Fall 8-1-2006

Customer Extranet Metrics

Marianne C. Gorecki
Dakota State University

CUSTOMER EXTRANET METRICS

A graduate project submitted to Dakota State University in partial fulfillment of the requirements for the degree of

Master of Science

in

Information Systems

August, 2006

By
Marianne C. Gorecki

Project Committee:

Dr. Stephen Krebsbach
Dr. Ronghua Shan
Tracy Sterner

DAKOTA STATE dsu UNIVERSITY

MSIS
PROJECT APPROVAL FORM
(Form #3)

Student Name: Marianne Constance Gorecki

Master's Project Title: Customer Extranet Metrics

Signature Approval:

Faculty supervisor: ______ Date: 12/4/06

Committee member: ______ Date: 12-1-06

Committee member: ______ Date: 12-10-06

Original to Office of Graduate Studies and Research
Acid-free copies with written reports to library

11/28/2006

ABSTRACT

Pass/Fail and Success/Failure are both terms we use to assess the impact of a specific task or project. Measuring items in a subjective, repeatable manner is key to assessing current status. The question to be asked is "How do you know that a project has been a success without using a metric?". Success can not be measured unless it can be realized. Metrics can not be realized unless they are defined. This paper will review an existing extranet site that lacks defined measurements and will discuss the issues in the current environment. These issues will be analyzed and converted into solutions. These solutions will have an execution plan defined for success and will have a review of the results and lessons learned.

DECLARATION

I hereby certify that this project constitutes my own product, that where the language of others is set forth, quotation marks so indicate, and that appropriate credit is given where I have used the language, ideas, expressions or writings of another.

I declare that the project describes original work that has not previously been presented for the award of any other degree of any institution.

Signed,

Marianne C. Gorecki

Marianne C. Gorecki

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

INTRODUCTION

An integral part to the success of any project, particularly a project in the IT industry is performance reporting. These metrics or Key Performance Indicators are tools used to assess the impact of a particular program, project or activity. These metrics usually are numeric values such as increasing customer satisfaction by 20%. However, these metrics can also be qualitative such as improve customer satisfaction. The latter would be achieved by a measurement that trends upward month over month.

Some may ask "What will metrics give me?". To an organization, tracking metrics will provide valuable information and benefits such as:

- Reducing subjectivity in assessing program status
- Providing leverage for funding for growth
- Employing accountability by program owners for success or failure of activities
- Identifying issues before becoming major problems

This paper will present you with the business scenario and the issues relating to the current business process. These issues will be discussed and solutions will be identified. A step-by-step execution plan will be laid out followed by the project results. Next steps for this project and lessons learned will conclude this paper.

Background

A secure customer extranet, created in the fall of 2003, provides easy access to valuable information, hot fixes, a variety of learning options and other online resources designed to make customers more productive. Since its original implementation, this site has undergone visual and functional enhancements; all designed to make the user experience better and more productive. This extranet is also a center point for a variety of programs areas including documentation, training, premium services, customer support and maintenance. The overarching goals of the extranet are to increase knowledge transfer with timely and accurate information providing a lower cost of ownership and higher quality implementations for the customer. In the upcoming section, issues with the current extranet processes will be identified and discussed.

Problems

Currently, there are almost 3 full years of data stored in the database. Some programs have less than others due to a rolling deployment of functionality. In taking over responsibility for this extranet, history of the site was reviewed through the documentation provided by the program's predecessors resulting in the identification of four main issues:

1. In depth metrics were non-existent
2. Analysis was primarily based on estimation
3. Limited availability for management to access data
4. Raw data was confusing to understand

Problem 1: Metrics

Important metrics to the web community include traffic measurements, transactions and download activities. These types of metrics quickly and easily show the value of a program to the community.

As stated previously, there are multiple years of data that are available for analysis. From historical documentation, it appeared that analysis had never been done on specific elements of the site. Such analysis would have provided a baseline measurement to measure success as enhancements and programs were introduced to the community.

Problem 2: Estimation

Measurement provides specific success criteria for projects. Through the use of measurements baselines and targets can be set which can be used later to demonstrate progress, repeatability and auditing of success and failure.

Success of elements of the site was primarily construed on opinions and educated guesses. This method of measurement would give a false sense of accomplishment and/or failure dependent on the 'view' of the site.

Problem 3: Limited Availability

Access to and availability of measurements is the key to communication of metrics. Without proper reports and easy access to details, data is just data. Through analysis, both high-level and detailed information about the site and its elements were not widely available to owners of the program areas. Basic ad-hoc measurements were included from time-to-time in management reports, however not communicated all stakeholders of content.

Problem 4: Data

The business value of using transactional data to produce scorecards and reports is to provide a 'status' or a view into a specific elements such as progress and popularity of each program area.

The site uses a SQL-based database management system. Due to the nature of relational databases and transactional systems, the data has been normalized for greatest performance. With the use of this system, raw data can be extremely confusing. Most transactions throughout the site were contained in one table with keys referencing other tables which include the real-text data that a novice user can understand. To make the data palatable, simple and complex queries would need to be written and ran against the system.

Objectives

To solve these issues, a clear and concise list of deliverables and objectives must be identified. These objectives were extrapolated from the issues in the form of solutions. From this, four main objectives were formed: 1) Metrics, 2) Ease of Use, 3) Time Usage and 4) Communication. These objectives will be discussed in more detail in the following sections.

Objective 1: Metrics

As identified earlier, there is an absence of measurements across all program areas. To fill this void, a list of metrics is to be developed. These measurements will be identified through input from upper management and program owners as well as the extranet administrator. These values will be used to measure a baseline of each program element as well as determine success factors for each area.

Objective 2: Ease of Use

Data was identified as being complex. A user must have at least basic SQL skills in order to query basic, meaningful information. To address this issue, a warehouse of data, will be collected for easy viewing and better access to information. This warehouse will contain data that has already been aggregated into meaningful measurements, thus allowing for the non-SQL user to access and make sense of the data captured.

Objective 3: Time Usage

Although, not identified as an issue, administration of reporting evolves into an objective. If all proposed metrics were to be run through a manual process then updated into a master table and/or graph, the time encapsulated in a monthly cycle could exceed 4-6 hours each month. To eliminate excess effort, an automated system will be produced to reduce processing time of metrics of as much as 90% each month.

Objective 4: Communication

Finally, as stated earlier, communication of metrics throughout the management team and staff is non-existent. To increase communications, a scorecard report to display metrics will be produced, in compliance with corporate standards, to communicate with team members, stakeholders and management. This scorecard will be published each month as part of the status report and will be displayed in a public area for employees to view.

To recap, this project will have 4 main deliverables based upon issues identified and objectives defined. These deliverables are:

- A list of metrics to be measured once per month.
- A data store will be created to house denormalized data that will be easy to use and access.
- An automated update and report creation process will be identified and developed to save administrative time.
- A scorecard report will be published monthly and will be included in monthly status reports as well as distributed within the company for staff to access.

SYSTEM DESIGN

Each deliverable for this project must have a clear scope defined in order to determine this project as complete and as a success. Each goal has at least one element that needs to be defined in scope. Factors in determining the scope of this project were the measurements needed, the level of detail provided for each measurement and the mechanism for sharing information. This section will show the associated scope definition for each objective.

Scope

Item 1: Measurements

There are a plethora of potential measurements contained within the transactions of the extranet. What metrics are important and meaningful to management? In order to determine which measurements are the right measurements, interviews with stakeholders and management will be executed. Proposed measurements will then be compared against the data for feasibility of capture. Any exceptions will be noted and reported to management.

Item 2: Level of Detail

How will we measure the details? The level of detail of each metric will determine the method of aggregation as well as the table design in the data store. Through discussions with stakeholders, this level of detail will need to be identified. Possible levels of customer detail include user level, role level, company level and program level. It was determined from this exercise that the program level was acceptable. The data gathered would be summarized at the program level with corporate values being excluded where appropriate.

Item 3: Sharing Mechanism

A scorecard is of no use if it does not provide the audience with a method to view the data as well as have the ability to find out details of each measurement in a timely manner. A mechanism to produce and to share measurements at the appropriate level of detail will need to be put in place.

Options include Excel or Access as data collection tools and Excel, Access and Crystal Reports as reporting tools. After examination of the options, Access was chosen as the platform for ease of use and for data access. By using Access stakeholders will have the ability to review the de-normalized tables to understand the details behind the measurements produced. Crystal Reports was chosen as the reporting tool as it gave the greatest flexibility in accomplishing the project goals.

With scope defined, the next step is to determine how these deliverables will be executed and scope expectations met. In the next section, the execution phase of this project will be defined and planned.

Plan Execution

Data Research

The first step in setting the execution plan is to research and understand the data. In this process, it was noticed that most tracking data was being stored on transactional tables. These transactional tables contained a variety of fields containing multiple pieces of information such as an ID and an action merged together. The database had been normalized. To achieve 'meaningful' information; tables had to be joined to obtain the desired outcome.

Interviews

After a through understanding of what data could be mined, informal interviews with stakeholders were scheduled. These discussions took place around what information would be useful in reporting the usage of each program area. Interviews took place with program level managers, directors and senior management. For examples of the questions asked, see Appendix A. The next step in the plan is to design the system.

Design

Decided during the interview process, information was to be summarized at the program-level in most cases. However, some metrics defined such as *Company with Most Users* and metrics that corporate needed to be filtered out for the final report, therefore company identification information was decided to include. It was also concluded that this information should be published on a monthly basis.

Data, with ease-of-use in mind, was denormalized into tables specific to a metric. A high-level data model was created to better understand what tables would need to be created. Figure 1 displays the main measurements. For a full page image, please refer to Appendix B.

