The Report Committee for Roberto Mario Vasquez
Certifies that this is the approved version of the following report:

A Project Plan for Improving the Performance Measurement Process - A Usability Case Study

APPROVED BY
SUPERVISING COMMITTEE:

Co-supervisors:

Herb Krasner

Dewayne E. Perry

A Project Plan for Improving the Performance Measurement Process - A Usability Case Study

by

Roberto Mario Vasquez, B.S.E.E

Report

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Master of Science in Engineering

The University of Texas at Austin

December 2010

Dedication

To my family and best friend, for their patience and support.

Acknowledgements

Specials thanks to Prof. Herb Krasner for his lectures, his patience, and the guidance he offered for my topic selection and throughout the writing of this report. I want to also thank Dr. Dewayne E. Perry for his time and comments in reviewing this document. Additionally, I want to acknowledge the support and cooperation of my manager and co-workers at INTEL Corporation. I also want to thank my editors.

November 16, 2010

Abstract

A Project Plan for Improving the Performance Measurement Process - A Usability Case Study

Roberto Mario Vasquez, M.S.E.

The University of Texas at Austin, 2010

Co-supervisors: Herb Krasner and Dewayne E. Perry

Many good software practices are often discarded because of the syndrome "there is not enough time, do it later", or "it is in our head and there is no time to write it down." As a consequence, projects are late, time frames to complete software modules are unrealistic and miscalculated, and traceability to required documents and their respective stakeholders do not exist. It is not until the release of the application that it is determined the functionalities do not meet the expectations of the end users and stakeholders. The effect of this can be detrimental to the individuals of the development team and the organization. Associating measurement and metrics to internal software processes and tasks, followed by analysis and continual evaluation, are key elements to close many of the repeated gaps in the life cycle of software engineering, regardless of the software methodology.

This report presents a usability case study of a customized application during its development. The application contains internal indicator modules for performance measurement processes captured at the level of a Request System application within a horizontal organizational group. The main goals for the usability surveys and case study were

(1st) to identify, define and evaluate the current gaps in the system and

(2nd) find new approaches and strategies with the intent to move the project in the right direction.

Gaps identified throughout the development process are included as indicators for process improvement. The result of the usability case study creates new goals and gives clear direction to the project. Goal-driven measurements and the creation of a new centralized collaborative web system for communication with other teams are parts of the solution. The processes and techniques may provide benefits to companies interested in applying similar tactics to improve their own software project processes.

Table of Contents

List of Tables x

List of Figures xi

INTRODUCTION 1

Motivation 2

1. LITERATURE - MEASUREMENT AND METRICS 4

Software Measurement 4

The Software Quality Framework 5

IEEE Std. 1061-1998 5

Goal-Question-Metric (GQM) 5

ISO/IEC 9126 6

2. OUR APPROACH 8

Our Usability Goal 8

3. THE USABILITY CASE STUDY IMPLEMENTATION 9

Background 9

The Validation and Testing Core Group 9

The Request - Indicator System 10

Management Perspective and Motivation 11

The Gaps 12

The Development Environment Gap 12

The Planning and Execution Process Phase 14

Measurement and Metrics Discussion with Team 15

Usability Case Study Approval from Management 15

The Pilot Program 17

Data Capturing of The Request System 17

The Surveys 18
Choosing the Users for the Usability Surveys Case Study 18
Creation of the Surveys 18
The Questions 19
Survey #1 - Awareness 21
The results of Survey #1 22
Comments and Feedback 23
Possible Improvement Action Plan 23
Email Notification after Survey #1 26
Usability Survey #2 – System Embracement 27
The Results of Survey #2 28
Analysis for Survey #2 28
Comments and Feedback 30
Possible Improvement Action Plan 31
Email Notification After Survey #2 32
Survey #3 - Indicators Evaluation - Usability 33
The Results of Survey #3 35
Comments and Feedback 37
Possible Improvement Action Plan 38
Email Notification After Survey #3 38
Development of a Collaboration Team Site 39
Results of the Request System Pilot Program 40
4. CONCLUSION 42
BIBLIOGRAPHY 46
Vita 49

