled due to a payroll law (which are often mandatory).	*Decrease the number of Repair/Fix Change Requests by 10%. This will be displayed on next quarter's report graphically.*	*An evaluation needs to be done to see how many Repair/Fix CRs existed for more than one year. The concern is over how many Repair/Fix CRs that are a result of recent project work.*

References

Afora International (2007). SCM Glossary of Terms. Retrieved on November 25, 2007 from http://www.afora.nl/gl_change_request.shtml

Dr. Dobb's Portal (2007). Testing & Debugging: Software Development Metric Tools Updated. Retrieved on July 19, 2007, from http://www.ddj.com/dept/debug/199902969

Ebert, Christof, Dumke, Reiner, Bundschuh, Manfred, Schmientendorf, Andreas (2005). Best Practices in Software Measurement. Berlin, Germany. Springer-Verlag Berlin Heidelberg.

Odhinn (2007). Iterative Development and the Leaning Tower of Pisa.. Retrieved on March 19, 2008 from http://www.fromthetrench.com/2007/01/21/iterative-development-and-the-leaning-tower-of-pisa/

Measure Foundry (2007). Buy Measure Foundry. Retrieved on January 12, 2008, from http://www.measurefoundry.com/purchase.asp

Microsoft Dynamics (2007). Supply Chain – Customer Relationship – Financial Management Software. Retrieved on January 12, 2008, from http://www.microsoft.com/dynamics/overview.mspx?mg_id=10189

U.S. Department of Labor (2004) Computer Software Engineers. Retrieved on July 17, 2007 from http://www.bls.gov/oco/ocos267.htm

Wiegers, Karl (1999). A Software Metrics Primer. Retrieved on July 19, 2007 from http://www.processimpact.com/articles/metrics_primer.html

Pandian, C. Ravindranath (2004). Software Metrics: A Guide to Planning, Analysis, and Application. Boca Raton, Florida. Auerbach Publications.

Schach, Stephen R. (2005). Object Oriented & Classical Software Engineering. New York, New York. McGraw-Hill.

Syntelligence (2005). The Metrics Game: Framework for Definition and Implementation. Retrieved on November 24, 2007 from http://www.syntelinc.com/syntelligence/index.aspx?id=627