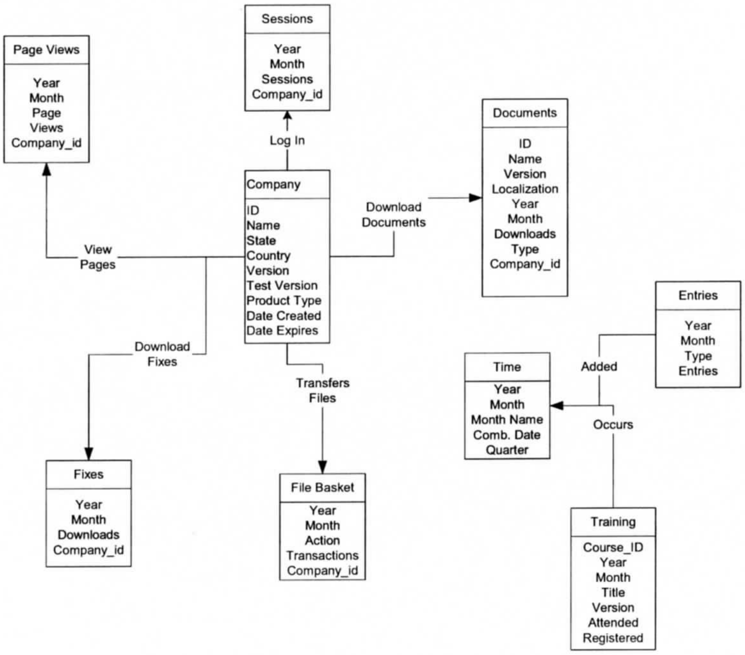


Figure 1. High-Level Object Relationship for Access Database

Some measurements were broken down further when built in Access. For example, one measurement is the *Number of General Document Downloads by Month* for non-Corporate users. This data was represented in one table design in Access with table fields shown in Table 1. For details on all tables and their respective data elements, please refer to Appendix C.

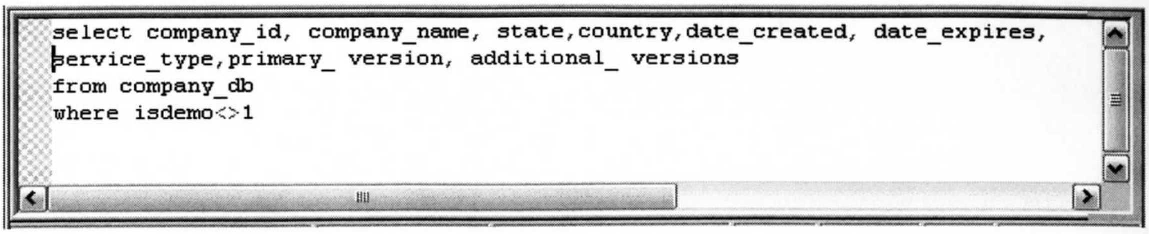
Table 1: Example of data grouping by measurement.

Field Name	Data Type	Description
Table: General_Docs		
File_id	Text (10)	Unique ID for a document
T_name	Text (100)	Name of the document
T_version	Text (10)	Version the document is for
T_local	Text (10)	Localization of the document
N_year	Integer	Year download took place
N_month	Integer	Month download took place
Company_id	Text (10)	Unique ID for a company
N_count	Integer	Number of downloads for the given time period

Development

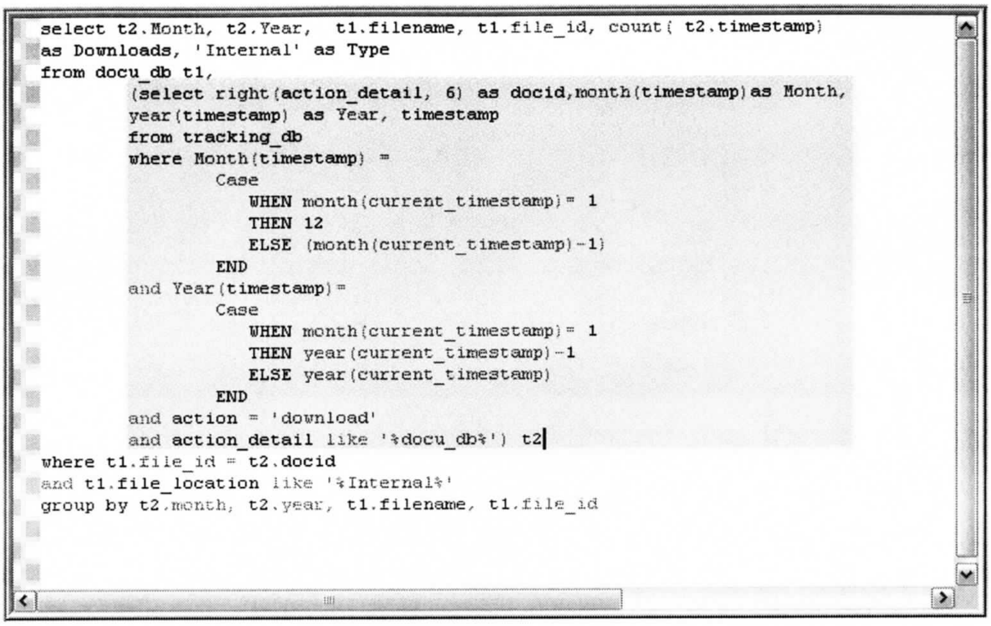
After all the Access tables were designed, a query or set of queries was developed for each measurement. Textual measurements such as *New Users* and *New Users this Month* required multiple queries where as other graphical measurements required complex queries.

Overall, to accomplish the measurements set by stakeholders, there are a total of 45 queries pulling data from various parts of the extranet all compiled in Access. Queries range from the simple queries as shown below in Figure 2. To queries that utilize SQL functions such as case statements, sub-queries and the right function (Figure 3).



```
select company_id, company_name, state,country,date_created, date_expires,
service_type,primary_ version, additional_ versions
from company_db
where isdemo<>1
```

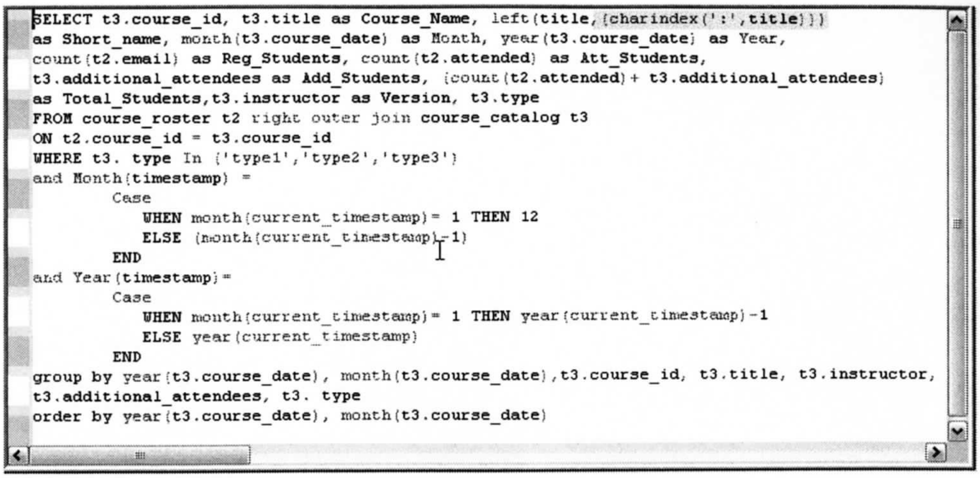
Figure 2. Simple Query Example



```
select t2.Month, t2.Year,  t1.filename, t1.file_id, count( t2.timestamp)
as Downloads, 'Internal' as Type
from docu_db t1,
        (select right(action_detail, 6) as docid,month(timestamp)as Month,
        year(timestamp) as Year, timestamp
        from tracking_db
        where Month(timestamp) =
                Case
                   WHEN month(current_timestamp)= 1
                   THEN 12
                   ELSE (month(current_timestamp)-1)
                END
        and Year(timestamp)=
                Case
                   WHEN month(current_timestamp)= 1
                   THEN year(current_timestamp)-1
                   ELSE year(current_timestamp)
                END
        and action = 'download'
        and action_detail like '%docu_db%') t2
where t1.file_id = t2.docid
and t1.file_location like '%Internal%'
group by t2.month, t2.year, t1.filename, t1.file_id
```