List of Tables

Table 1: Improvement Action Plan as Results of Survey #1. 25

Table 2: Summary of Survey #2 - System Embracement 29

Table 3: Survey #2 - Improvement Action Items.....31

Table 4: Number of Total Requests Created and Received per Month by Users 40

List of Figures

Figure 1: Software Quality - ISO 9126 (21).....6

Figure 2: Survey #1 - Awareness and Notifications Questions.....21

Figure 3: Survey #1 - Awareness - Notifications Questions Results 22

Figure 4: Survey #2 - Usability System Embracement - Request.....27

Figure 5: Survey #2 - System Embracement Results 28

Figure 6: Survey #3 - Usability of Indicators.....33

Figure 7: Man Hours Distribution/Owners.....35

Figure 8: Survey #3 - Results - Usability of Indicators.....36

Figure 9: The Request System Collaboration Web Site - Surveys.....39

INTRODUCTION

We often think that after years of trying to incorporate software methodologies into our projects we will be able to successfully complete a project under budget, on time, and bug-free. But, the reality is that reaching that goal is rarely the case.

In the last decade, many best practices for software development have evolved (1), (2), (3), and new software methodologies have appeared that have been evolving for several decades (4). Most of those methodologies still have roots based on ideologies that are not new concepts to many software engineers, but the new terminologies and framework environments keep them fresh and appealing to the new generation of software engineers. The facts are that we have great principals that are behind many of those methodologies with their own pros and cons, and we know that no one methodology can be the solution to all projects. Some papers found in field literature have shown that no matter what methodology software process is followed; projects are still very disappointing in terms of cost, schedule, and quality (5). A single bad decision and the environmental framework that surrounds it can affect the outcome of the entire team regardless of the software methodology being applied. Therefore, a methodology can only be as good as the knowledge and skill sets of the people who implement it. Further, communication is a key factor among the engineers, designers and users. The human factor is still something that cannot be ignored. Observations of different businesses and teams demonstrate that communication breakdowns shown in a study done by Krasner, Curtis and Iscoe in 1987 (6) are still very prevalent in many organizations.

In addition to the various software methodologies in the last decade, there has been a lot of progress in design programs related to software quality, as well as process

improvement techniques and metrics initiatives (7). Among them, the ISO 15939, CMMI, and ISBSG provide measurement process standards (8), (9). The integration of those models (8) appears to produce valuable software engineering and management disciplines that could be a turn-key solution to close the gaps in our own project and software methodologies.

Motivation

Literature and observations from within several companies show that the implementation of measurement programs can be very challenging, and even controversial, because of the lack of knowledge of software measurement metrics.

For the Software Systems Measurement and Metrics course in the summer of 2010, out of a survey sent to thirteen colleagues asking what software measurement metrics were being used in their organization, only two provided useful information. For the Software Engineering Life Cycle course in the summer of 2009, a similar survey was sent to a different set of fourteen individuals including software developers, software consultants, project managers, and directors. Only two responses were received. In an effort to determine why the other people did not respond, it was learned the 'no response' results were mainly because software metrics are not being used in their projects. For most of them, the words measurement and metric were very unclear. Some said that measurement metrics were only related to the cost of their project, the schedule deadline given by their managers, and the client's satisfaction.

From discussion of software measurement metrics with some managers, it is understood that performance metrics are a key factor to provide the needed feedback to improve the business and development processes within projects. Unfortunately, when

building this type of performance measurement program, most program developers frequently fail to provide the expected results because of the way the programs are implemented, or due to the fact that these programs are often misinterpreted by management (10). The hope for us, as software engineers, is that management can properly understand measurement metrics, and other team members can apply them to close the gaps in our own software methodologies processes. The greatest challenge is how to identify and best build the infrastructure and supportive framework environment to implement the proper measurement program, taking into account that human decision factors are always present. It is this rationale that is the root of the motivation for this report.