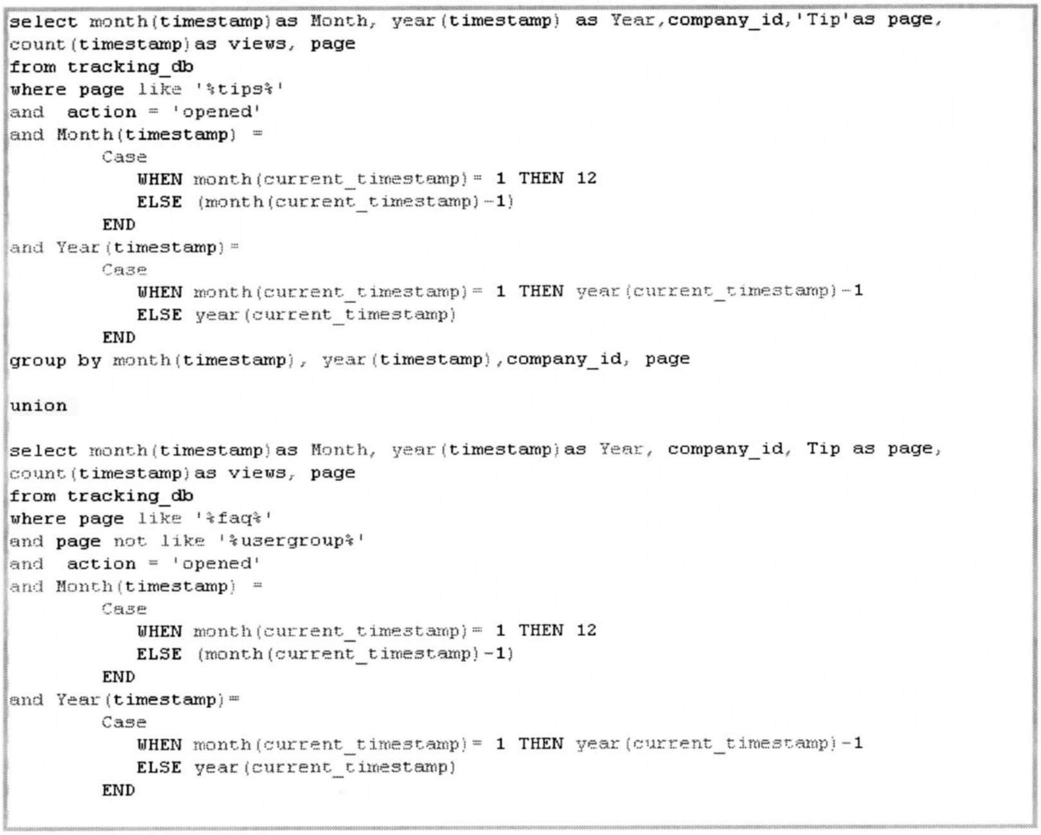
Figure 3. Complex SQL Functions Example

Other query functions used in the development process were the charindex function (Figure 4) as well as the use of Union queries (Figure 5). For information about all the queries used, please refer to Appendix E.



```
SELECT t3.course_id, t3.title as Course_Name, left(title,(charindex(':',title)))
as Short_name, month(t3.course_date) as Month, year(t3.course_date) as Year,
count(t2.email) as Reg_Students, count(t2.attended) as Att_Students,
t3.additional_attendees as Add_Students, (count(t2.attended)+ t3.additional_attendees)
as Total_Students,t3.instructor as Version, t3.type
FROM course_roster t2 right outer join course_catalog t3
ON t2.course_id = t3.course_id
WHERE t3. type In ('type1','type2','type3')
and Month(timestamp) =
        Case
           WHEN month(current_timestamp)= 1 THEN 12
           ELSE (month(current_timestamp)-1)
        END
and Year(timestamp)=
        Case
           WHEN month(current_timestamp)= 1 THEN year(current_timestamp)-1
           ELSE year(current_timestamp)
        END
group by year(t3.course_date), month(t3.course_date),t3.course_id, t3.title, t3.instructor,
t3.additional_attendees, t3. type
order by year(t3.course_date), month(t3.course_date)
```

Figure 4. Example use of the CharIndex Function



```
select month(timestamp)as Month, year(timestamp) as Year,company_id,'Tip'as page,
count(timestamp)as views, page
from tracking_db
where page like '%tips%'
and  action = 'opened'
and Month(timestamp) =
        Case
           WHEN month(current_timestamp)= 1 THEN 12
           ELSE (month(current_timestamp)-1)
        END
and Year(timestamp)=
        Case
           WHEN month(current_timestamp)= 1 THEN year(current_timestamp)-1
           ELSE year(current_timestamp)
        END
group by month(timestamp), year(timestamp),company_id, page

union

select month(timestamp)as Month, year(timestamp)as Year, company_id, Tip as page,
count(timestamp)as views, page
from tracking_db
where page like '%faq%'
and page not like '%usergroup%'
and  action = 'opened'
and Month(timestamp) =
        Case
           WHEN month(current_timestamp)= 1 THEN 12
           ELSE (month(current_timestamp)-1)
        END
and Year(timestamp)=
        Case
           WHEN month(current_timestamp)= 1 THEN year(current_timestamp)-1
           ELSE year(current_timestamp)
        END
```

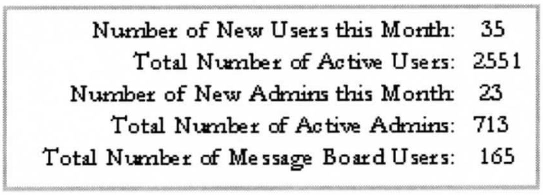
Figure 5. Example of a Union Query

Data Validation

After the development process was finished, each of the queries were validated using independent queries through Query Analyzer as well as through manual counts and visual checks from within the extranet site itself. Once all measurements were deemed accurate, the report creation process commenced.

Report Creation

With data validation complete and measurements defined, the layout of the final report had to be determined. Since there were 45 queries in all that were required in the final report, a 3-page, 8-section per page report would be the layout of choice. These sections would either contain a graphical measurement (Figure 6) or a set of textual measurements (Figure 7).



Number of New Users this Month:	35
Total Number of Active Users:	2551
Number of New Admins this Month:	23
Total Number of Active Admins:	713
Total Number of Message Board Users:	165

Figure 6: Textual Measurement Example

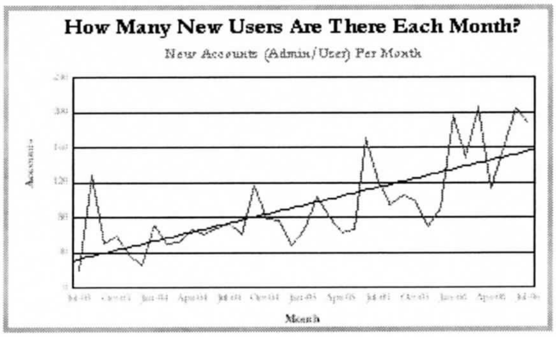
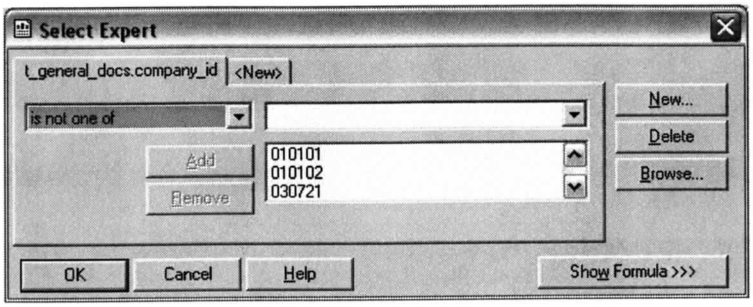


Figure 7: Graphical Measurement Example

In order to accomplish the 8-sections per page layout, each data element had to be created independently as a sub-report. In all, 30 sub-reports were created. For each report, filters were defined were necessary to exclude corporate values (Figure 8). As a requirement from management, linear trend lines were added to each report as seen in Figure 7. The final design was achieved be pulling all 30 sub-reports into one report using Crystal Reports (Figure 9).

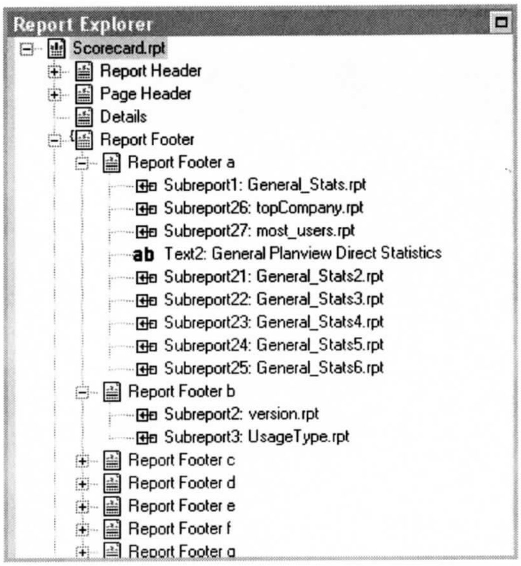


t_general_docs.company_id

is not one of

010101
010102
030721

Figure 8: Example of Filtering through Crystal Reports



```
Scorecard.rpt
  Report Header
  Page Header
  Details
  Report Footer
    Report Footer a
      Subreport1: General_Stats.rpt
      Subreport26: topCompany.rpt
      Subreport27: most_users.rpt
      ab Text2: General Planview Direct Statistics
      Subreport21: General_Stats2.rpt
      Subreport22: General_Stats3.rpt
      Subreport23: General_Stats4.rpt
      Subreport24: General_Stats5.rpt
      Subreport25: General_Stats6.rpt
    Report Footer b
      Subreport2: version.rpt
      Subreport3: UsageType.rpt
    Report Footer c
    Report Footer d
    Report Footer e
    Report Footer f
    Report Footer g
```

Figure 9: Report Explorer Sample from Crystal Reports Showing Sub-Reports

From an aesthetic and corporate perspective, each sub-report created had to meet corporate standards as well as have a sense of commonalty between each graph. These standards included:

- Line colors to match corporate colors
- Fonts that match corporate style
- Consistent graph sizes
- Consistent graph elements – axis, title, subtitles and filter notations

Report Validation

After each report was created, it was validated against the original queries to ensure that the data presented was accurate. This was done by comparing against the base Access tables containing the summarized data as well as by using independent queries through Query Analyzer.

Variations in Plan

In comparing the execution plan to reality, various aspects of the plan differed to what was executed and/or delivered. In the original plan, user data was planned to be pulled into Access during the data collection process. During the development of the queries it was decided not to pull this information into the data structure since the scope was program level. The extra data added no benefit at this time.

In the project timeline, it was noted that the project was to be done mid-July. The project slipped from plan due to tool education, travel and a family emergency. The project ran 2-3 weeks behind schedule.

Overall, the execution plan implemented without major variations from plan with exceptions noted above. Duration estimates for each activity in the plan did not vary much from plan, although as stated earlier, the timeline had slipped.

RESULTS AND DISCUSSION

Results

With the project now implemented and reports created, it is time to look at and evaluate the final deliverables. For review the main deliverables were:

- A list of metrics to be measured once per month.
- A data store will be easy to use and access.
- An automated update and report creation process
- A scorecard report created and published monthly

The following sections will discuss each of these deliverables in detail and the associated results achieved.

Metrics

Prior to the implementation of this project, metrics had not been implemented nor had any been defined. All status reporting was accomplished by view or opinions on how someone thought a specific element was performing.