The usability case study presented in this report is a product of observed gaps from a software consultant perspective from within the current organization. This perspective is in relation to a new customized request system application with pre-built indicator modules that capture performance measurement metrics data for evaluation. This case study is related to the usability measurement of the request system and the indicators built within the application. The evaluation of the usability is executed with surveys and data collection methods over three consecutive months during the request system indicators pilot program. In the next sections, common terminologies are defined and reasons given for measurement and metrics as they are found in literature. The process and execution of the usability survey case study and its data analysis follows. The results will be presented in the last section followed by a conclusion of the effectiveness and value of the case study.

1. LITERATURE - MEASUREMENT AND METRICS

In the life cycle of software engineering we often find ourselves wondering about the status of a project, how are we doing, the resources we will need to complete the project in order to meet the allocated budget, the performance and productivity of the team members, the risks of our processes, and how effective the implemented process is. To provide proper answers to these questions, it is necessary to have understanding as well as control over the process. In addition, we must be able to predict and adjust the outcomes.

Software Measurement

Literature shows that the software measurement process has different meanings for different people. Entire books and many articles are found dedicated to defining the software measurement process (3), (11), (12), (13) which can be taken from many different points of view.

We can deduce that measurements depend on stakeholders' needs, their goals of what to control, and how to make improvements. Thus measurements are important not only for the numbers, but also for giving us access to the information we need to understand, manage, and improve our business processes. Swanson (14) presents us with a matrix of performance levels and performance variables that occur at the organization, process, or individual level. The performance variables are the Mission Goal, System Design, Capacity, Motivation and Expertise. Applying the right strategies and creating the proper environment within the organization are important to effectively implementing

a performance measurement framework. The survey processes presented in this paper are influenced by process improvement techniques from a software quality framework.

The Software Quality Framework

There are many standard frameworks in the software industry that have evolved in order to evaluate software quality. As a baseline for this report we have chosen three frameworks to evaluate software quality measurement:

- The IEEE Standard 1061-1998 (15)
- The Goal-Question-Metric (GQM) (16), (17)
- ISO/IEC 9126

IEEE STD. 1061-1998

The IEEE Std. 1061-1998 presents us with a methodology that is used throughout each of the phases of the software life cycle. It establishes the quality of requirements, and presents a framework for "identifying, implementing, analyzing, and validating the process and product software quality metrics of the software system". The 1061 is one of the first IEEE issued standards that deals with quality metrics (18) and its execution is iterative (19).

GOAL-QUESTION-METRIC (GQM)

The Goal-Question-Metric (GQM) is another paradigm that helps define measurements at the level of the software project, processes, and products. The GQM paradigm of Basili presents us with the notion that measurements must be goal-oriented,

and that the primary question is not “What measurement should I use?” but rather “What do I need to improve?” (3). The GQM paradigm consists of three levels: Conceptual, Operational and Quantitative. The GQM uses a top down design approach. The approach is defined in three hierarchical layers: the goal, the questions, and the metrics. A detailed definition and implementations can be found in Basili and Rombach (18), (17).

ISO/IEC 9126

The ISO/IEC 9126 is one of the software quality measurement standard frameworks by the International Organization for Standardization (ISO). Its approach represents the view in terms of the customer, users, supplier, managers, and developers. The characteristics involved as the quality model are categorized in terms of Functionality, Reliability, Usability, Efficiency, Maintainability, and Portability (20). Figure 1 shows the ISO/IEC 9126 with its associated sub-characteristics.