From this exercise, a set of measurements were identified and documented. Metrics ranged from individual counts such as *Number of New Users* to ranges of data such as *Document Downloads per Month.* These measurements gave each stakeholder a qualitative view into the past history of their program areas. These 'views' will serve as a baseline for future releases of content. Reaction to these new metrics has been positive and has given the overall teams a sense of accomplishment into their work.

Data store

The ability to have data that is easy to understand and access had been a roadblock in gathering the extranet data and sharing it with others. Recalling the issue, it was noted that the extranet data was stored in a normalized database as it should be. However, data in this format can be extremely difficult understand to the untrained SQL user.

In order aggregate and provide the data into an easy-to-use format, the data collected was captured into an offline Access database. This allows for access to the information without having to have access to the database server. Novice users now have the ability to mine the data easily and affectively. Users will not have to understand relational databases and will not have to have the knowledge of SQL expressions and structure.

Access to this 'data store' has not been distributed as of yet, however, it has already proven useful for ad-hoc questions on specific metrics.

Update and Report Creation

After the base queries were produced, the extraction method of data had been set. These queries were in a state to be able to run against the database in an ad-hoc manner. In order to run each query and capture the data into Excel to produce tabular tables and graphs took an excess of 4-6 hours for one month's worth of measurements. This proved that measurements, if ran in an adhoc manner, can take hours each month to generate comparable data to the scorecard in this project.

The new method of capturing and processing the data is fast and easy using Access with Crystal Reports. Reaction to the time savings and the timeliness of the monthly report has been praised.

Scorecard

As mentioned above, producing a scorecard using adhoc means can take hours. Using the new method and reporting tools, a new scorecard can be produced in a matter of minutes.

Through compilation of the scorecard, these measurements are captured and disseminated with each monthly status report. In addition, a printed copy of each scorecard is displayed in a public place for viewing as well as it is published as an internal document on the extranet. This publishing process has raised awareness of the activities and successes of all programs involved.

In all aspects, it is felt that the deliverables of this project have been met and have been met with success. The total result of this project has not only save time and effort but has provided a window into the extranet that was non-existent in the past. These affects will provide return on the programs for years to come.

CONCLUSIONS

Lessons Learned

Learning from your past experiences is key to success in future endeavors. From this, it is important to discuss what pitfalls emerged throughout this project.

Reflecting back on the work involved in creating these metrics, a consistent naming convention would of made understanding and working with the data easier. For example, *Year* should be named the same throughout all tables. In the case of this data store, it was named *Year* in some tables and *n_year* in others reflecting a numeric value of year. If caught prior to design and development, a lot of hassle in query and report creation could have been avoided.

While working on this project, it also became clear that when working on any project time should be set aside for individual development. It is imperative that the person creating the plan accounts for time needed to get familiar with the tools used in the creation and development of the final product. In this case, this was my first experience using Crystal Reports for something other than tabular data. Although my ignorance in chart and graph creation in Crystal was overcome, time for learning should have been laid out in the original project plan.

Lessons learned are positive in nature therefore these lessons will not be taken for granted. These will assist in future additions to this project and other development efforts in the future.

Future Enhancements / Additions

When a project comes to completion it is normal to think about what is next. Since the extranet is a living and active site, there will be changes to be made or added in metrics and enhancements to what has been delivered to be developed. Below you will find some of the proposed 'next steps' for this project.

Detailed Program Reports

It is common to hear "What does that measurement mean?" or "Can I see the detail behind that number?". To answer these questions, reports will need to be created that display details within in each program area that have the ability to drill into the data behind the scorecard. For example, one could drill into training topics to assist in the course retirement process. These reports will also measure the success of specific courses or documents. Other uses of these detailed reports will come in handy during a recall of a support fix or to identify a target audience for an announcement of a new program.

Detailed Company Reports

It is also important to consider the internal reporting needs on company details. Reports that are specific to companies will be useful for the sales and consulting staff. Before a site visit, the corporate representative could run a report on the specific metrics for the company. This report can be helpful in ensuring the client is aware of and has access to the valuable information and opportunities available to them.

Web Reporting

When considering reporting needs for our customers, reports that a company can run on themselves would prove useful. After a re-write of the site to a more modern architecture, the addition of dynamic reports in the form of dashboards for administrators to run against their company should be implemented. These reports would assist the administrator to better understand the level of involvement of his or her users.

Localized Reports

Currently all reporting is done containing all customer regions. Since the corporate world is international, reports specific for global regions would provide valuable data. Content that is currently captured during the set-up process can provide a country filter to allow for reports specific to America, Europe and beyond. This type of report would be of great use to understand the community presence in these areas as well as provide a useful tool for the staff and consultants when promoting the programs in non-US areas.

Detailed Goals

Finally, as stated at the beginning of this report, metrics are tools that assess the impact of a program. This project has accomplished the first step in providing qualitative metrics by identifying baseline metrics for various elements of each program area. Now that a baseline is set, the next step is to work with each program sponsor to identify quantitative goals such as increasing usage by 20% in the next 6 months. By establishing specific goals, they will have a better measurement of success or failure.

Overall, this project has been a success. Deliverables were identified and met as well as valuable lessons learned recognized. The benefits of this project will continue to be visible for some time to come.

REFERENCES

Step Two Designs Pty Ltd. (2003). Metrics for knowledge management and content management [Electronic version]

Waggener, S. & Zoppi, S. (2005). The metrics of IT: Management by measurement [Electronic version]

APPENDICES

Appendix A: Sample Interview questions

Q: Who are the individuals that will need access to these measurements and reports?

A: The program managers (stakeholders) and the management team.

Q: How do the stakeholders and staff expect to review the metrics in the scorecard?

A: A printed copy for posting and a pdf version for downloading off the site as an internal document.

Q: How far do you want the measurements to go back?

A: Measurements should represent the life of the element – for example sessions should represent the number of sessions from the birth of the site, however the number of chat sessions should represent the time that the element was released.

Q: What are the top five pieces of information you expect to receive from this scorecard?

A: Traffic measurements, download statistics, top items such as most active customer, ability to track training registrations and attendance and premium content viability.

Appendix B: Data Diagram

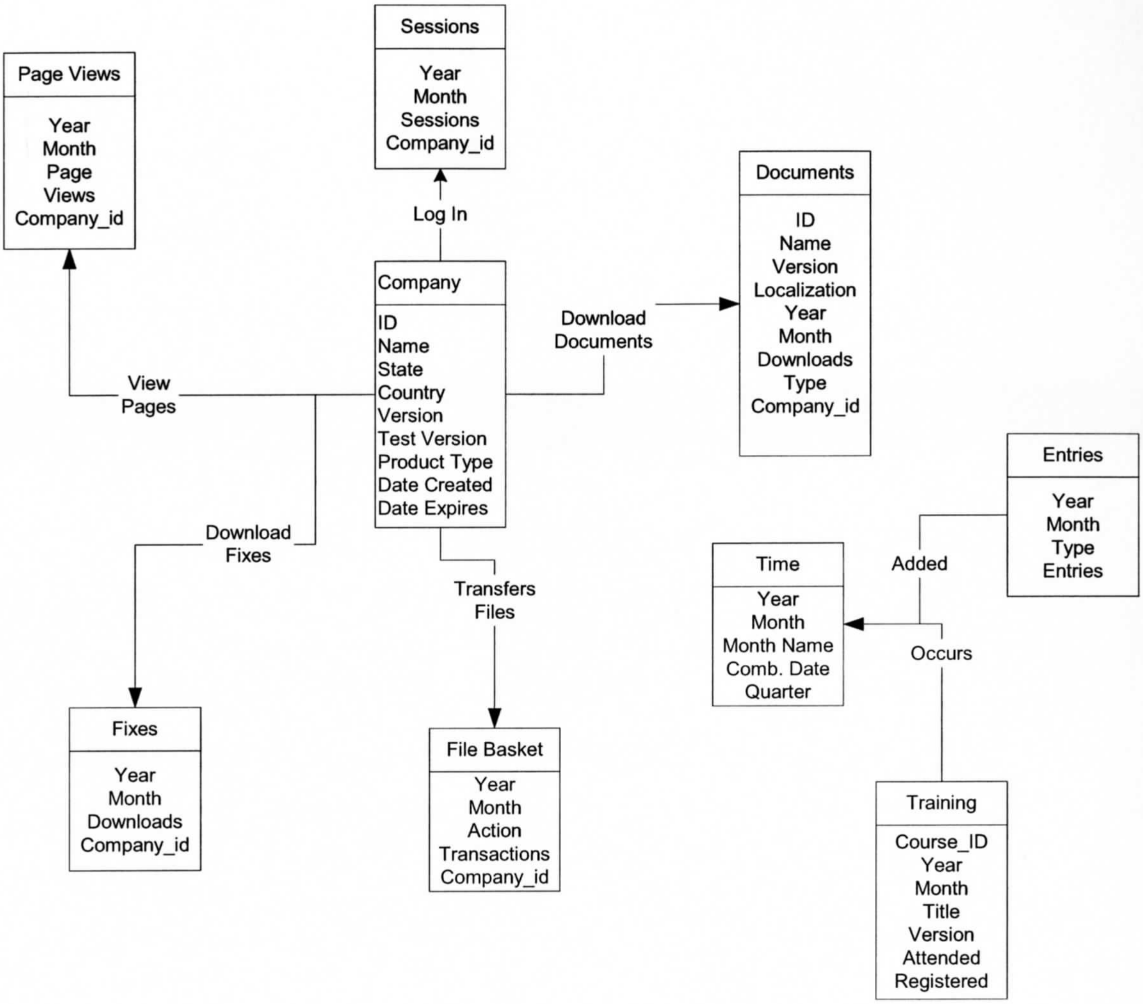


Figure 10: High-Level Object Relationship for Access Database

Appendix C: Data Table

Table 2: Table/Field representation

Field Name	Data Type	Description
Table: Company		
Company_id	Text (10)	Unique ID for a company
Company_name	Text (50)	Name of a company
State	Text (2)	State company resides
Country	Text (2)	Country company resides
Date_Created	Date/Time	Date company was created into the site
Date_Expires	Date/Time	Date company expires from using the site
Product_Type	Text (20)	Type of product company purchased
Primary_Version	Text (10)	Version of product company is using
Test_version	Text (30)	Version of product company is testing
Table: Filebasket		
Year	Integer	Year transaction took place
Month	Integer	Month transaction took place
Action	Text (50)	Upload or Download
Transactions	Integer	Number of actions for the given time period
Company_id	Text (10)	Unique ID for a Company
Table: Fixes		
Year	Integer	Year download took place
Month	Integer	Month download took place
Downloads	Integer	Number of downloads for the given time period
Company_id	Text (10)	Unique ID for a company
Table: General_Docs		
File_id	Text (10)	Unique ID for a document
T_name	Text (100)	Name of the document
T_version	Text (10)	Version the document is for
T_local	Text (10)	Localization of the document
N_year	Integer	Year download took place
N_month	Integer	Month download took place
Company_id	Text (10)	Unique ID for a company
N_count	Integer	Number of downloads for the given time period
Table: Instant_Chat		
Year	Integer	Year view took place

Field Name	Data Type	Description
Month	Integer	Month view took place
Views	Integer	Number of views for the given time period
Company_id	Text (10)	Unique ID for a company

Table: Internal_Docs

File_id	Text (10)	Unique ID for a document
T_name	Text (100)	Name of the document
N_year	Integer	Year download took place
N_month	Integer	Month download took place
N_count	Integer	Number of downloads for the given time period

Table: Knowledge_Entries

Year	Integer	Year entry was added
Month	Integer	Month entry was added
Count	Integer	Number of entries for the given time period
Type	Text (50)	Type of entry (post, FAQ, Tip)

Table: Learning_Center

Company_id	Text (10)	Unique ID for a company
Year	Integer	Year view took place
Month	Integer	Month view took place
Page	Text (250)	Name of page
Views	Integer	Number of views for the given time period

Table: MB_accounts

Year	Integer	Year account was created
Month	Integer	Month account was created
Company_id	Text (10)	Unique ID for a company
Accounts	Integer	Number of accounts created for the given time period

Table: MB_Views

Year	Integer	Year view took place
Month	Integer	Month view took place
Company_id	Text (10)	Unique ID for a company
Views	Integer	Number of views for the given time period

Table: Misc_Docs

File_id	Text (10)	File ID for the document

Field Name	Data Type	Description
T_name	Text (100)	Name of document
N_year	Integer	Year download took place
N_month	Integer	Month download took place
N_count	Integer	Number of downloads for the given time period
Company_id	Text (10)	Unique ID for a company

Table: OLDoc_Views

Year	Integer	Year view took place
Month	Integer	Month view took place
Company_id	Text (10)	Unique ID for a company
Views	Integer	Number of views for the given time period

Table: Online_training

Course_id	Text (10)	Unique ID for a course
Course_name	Text (250)	Name of course
Month	Integer	Month course is offered
Year	Integer	Year course is offered
Reg_students	Integer	Number of students who registered
Att_students	Integer	Number of students who registered and then attended
Add_students	Integer	Number of additional students who attended but did not register
Tot_students	Integer	Number of the total students in attendance (att_students + add_students)
Version	Integer	Version class is offered for
Type	Text(50)	Type of course (Mentoring, General, Premium)

Table: Training_Article

Company_id	Text (10)	Unique ID for a company
Year	Integer	Year view took place
Month	Integer	Month view took place
Page	Text (250)	Name of page
Views	Integer	Number of views for the given time period

Table: Training_News

Company_id	Text (10)	Unique ID for a company
Year	Integer	Year view took place
Month	Integer	Month view took place
Views	Integer	Number of views for the given time period

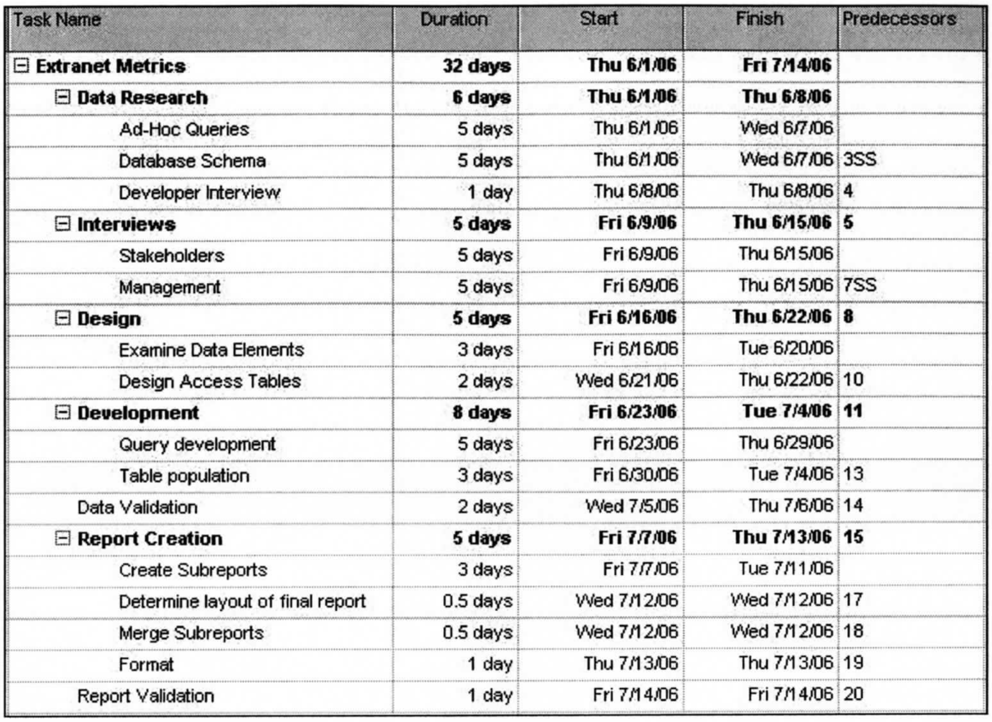
Field Name	Data Type	Description
Table: Premium_Docs		
File_id	Text (10)	Unique ID for the document
T_name	Text (100)	Name of the document
T_version	Text (10)	Version the document is for
T_local	Text (10)	Localization of the document
N_year	Integer	Year the document was downloaded
N_month	Integer	Month the document was downloaded
Company_id	Text (10)	Unique ID for a company
N_count	Integer	Number of downloads for a given time period
Table: Premium_Sessions		
Company_id	Text (10)	Unique ID for a company
Year	Integer	Year session took place
Month	Integer	Month session took place
Sessions	Integer	Number of sessions for a given time period
Table: Extranet_News		
Year	Integer	Year view took place
Month	Integer	Month view took place
Views	Integer	Number of views for a given time period
Company_id	Text (10)	Unique ID for a company
Table: Sessions		
Company_id	Text (10)	Unique ID for a company
Sessions	Integer	Number of sessions for a given time period
Year	Integer	Year session took place
Month	Integer	Month session took place
Table: Time		
Year	Integer	4-digit year
Month	Integer	2-digit month
Month_Name	Text(10)	Short name of the month (i.e. Jul)
Quarter	Text(10)	2-character description of the quarter (i.e. Q1)
Combined_Date_Name	Text(50)	Combination of Month and Year (i.e. Jul-06)
Combined_Date	Text(50)	Numeric combination of Year and Month (i.e. 06-07)
Table: Tip_FAQ_Views		
Year	Integer	Year view took place

Field Name	Data Type	Description
Month	Integer	Month view took place
Company_id	Text (10)	Unique ID for a company
Type	Text (50)	Type of entry (Tip or FAQ)
Views	Integer	Number of views for a given time period

Table: Users

Year	Integer	Year user account created
Month	Integer	Month user account created
Type	Text (20)	Type of account (Admin or Standard User)
Accounts	Integer	Number of Accounts
Company_id	Text (10)	Unique ID for a company

Appendix D: Project Activities and Timeline



Task Name	Duration	Start	Finish	Predecessors
⊟ **Extranet Metrics**	**32 days**	**Thu 6/1/06**	**Fri 7/14/06**	
⊟ **Data Research**	**6 days**	**Thu 6/1/06**	**Thu 6/8/06**	
Ad-Hoc Queries	5 days	Thu 6/1/06	Wed 6/7/06	
Database Schema	5 days	Thu 6/1/06	Wed 6/7/06	3SS
Developer Interview	1 day	Thu 6/8/06	Thu 6/8/06	4
⊟ **Interviews**	**5 days**	**Fri 6/9/06**	**Thu 6/15/06**	**5**
Stakeholders	5 days	Fri 6/9/06	Thu 6/15/06	
Management	5 days	Fri 6/9/06	Thu 6/15/06	7SS
⊟ **Design**	**5 days**	**Fri 6/16/06**	**Thu 6/22/06**	**8**
Examine Data Elements	3 days	Fri 6/16/06	Tue 6/20/06	
Design Access Tables	2 days	Wed 6/21/06	Thu 6/22/06	10
⊟ **Development**	**8 days**	**Fri 6/23/06**	**Tue 7/4/06**	**11**
Query development	5 days	Fri 6/23/06	Thu 6/29/06	
Table population	3 days	Fri 6/30/06	Tue 7/4/06	13
Data Validation	2 days	Wed 7/5/06	Thu 7/6/06	14
⊟ **Report Creation**	**5 days**	**Fri 7/7/06**	**Thu 7/13/06**	**15**
Create Subreports	3 days	Fri 7/7/06	Tue 7/11/06	
Determine layout of final report	0.5 days	Wed 7/12/06	Wed 7/12/06	17
Merge Subreports	0.5 days	Wed 7/12/06	Wed 7/12/06	18
Format	1 day	Thu 7/13/06	Thu 7/13/06	19
Report Validation	1 day	Fri 7/14/06	Fri 7/14/06	20