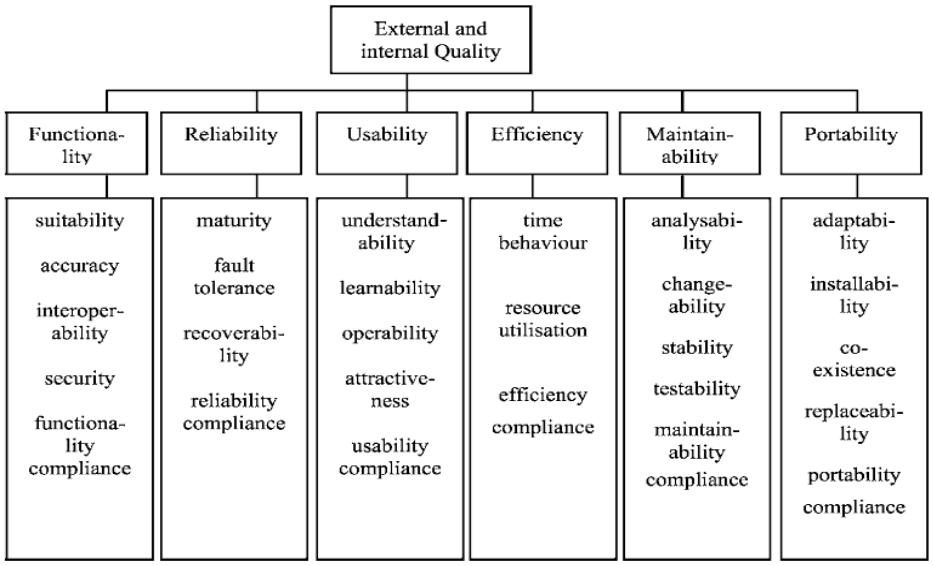


Figure 1: Software Quality - ISO 9126 (21)

Literature shows that there are many definitions for software quality in relation to usability (22), (23), (24), (21). For the goals of this report, we are interested in the usability characteristic of the model. This model uses some criteria with metrics that are relevant to some of the measurements we present for our survey, such as effectiveness and satisfaction factors, where measurement for percentage of user's favorable and unfavorable comments are captured. Nevertheless, the vast amount of variation of usability measurements is an indication that usability standards are still evolving, and our measurements metrics are best captured from the needs of what is relevant to the organization.

2. OUR APPROACH

The survey evaluation presented in this report explored the usability of an internal development project called the Request Indicator (RI) system application, which was developed within an internal organization of a large corporation and run as a pilot program. The approach used surveys using a Goal-Question Metric (GQM) approach and the characteristics of a usability model.

Our Usability Goal

For this case study, usability was defined from both an internal and external view. For the internal view we wanted to know whether or not the RI system was convenient and practical to use, ready for final deployment, and had the appropriate level of acceptance among the end users. From an external view, we wanted to measure the satisfaction characteristics from the end user perspective. We used surveys to measure the "percentage of user's favorable and unfavorable comments" (21), and we looked at data collected in the RI system to observe the frequency of use. We also evaluated the user's awareness in relation to the pilot program and whether or not it met their performance criteria.

3. THE USABILITY CASE STUDY IMPLEMENTATION

For the purpose of this paper, the user names of any participants, as well as the internal group of the organization, were changed to Validation and Testing Core Group (VTCG) and Product Validation Engineering (PVE) Group in order to protect the privacy of the internal groups and the individuals within.

Background

During a GQM exercise in the summer of 2010 with some members of the VTCG used measurement performance indicators based on specific goals derived from the perspective of the project manager. The VTCG members were presented an approach to identify and validate existing performance metrics.

Management of the VTCG showed a great interest in pursuing the validation and collection of metrics for the RI system. Implementation for the RI system already began, and it was deployed in the last week of June of 2010 as a pilot program.

THE VALIDATION AND TESTING CORE GROUP

To better understand what the VTCG did for the rest of the company, it was important to understand the structure of operation for this group in relation to its parent group. We will refer to the parent of the VTCG as the Product Validation Engineering Group (PVEG). The PVEG is an organization unit begun four years ago and grew from a few individuals to several hundred. The PVEG is part of the INTEL Corporation, a large hierarchical organization and one of the largest technology companies in the world. Internal group units within the company operate in different organizational structures.