Figure 11: Work Breakdown Structure of the Customer Extranet Metrics Project

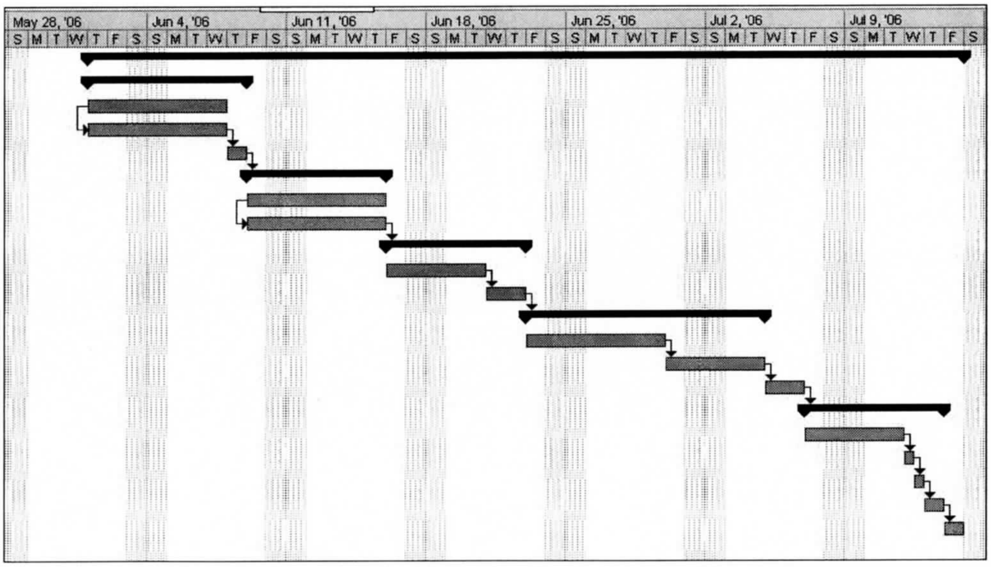


Figure 12: Gantt Chart of the Customer Extranet Metrics Project

Appendix E: Measurements and Associated Details

Measurement: General Site Statistics

Report: Provide general information on customer and user statistics

Goal: Informational Only

Program Affected: Web

Associated Queries:

```
--New companies for month
        SELECT count(company_id) as Companies,  month(date_created)as Month,
year(date_created) as Year
        FROM company_db
        where Month(date_created) =
                Case
                  WHEN month(current_timestamp)= 1 THEN 12
                  ELSE (month(current_timestamp)-1)
                END
        and Year(date_created)=
                Case
                  WHEN month(current_timestamp)= 1 THEN
year(current_timestamp)-1
                  ELSE year(current_timestamp)
                END
        and isdemo <> 1
        and date_expires > current_timestamp
        group by year(date_created), month(date_created)
        order by year(date_created), month(date_created)

--Number of active companies
        select count(company_id) as Active_Companies
        from company_db
        where date_expires >(current_timestamp)
        and isdemo <> 1

--Company with most sessions for month
        SELECT TOP 1 SumOfsessions, company_name, month, year
        FROM q_sum_sessions;
```

```
--Company with most users
        select top 1 t2.company_name, count(lid) as count
        from member_db t1, company_db t2
        where t1.company_id = t2.company_id
        and t2.date_expires > current_timestamp
        and t2.isdemo <> 1
        and t1.active <> 0
        and t1.company_id not in
('010101','010102','030721','439992','707041','594329','621361')
        group by t2.company_name
        order by count(lid)desc

--Number of new users for month
        SELECT Month, Year, account_type, Count(Users) AS CountOfUsers
        FROM q_new_users
        WHERE (((company_id) Not In
('010101','010102','030721','621361','594329','439992'))
        AND ((account_type)="Standard"))
        GROUP BY Month, Year, account_type
        HAVING (((Year)=IIf(Month(Now())=1,Year(Now())-1,(Year(Now()))))
        AND ((Month)=IIf(Month(Now())=1,12,(Month(Now()))-1));

--Total number of active users
        select count(lid) as Active_Users
        from member_db t1, company_db t2
        where t1.company_id = t2.company_id
        and t2.date_expires >(current_timestamp)
        and t2.isdemo <> 1
        and t1.active <> 0
        and t1.company_id not in
('010101','010102','030721','439992','707041','594329','621361')
        and t1.account_type = 'standard'

--Number of new admins for month
        SELECT Month, Year,.account_type, Count(Users) AS CountOfUsers
        FROM q_new_users
        WHERE (((company_id) Not In
('010101','010102','030721','621361','594329','439992'))
        AND ((account_type)="Admin"))
        GROUP BY Month, Year, account_type
        HAVING (((Year)=IIf(Month(Now())=1,Year(Now())-1,(Year(Now()))))
        AND ((Month)=IIf(Month(Now())=1,12,(Month(Now()))-1));
```

```
--Total number of active admins
        select count(lid) as Active_Admins
        from member_db t1, company_db t2
        where t1.company_id = t2.company_id
        and t2.date_expires >(current_timestamp)
        and t2.isdemo <> 1
        and t1.active <> 0
        and t1.company_id not in
('010101','010102','030721','439992','707041','594329','621361')
        and t1.account_type = 'ADMIN'


--Total number of message board users
        select count( t2.mb_alias) as No_MB
        from company_db t3, member_db t1 left outer join mb_accounts t2
        on t1.lid = t2.lid
        where t1.company_id = t3.company_id
        and t1.active <> 0
        and company_id not in ('010102','010101')
        and date_expires > current_timestamp
```

Measurement: Customer Versions

Report: Provide management with a snapshot of what software versions customers are currently using.

Goal: Informational Only

Program Affected: Web

Associated Query:

```
        select company_id, company_name, state,country,date_created, date_expires,
service_type,primary_ version, additional_ versions
        from company_db
        where isdemo<>1
```

Measurement: Customer Usage Type

Report: Provide management with a snapshot of which access types customers have

Goal: Informational Only

Program Affected: Web

Associated Query:

```
        select company_id, company_name, state,country,date_created, date_expires,
service_type,primary_ version, additional_ versions
        from company_db
        where isdemo<>1
```

Measurement: Sessions

Report: Provide a monthly summary of sessions per month.

Goal: Positive trend line; Spikes when new features are released

Program Affected: Web

Associated Query:

```
        SELECT count(DISTINCT sessionid) as sessions, month(timestamp)as Month,
year( timestamp)as Year, company_id
        FROM tracking_db
        WHERE Month(timestamp) =
                Case
                  WHEN month(current_timestamp)= 1 THEN 12
                  ELSE (month(current_timestamp)-1)
                END
        and Year(timestamp)=
                Case
                  WHEN month(current_timestamp)= 1 THEN
year(current_timestamp)-1
                  ELSE year(current_timestamp)
                END
        group by month(timestamp), year(timestamp), company_id
```

Measurement: New Users

Report: Provide a monthly summary of new accounts per month.

Goal: Positive trend line

Program Affected: Web

Associated Query:

```
        SELECT count(lid) as Users, company_id, month(account_created)as Month,
year(account_created) as Year, account_type
        FROM member_db
        where monthMonth(account_created) =
                Case
                  WHEN month(current_timestamp)= 1 THEN 12
                  ELSE (month(current_timestamp)-1)
                END
        and Year(account_created)=
                Case
                  WHEN month(current_timestamp)= 1 THEN
year(current_timestamp)-1
                  ELSE year(current_timestamp)
                END
        group by year(account_created), month(account_created), company_id,
account_type
        order by year(account_created), month(account_created), company_id
```

Measurement: Article Popularity

Report: Provide a monthly summary of views to the Learning Center (Article Library)

Goal: Positive trend line

Program Affected: Web

Associated Query:

```
        select month(timestamp) as Month, year(timestamp)as Year, company_id,
        'Articles'as page, count(timestamp)as views, page
        FROM tracking_db
        WHERE Month(timestamp) =
                Case
                  WHEN month(current_timestamp)= 1 THEN 12
                  ELSE (month(current_timestamp)-1)
                END
        and Year(timestamp)=
                Case
                  WHEN month(current_timestamp)= 1 THEN
year(current_timestamp)-1
                  ELSE year(current_timestamp)
                END
        and action='Viewed'
        and page Like '%learning.asp'
        group by month(timestamp), year(timestamp), company_id, page
```

Measurement: Message Board Views

Report: Provide a monthly summary of views to the Message Boards

Goal: Positive trend line showing growing interest in feature

Program Affected: Web

Associated Query:

```
        select month(timestamp)as Month, year(timestamp)as
Year,company_id,'MB_View'as page,count(timestamp)as views
        from tracking_db
        where action='MB Topic Review'
        and Month(timestamp) =
                Case
                  WHEN month(current_timestamp)= 1 THEN 12
                  ELSE (month(current_timestamp)-1)
                END
        and Year(timestamp)=
                Case
                  WHEN month(current_timestamp)= 1 THEN
year(current_timestamp)-1
                  ELSE year(current_timestamp)
                END
        group by month(timestamp), year(timestamp),company_id
```

Measurement: Knowledge Entries

Report: Provide a monthly summary of additions to the Knowledge Base

Goal: Sharp increase (positive trend line) in activity showing growing participation from the user community