The VTCG operates in a horizontal structure under the umbrella of the PVEG, and consists of about eight primary team members. Some of those team members are also team leads for their own sub-teams. The VTCG manages and controls the business operation for Purchasing, Receiving, Inventory, and other infrastructure operations in one of the labs. The lab, in its simple definition, is one of a few core world-centers where validation and testing for some of the company's products takes place. Each of the aforementioned VTCG business operations is also referred to as a "group unit". In addition to these group units, three members of the VTCG are part of the Application Development Group Unit, which build application tools needed for the other operations of the group units within the entire VTCG team.

THE REQUEST - INDICATOR SYSTEM

The operation of daily activities between the group units require request-tickets that originate from one group unit and are sent to another group unit within the VTCG or from other internal organization groups of the PVEG. Keeping track of the time it takes to fulfill a request, as well as the status, stages, and other tasks associated with that request, has been a challenge to manage because of the dynamic framework of operation. For this reason a new application system was added to the list of applications developed by the Application Group Unit of VTCG, otherwise known as the RI systems. The main goal of this application is to track the level of performance on the execution of requests by group units and the members of these groups.

The development of the RI system started sometime in early in 2010. The system consists of two main framework components, the request form module and the indicators, thus the name of Request Indicators system. Currently, all members of the groups can

make requests to other team members or create requests for themselves. The main goal, from a management perspective, is that users can track their own time, check the number of requests assigned, and management can see the time it takes to complete a request as well as the total load of requests assigned to an individual or group unit. The indicators are graphical drill down charts that management refers to as performance indicators. A description and figure of those indicators are found in the Survey #3 section.

Management perceives the indicators would allow them to easily view and analyze different types of information to measure the performance of the team. Some of that information includes the man-hours as well as the number of requests made by both the team and the members of each group unit. We can think of the RI system as an internal customized ticket-tracking system with performance indicators tailored to support all the group units of the VTCG.

MANAGEMENT PERSPECTIVE AND MOTIVATION

For the management, the main goals of this system are to improve the performance measurement process using a RI system with its own customized internal indicator modules. Showing visibility and having a method to measure productivity is very important to the VTCG, especially when the company goes through many re-organizations as part of the effects of the global economy. The increase of offshore outsourcing can highly impact local resource allocation, budgets and local jobs (25). Therefore, one way of gaining visibility, according to the VTCG manager, is to be able to produce the metrics needed for showing, controlling, and improving performance and resource utilization at the level of the current request process.

The Gaps

The idea of building a performance measurement RI system was the right thing to do from a project management perspective. Making this application useful to the staff as a performance indicator was a challenge because of the fast-paced environment and the different dynamic responsibilities of each group. This application introduced a system with metrics that had never been used by any of the group units and staff. How to build the correct performance tool to obtain the buy-in of the other members, and to know that the application team is building the right tool based on the needs of each group unit rather than "a one version fits all" was important to know during the pilot program. Based on that rationale, a usability case study was proposed and accepted by other members of the team, including both management and the RI system development team.

THE DEVELOPMENT ENVIRONMENT GAP

The VTCG operates in a very fast-paced working environment. For the application development team, each member works on different applications and is expected to gather requirements, as well as design, develop, deploy, and maintain the application. This can be a problem for the developer when there is a short time frame to produce large, complex applications without spending enough time on the requirements, analysis, and design of the product. Others problems within the team, or gaps that develop, include:

- Lack of centralized document repository. A centralized system to store all files and documents for each of the team members is neglected as everyone else is trying to meet the demands of the other group units.

- Lack of requirement application documentation. A formal general requirement document for the functions of the RI system has never been documented. Although the purchasing group unit is already using an older RI system, most of the requirements are still mental models between project management and development team. Therefore, it is not possible to verify, validate, prioritize, and provide stakeholder traceability of the functional and non-functional requirements of this application.
- Lack of vision document. Requirement and architectural processes are informally captured, but there is no formal vision document for this application. This is a fundamental document so that anyone working on the project can easily understand the main objectives and major functions of the system. As quoted by Philippe Krutchen, who is considered the father of the Rational Unified Process, "If, instead of a fully robust process, I were permitted to develop only one document, model, or other artifact in support of a software project, a short, well-crafted Vision document would be my choice" (26).
- Limited stakeholder participation. Since most of the requirements came directly from the head of the VTCG, the defined indicators of the RI system will try to capture data such as man-hours at the individual level. That, in itself, seems like a questionable indicator because of the dynamic environment of operation for each group unit.

The Planning and Execution Process Phase

The usability case study, measurement processes, and activities that took place during the three months of the request system indicators pilot program included:

- Measurement and metrics discussion with team
- Usability case study approval from management
- Data capture of the new request system
- Selection of participants for the usability survey
- Creation of the surveys
- Distribution of the surveys in stages
    - Distribution of e-mail notifications to all participants
- Development of a new centralized Microsoft Collaboration SharePoint (27) team site
    - Posting of the results of the surveys in Survey SharePoint library collection
- Analysis of the results of the surveys
    - Improvement Action Plan
- Distribution/Presentation of the data to team members and management team

Some activities such as the distribution of surveys and the creation of a centralized website were done in concurrence with each other. Development of the centralized Microsoft Collaboration SharePoint site was not part of the objective for this study. However, it became a key element of the process of the usability case study to support the release of information about the survey and its results. As a side effect, it also became a solution for closing many gaps of the development environment within the

VTCG in relation to having a central place for document collection, for informing others of the status of the application, and an opportunity for people to provide feedback, recommendation, or critiques during the development life cycle of the RI system as well as other applications in progress.

MEASUREMENT AND METRICS DISCUSSION WITH TEAM

Prior to the implementation of the proposed usability case study for the current RI system, discussions with the manager and other team members were necessary. These discussions produced a buy-in of the activities to follow in the next month, which would also involve other team members from sub-group units of the VTCG. The usability case study was discussed with the development team to make sure these activities were for the sole purpose of improving the process and to try to close any gaps for the implementation and deployment of the RI system. As a result of those discussions, and with the support of the manager, the team accepted and understood the processes expected during the execution of the survey activities.

USABILITY CASE STUDY APPROVAL FROM MANAGEMENT

Since the GQM exercise was done within the VTCG, management has been very supportive of the measurement processes presented. As a result, a formal approval agreement with the software developer team lead and the manager of the VTCG was requested and granted. After final agreement and acceptance of this study, access to the RI system database was granted by the team lead of the application group for evaluating:

- How often are current users of the request system using the system during the pilot program?

- Which group units will be affected by the system?

For the usability case study, a survey approach and a brief analysis of the database were accepted as a practical and best approach for the current state of the project. It was agreed that the results of the usability survey would be discussed and analyzed within the team to better understand what type of measurements and processes must be addressed and implemented with the sole purpose of trying to close any gaps that could be found for the RI system.

The Pilot Program

DATA CAPTURING OF THE REQUEST SYSTEM

During the creation and distribution of the surveys, the RI system pilot was already running. For this case study, I collected data for the RI system pilot from June 28, 2010 until September 30, 2010. The main goal of the pilot program was to initiate the usage of the RI system by members of different group units. Management expected users to make request tickets for other members of the team, or even for themselves, and to provide feedback to the request development team. As mentioned earlier, the RI system also consists of indicators that show the number of request tickets, their owners, the man-hours, the open or closed status of a request, and distribution of requests by team groups and members. We will refer to the participants of this case study as the users of the RI system, who are the "requesters" or "owners" of a task.