Program Affected: Web

Associated Queries:

```
        select month(tip_date)as Month, year(tip_date)as Year, 'Tip' as Type,
count(tip_id)as Entries
        from tip_db
        WHERE Month(tip_date) =
                Case
                  WHEN month(current_timestamp)= 1 THEN 12
                  ELSE (month(current_timestamp)-1)
                END
        and Year(tip_date)=
                Case
                  WHEN month(current_timestamp)= 1 THEN
year(current_timestamp)-1
                  ELSE year(current_timestamp)
                END
        group by month(tip_date), year(tip_date)

        union

        select month(date_added)as Month, year(date_added)as Year, 'FAQ' as Type,
count(faq_id)as Entries
        from faq_db
        WHERE Month(date_added) =
                Case
                  WHEN month(current_timestamp)= 1 THEN 12
                  ELSE (month(current_timestamp)-1)
                END
        and Year(date_added)=
                Case
                  WHEN month(current_timestamp)= 1 THEN
year(current_timestamp)-1
                  ELSE year(current_timestamp)
                END
        group by month(date_added), year(date_added)

        union
        select month(post_created_date)as Month, year(post_created_date)as Year,
'MB_Post' as Type, count(post_id)as Entries
        from mb_posts
        where Month(post_created_date) =
                Case
                  WHEN month(current_timestamp)= 1 THEN 12
                  ELSE (month(current_timestamp)-1)
                END
        and Year(post_created_date)=
                Case
                  WHEN month(current_timestamp)= 1 THEN
year(current_timestamp)-1
                  ELSE year(current_timestamp)
                END
        group by month(post_created_date), year(post_created_date)
```

Measurement: Message Board Registrations

Report: Provide a monthly summary new message board accounts

Goal: Positive trend line showing users are growing in participation

Program Affected: Web

Associated Query:

```
        SELECT count(action_detail) as Accounts, year(timestamp) as Year,
month(timestamp) as Month, company_id
        FROM  tracking_db
        WHERE action='MB Account Created'
        and Month(timestamp) =
                Case
                  WHEN month(current_timestamp)= 1 THEN 12
                  ELSE (month(current_timestamp)-1)
                END
        and Year(timestamp)=
                Case
                  WHEN month(current_timestamp)= 1 THEN
year(current_timestamp)-1
                  ELSE year(current_timestamp)
                END
        group by year(timestamp), month(timestamp), company_id
        order by year(timestamp), month(timestamp)
```

Measurement: Tips and FAQ Views

Report: Provide a monthly summary of views to the Tips and Hints and FAQs

Goal: Positive trend line showing users are enabling themselves to search for answers

Program Affected: Web

Associated Queries:

```
        select month(timestamp)as Month, year(timestamp) as
Year,company_id,'Tip'as page,count(timestamp)as views, page
        from tracking_db
        where page like '%tips%'
        and  action = 'opened'
        and Month(timestamp) =
                Case
                  WHEN month(current_timestamp)= 1 THEN 12
                  ELSE (month(current_timestamp)-1)
                END
        and Year(timestamp)=
                Case
                  WHEN month(current_timestamp)= 1 THEN
year(current_timestamp)-1
                  ELSE year(current_timestamp)
                END
        group by month(timestamp), year(timestamp),company_id, page
        UNION
        select month(timestamp)as Month, year(timestamp)as Year, company_id,
Tip'as page, count(timestamp)as views, page
        from tracking_db
        where page like '%faq%'
        and page not like '%usergroup%'
        and  action = 'opened'
        and Month(timestamp) =
                Case
                  WHEN month(current_timestamp)= 1 THEN 12
                  ELSE (month(current_timestamp)-1)
                END
        and Year(timestamp)=
                Case
                  WHEN month(current_timestamp)= 1 THEN
year(current_timestamp)-1
                  ELSE year(current_timestamp)
                END
        group by month(timestamp), year(timestamp), company_id, page
```

Measurement: Miscellaneous Document Downloads

Report: Provide a monthly summary of miscellaneous downloads

Goal: Positive trend line

Program Affected: Web

Associated Query:

```
select t2.Month, t2.Year,  t1.filename, t1.file_id, count( t2.timestamp) as Downloads,
'Misc' as Type, t2.company_id
from misc_files_db t1,
        (select right(action_detail, 6) as docid,month(timestamp)as Month,
        year(timestamp) as Year, timestamp, company_id
        from tracking_db
where Month(timestamp) =
                Case
                  WHEN month(current_timestamp)= 1 THEN 12
                 ELSE (month(current_timestamp)-1)
                END
and Year(timestamp)=
                Case
                  WHEN month(current_timestamp)= 1
            THEN year(current_timestamp)-1
                  ELSE year(current_timestamp)
                END
        and action = 'download'
        and action_detail like '%misc_files_db%') t2
where t1.file_id = t2.docid
group by t2.month, t2.year, t1.filename, t1.file_id,  t2.company_id
order by t2.year, t2.month
```

Measurement: Extranet Newsletter

Report: Provide a monthly summary of views of the extranet newsletter

Goal: Positive trend line

Program Affected: Web

Associated Query:

```
select month(timestamp)as Month, year(timestamp) as Year,company_id as
Company_ID,'News' as Page,
count(timestamp)as Views
from tracking_db
where page like '%en/newsletter%'
and  action = 'opened'
Month(timestamp) =
        Case
          WHEN month(current_timestamp)= 1 THEN 12
          ELSE (month(current_timestamp)-1)
        END
and Year(timestamp)=
        Case
          WHEN month(current_timestamp)= 1 THEN year(current_timestamp)-1
          ELSE year(current_timestamp)
        END
group by month(timestamp), year(timestamp),company_id
order by month(timestamp), year(timestamp)
```

Measurement: General Training Statistics

Report: Provide general information on training popularity and general training statistics

Goal: Informational Only

Program Affected: Training

Associated Queries:

```
--Most registered
        SELECT TOP 1 year, month,  course_name, Sum(reg_students) AS
        SumOfreg_students, version, short_name
        FROM t_online_training
        GROUP BY year, month, course_name, version, short_name
        HAVING (((year)=IIf(Month(Now())=1,Year(Now())-1,(Year(Now()))))
        AND ((month)=IIf(Month(Now())=1,12,(Month(Now()))-1))
        ORDER BY Sum(reg_students) DESC;

--Least registered
        SELECT TOP 1 year, month,  course_name, Sum(reg_students) AS
        SumOfreg_students, version, short_name
        FROM t_online_training
        GROUP BY year, month, course_name, version, short_name
        HAVING (((year)=IIf(Month(Now())=1,Year(Now())-1,(Year(Now()))))
        AND ((month)=IIf(Month(Now())=1,12,(Month(Now()))-1))
        AND ((Sum(reg_students))<>0))
        ORDER BY Sum(reg_students);

--Most attended
        SELECT TOP 1 year, month, course_name, Sum(tot_students) AS
        SumOftot_students, version, short_name
        FROM t_online_training
        GROUP year, month, course_name, version, short_name
        HAVING (((year)=IIf(Month(Now())=1,Year(Now())-1,(Year(Now()))))
        AND ((month)=IIf(Month(Now())=1,12,(Month(Now()))-1))
        ORDER BY Sum(tot_students) DESC;
```

```
--Least attended
        SELECT TOP 1 year,  month, course_name, short_name, Sum(tot_students) AS
        SumOftot_students, version
        FROM t_online_training
        GROUP BY year, month, course_name, short_name, version
        HAVING (((year)=IIf(Month(Now())=1,Year(Now())-1,(Year(Now()))))
        AND ((month)=IIf(Month(Now())=1,12,(Month(Now()))-1))
        AND ((Sum(tot_students)) Is Not Null))
        ORDER BY Sum(tot_students);

--Number of classes for the month
        SELECT year, month, Count(course_id) AS CountOfcourse_id
        FROM t_online_training
        GROUP BY year, month
        HAVING (((year)=IIf(Month(Now())=1,Year(Now())-1,(Year(Now()))))
        AND ((month)=IIf(Month(Now())=1,12,(Month(Now()))-1))

--Number of premium training customers
        SELECT Count(company_id) AS CountOfcompany_id
        FROM t_company
        WHERE (((product_type)="pla" Or (product_type)="pret")
        AND ((company_id) Not In ('010101','010102','030721','621361','707041'))
        AND ((date_expires)>Now()));
```

Measurement: Training Roster and Attendance

Report: Provide high-level, monthly reports on registrations vs. attendance

Goal: Increase registrations and reduce no-shows to class

Program Affected: Training

Associated Query:

```
SELECT t3.course_id, t3.title as Course_Name, left(title,(charindex(':',title))) as
Short_name, month(t3.course_date) as Month, year(t3.course_date) as
Year,count(t2.email) as Reg_Students, count(t2.attended) as Att_Students,
t3.additional_attendees as Add_Students, (count(t2.attended)+ t3.additional_attendees)
as Total_Students,t3.instructor as Version, t3.type
FROM course_roster t2 right outer join course_catalog t3
ON t2.course_id = t3.course_id
WHERE t3. type In ('type1','type2','type3')
and Month(timestamp) =
        Case
          WHEN month(current_timestamp)= 1 THEN 12
          ELSE (month(current_timestamp)-1)
        END
and Year(timestamp)=
        Case
          WHEN month(current_timestamp)= 1 THEN year(current_timestamp)-1
          ELSE year(current_timestamp)
        END
group by year(t3.course_date), month(t3.course_date),t3.course_id, t3.title,
t3.instructor, t3.additional_attendees, t3. type
order by year(t3.course_date), month(t3.course_date)
```

Measurement: Documentation General Statistics

Report: Provide general information on document popularity and general statistics

Goal: Informational only

Program Affected: Documentation

Associated Queries:

```
--Sum of General Documents
        SELECT t_name, Sum(n_count) AS SumOfn_count, n_month, n_year
        FROM t_general_docs
        WHERE (((company_id) Not In ('123345', '123456')))
        GROUP BY t_name, n_month, n_year
        HAVING (((n_year)=IIf(Month(Now())=1,Year(Now())-1,(Year(Now()))))
        AND ((n_month)=IIf(Month(Now())=1,12,(Month(Now()))-1))
        ORDER BY Sum(n_count) DESC;

--Top General Document
        SELECT TOP 1 SumOfn_count, t_name, n_month, n_year
        FROM q_sum_gen_docs;

--Sum of Premium Documents
        SELECT t_name, Sum(n_count) AS SumOfn_count, n_month, n_year
        FROM t_premium_docs
        WHERE (((company_id) Not In ('123345', '123456')))
        GROUP BY t_name, n_month, n_year
        HAVING (((n_year)=IIf(Month(Now())=1,Year(Now())-1,(Year(Now()))))
        AND ((n_month)=IIf(Month(Now())=1,12,(Month(Now()))-1))
        ORDER BY Sum(n_count) DESC;

--Top Premium Document
        SELECT TOP 1 SumOfn_count, t_name,  n_month, n_year
        FROM q_sum_premium_docs;

--Sum of Internal Documents
        SELECT t_name, Sum(n_count) AS SumOfn_count, n_month, n_year
        FROM t_internal_docs
        GROUP BY t_name, n_month, n_year
        HAVING (((n_year)=IIf(Month(Now())=1,Year(Now())-1,(Year(Now()))))
        AND ((n_month)=IIf(Month(Now())=1,12,(Month(Now()))-1))
        ORDER BY Sum( n_count) DESC;

--Top Internal Document
        SELECT TOP 1 SumOfn_count, t_name, n_month, n_year
        FROM q_sum_int_docs;
```

Measurement: Documentation Downloads

Report: Provide 3 reports monthly on the download counts for general documents, internal documents as well as premium documents.

Goal: Increased downloads – positive trend line

Program Affected: Documentation

Associated Queries:

```
--General Documentation
        select t2.Month, t2.Year,  t1.filename, t1.file_id, count( t2.timestamp) as
Downloads, 'General' as Type, t2.company_id, t1.version, t1.localization
        from docu_db t1,
                (select right(action_detail, 6) as docid,month(timestamp)as Month,
        year(timestamp) as Year, timestamp, company_id
                from tracking_db
        where Month(timestamp) =
                        Case
                          WHEN month(current_timestamp)= 1 THEN 12
                          ELSE (month(current_timestamp)-1)
                        END
        and Year(timestamp)=
                        Case
                          WHEN month(current_timestamp)= 1
                    THEN year(current_timestamp)-1
                          ELSE year(current_timestamp)
                        END
                and action = 'download'
                and action_detail like '%docu_db%') t2
        where t1.file_id = t2.docid
        and t1.file_location not like '%Internal%'
        and t1.file_location not like '%Premium%'
        group by t2.month, t2.year, t1.filename, t1.file_id,  t2.company_id,  t1.version,
t1.localization

--Internal Documentation
        select t2.Month, t2.Year,  t1.filename, t1.file_id, count( t2.timestamp) as
Downloads, 'Internal' as Type
        from docu_db t1,
                (select right(action_detail, 6) as docid,month(timestamp)as Month,
        year(timestamp) as Year, timestamp
                from tracking_db
        where Month(timestamp) =
                        Case
```

```
                                  WHEN month(current_timestamp)= 1 THEN 12
                                  ELSE (month(current_timestamp)-1)
                                END
           and Year(timestamp)=
                                Case
                                  WHEN month(current_timestamp)= 1
                         THEN year(current_timestamp)-1
                                  ELSE year(current_timestamp)
                                END
                   and action = 'download'
                   and action_detail like '%docu_db%') t2
           where t1.file_id = t2.docid
           and t1.file_location like '%Internal%'
           group by t2.month, t2.year, t1.filename, t1.file_id

--Premium Documentation
           select t2.Month, t2.Year,  t1.filename, t1.file_id, count( t2.timestamp) as
Downloads, 'Premium' as Type, t2.company_id, t1.version, t1.localization
           from docu_db t1,
                   (select right(action_detail, 6) as docid,month(timestamp)as Month,
           year(timestamp) as Year, timestamp, company_id
                   from tracking_db
           where Month(timestamp) =
                                Case
                                  WHEN month(current_timestamp)= 1 THEN 12
                                  ELSE (month(current_timestamp)-1)
                                END
           and Year(timestamp)=
                                Case
                                  WHEN month(current_timestamp)= 1
                         THEN year(current_timestamp)-1
                                  ELSE year(current_timestamp)
                                END
           and action = 'download'
           and action_detail like '%docu_db%') t2
           where t1.file_id = t2.docid
           and t1.file_location like '%Premium%'
           and t1.file_location not like '%Internal%'
           group by t2.month, t2.year, t1.filename, t1.file_id, t2.company_id,  t1.version,
t1.localization
```

Measurement: Online Document Views

Report: Provide a monthly summary of views of the online documentation

Goal: Increased downloads – positive trend line

Program Affected: Documentation

Associated Query:

```
        select month(timestamp)as Month, year(timestamp)as Year, company_id,
        'OL_Docs'as page, count(timestamp)as views, page
        FROM tracking_db
        WHERE month Month(timestamp) =
                Case
                  WHEN month(current_timestamp)= 1 THEN 12
                  ELSE (month(current_timestamp)-1)
                END
        and Year(timestamp)=
                Case
                  WHEN month(current_timestamp)= 1 THEN
year(current_timestamp)-1
                  ELSE year(current_timestamp)
                END
        and page Like '%robohelp%'
        group by month(timestamp), year(timestamp), company_id, page
```

Measurement: File Basket Usage

Report: Provide a monthly summary of views of the online documentation

Goal: Positive trend line showing the increased usage of the secure file transfer

Program Affected: Support

Associated Query:

```
        select month(timestamp)as Month, year(timestamp)as
Year,company_id,count(timestamp)as transactions, action
        FROM tracking_db
        WHERE Month(timestamp) =
                Case
                  WHEN month(current_timestamp)= 1 THEN 12
                  ELSE (month(current_timestamp)-1)
                END
        and Year(timestamp)=
                Case
                  WHEN month(current_timestamp)= 1 THEN
year(current_timestamp)-1
                  ELSE year(current_timestamp)
                END
        and ((action_detail Like 'filebasket%' and action='download') OR (action ='file
uploaded'))
        GROUP BY month(timestamp), year(timestamp),company_id, action
```

Measurement: Support Downloads

Report: Provide a monthly summary of downloads of fixes and service packs

Goal: Positive trend line

Program Affected: Support

Associated Query:

```
        select month(timestamp)as Month, year(timestamp)as Year, company_id,
count(timestamp)as downloads
        FROM tracking_db
        WHERE Month(timestamp) =
                Case
                  WHEN month(current_timestamp)= 1 THEN 12
                  ELSE (month(current_timestamp)-1)
                END
        and Year(timestamp)=
                Case
                  WHEN month(current_timestamp)= 1 THEN
year(current_timestamp)-1
                  ELSE year(current_timestamp)
                END
        and action='download'
        and action_detail Like 'fix_db%'
        GROUP BY month(timestamp), year(timestamp), company_id
        ORDER BY year(timestamp), month(timestamp)
```

Measurement: Instant Chat Views

Report: Provide a monthly summary of views of Instant Chat

Goal: Positive trend line

Program Affected: Support

Associated Query:

```
        select month(timestamp)as Month, year(timestamp)as Year, company_id,
        'Live_Help'as page,count(timestamp)as views, page
        FROM tracking_db
        WHERE Month(timestamp) =
                Case
                  WHEN month(current_timestamp)= 1 THEN 12
                  ELSE (month(current_timestamp)-1)
                END
        and Year(timestamp)=
                Case
                  WHEN month(current_timestamp)= 1 THEN
year(current_timestamp)-1
                  ELSE year(current_timestamp)
                END
        and page Like '%instantservice%'
        group by month(timestamp), year(timestamp),company_id, page
        order by year(timestamp), month(timestamp)
```

Measurement: Premium Downloads

Report: Provide a monthly summary of downloads from the Premium area

Goal: Positive trend line

Program Affected: Premium Services

Associated Query:

```
        select t2.Month, t2.Year,  t1.filename, t1.file_id, count( t2.timestamp) as
Downloads, 'Premium' as Type, t2.company_id, t1.version, t1.localization
        from docu_db t1,
                (select right(action_detail, 6) as docid,month(timestamp)as Month,
        year(timestamp) as Year, timestamp, company_id
                from tracking_db
        where Month(timestamp) =
                Case
                  WHEN month(current_timestamp)= 1 THEN 12
                  ELSE (month(current_timestamp)-1)
                END
        and Year(timestamp)=
                Case
                  WHEN month(current_timestamp)= 1 THEN
year(current_timestamp)-1
                  ELSE year(current_timestamp)
                END
                and action = 'download'
                and action_detail like '%docu_db%') t2
        where t1.file_id = t2.docid
        and t1.file_location like '%Premium%'
        and t1.file_location not like '%Internal%'
        group by t2.month, t2.year, t1.filename, t1.file_id, t2.company_id,  t1.version,
t1.localization
```

Measurement: Premium Sessions

Report: Provide a monthly summary of sessions to the Premium area

Goal: Positive trend line showing increased traffic to premium areas

Program Affected: Premium Services

Associated Query:

```
        SELECT count(DISTINCT sessionid) as sessions, month(timestamp)as Month,
year(timestamp)as Year, company_id
        FROM tracking_db
        WHERE Month(timestamp) =
                Case
                  WHEN month(current_timestamp)= 1 THEN 12
                  ELSE (month(current_timestamp)-1)
                END
        and Year(timestamp)=
                Case
                  WHEN month(current_timestamp)= 1 THEN
year(current_timestamp)-1
                  ELSE year(current_timestamp)
                END
        And page Like '%Premium%'
        And page Not Like ('/en/premium/about.asp')
        group by month(timestamp), year(timestamp), company_id
        order by year(timestamp), month(timestamp)
```