The Surveys

This pilot program requires the participation of others to be able to receive feedback of their experience of the system in order to improve it. If a system is running as a pilot program and the participant has not been informed, or he does not know who to contact or how to access the system, then we are not running a pilot program. Consequently, we run the risk of losing a unique opportunity to collect valuable feedback from the stakeholders and the proper information for the usability surveys case study.

CHOOSING THE USERS FOR THE USABILITY SURVEYS CASE STUDY

The usability case study included a total of fifteen people. Nine of the chosen participants were part of the owners list of the request system. The other six participants were part of the PVE group, so their role would be mainly the requesters. They also included some managers and team leads from other group units external to the VTGC group. To avoid any data bias, The manager and the request system development team of the VTGC group were not part of the surveys. During the initial stage of the pilot program, there was a belief that the feedback of the participant was important, but it was unclear as to who was even aware of the pilot program existence. These facts were explored during the first survey as discussed in the survey #1 section.

CREATION OF THE SURVEYS

The information the VTGC group wanted to obtain from the usability survey case study would determine whether the request system development team was building the right metrics, and whether or not the driving goals from the management perspective would be satisfied by the system. Using a GQM first stages approach, questions were

generated as goals and sub-goals of what we wanted to learn. From those questions, and an assessment of the current state of the project, three surveys were created and divided into three main categories:

- Survey 1 - Awareness
- Survey 2 - System Embracement - Usability
- Survey 3 - Indicators Evaluation - Usability

Each survey was short, and would not take much time for the participants to respond with a quick "Yes" or "No" answer for Surveys 1 and 2, and a five-point rating scale for Survey 3.

The questions were progressive in nature. For example, if a user has not accessed the request system, the following questions about the system would not need to be answered. Another aspect of the survey questions was to build them in such a way that would be easily analyzed from a favorable or unfavorable perspective, so everyone could easily understand them in a graphical, indicator-like format.

THE QUESTIONS

Partial listings of the questions during the GQM process are listed below. Notice the (S#) appearing next to the questions indicates that survey number that led to the answers.

- Is everyone aware of the pilot program? (S1)
- Is there any resistance from the team, which could potentially make the usage of the application, or the current goals of the program fail? (S3)

- Does everybody share the same vision of the performance metrics as envisioned by management? (S2)
- Are the expected performance metrics easily captured in the current framework of operation? (S2)
- Are team members providing feedback to the development team? (S2)
- Are the proper requirements for the RI system being captured? (S2, S3)
- Is the application good enough for a final release at the end of the pilot program? (S2, S3)
- Are people embracing the application? (S2, S3)
- Is everyone comfortable with the tool? (S2, S3)
- Are the indicators a good measurement for meeting the performance goals of the team? (S3)
- Are the request forms easy to use? (S2)

Survey #1 - Awareness

Survey #1 was distributed in the last week of August 2010, almost two months after the initial deployment of the Request System Indicators pilot program. The questions for this survey were categorized in Awareness and Notification questions as shown in Figure 2.

Awareness Questions:

1	Have you ever been part of a performance improvement metric program?	**Yes No**
2	Are you aware that a new Request System with indicators as metrics is being developed and available as a pilot program?	**Yes No**
3	Do you know if you are one of the users of the request system?	**Yes No**
If you answered Yes to any above question then continue with the following, else skip to question 7.		
4	Do you know where/how to access the Request system?	**Yes No**
5	Has management or your supervisor informed you of the main motivation and the goals of the **Request** module portion of this pilot program?	**Yes No**
6	Has management or your supervisor informed you of the main motivation and the goals of the **Indicators** module portion that are part of the new request system?	**Yes No**

Notification Update Questions:

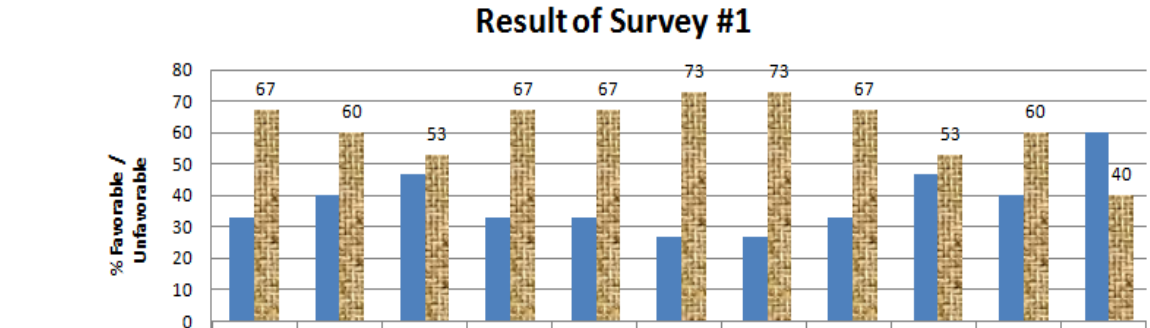
7	In the last two months have you received any verbal or notification by email about updates to the **request** system?	**Yes No**
8	Have you ever provided feedback to the application team of the new **request** system?	**Yes No**
9	Have you ever participated in the requirement process for the creation of the request system?	**Yes No**
10	Do you know how you can provide your feedback for the request system application?	**Yes No**
11	Do you think a BLOG web site would be useful to provide feedback about the request-indicators application?	**Yes No**

Figure 2: Survey #1 - Awareness and Notifications Questions

Survey #1 helped identify how aware the users were of the on-going RI system Indicators Pilot Program. It also helped identify the social environment in which the system was deployed, and the interaction that the participants had with the system and the development team. Feedback and participation of all stakeholders were critical for the improvement of the application.

THE RESULTS OF SURVEY #1

The results of Survey #1 are graphically shown in terms of favorable and unfavorable answers in Figure 3.



	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11
Yes - Favorable%	5 (33%)	6 (40%)	7 (47%)	5 (33%)	5 (33%)	4 (27%)	4 (27%)	5 (33%)	7 (47%)	6 (40%)	9(60%)
No - Unfavorable%	10 (67%)	9 (60%)	8 (53%)	10 (67%)	10 (67%)	11 (73%)	11 (73%)	10 (67%)	8 (53%)	9 (60%)	6 (40%)

Figure 3: Survey #1 - Awareness - Notifications Questions Results

In Figure 3, "Q#" represents the question number associated with the percentage of the favorable and unfavorable answers. This survey shows that out of the 15 participants, most questions had unfavorable responses. This survey was broken into two categories: awareness and notification update questions. For the awareness questions, in response to Question #1, "Have you ever been part of the performance improvement metric program?" ten, (67%) users had never been part of a performance metric program.

Thus, training for this type of program should be an area of improvement. Responses to Question 2 show that nine (60%) of the participants were not aware of the pilot program. The responses to all the other questions could be interpreted as a lack of communication between team members, team leads, and management. This is an inefficient environment for a pilot program to run if one expects others to use the system. The answers to the notification and update section show that 11 (73%) users had not received any notification for any updates of the RI system, and 10 (67%) had never participated in any requirement process of the RI system. Most users, nine (60%) agreed that a blog, or a team discussion for the RI system would be useful to them so they could provide feedback to the development team. The process of the distribution of the survey made people aware of the RI system pilot program.

COMMENTS AND FEEDBACK

Four (27%) out of the 15 people added some comments in relation to question 11, that a Wiki or Team Discussion component could be a solution for allowing team members to provide feedback. Another person suggested that demos would be a good idea since people do not have time for blog sites. Those comments led to the realization that a SharePoint collaboration team site could be a supportive artifact as a process improvement to the surveys.

POSSIBLE IMPROVEMENT ACTION PLAN

Based on the results of the survey, metrics alone have no meaning until we do something with the data. Therefore improvement action plans should be provided for execution (1) (12). The improvement action plans should be aligned with strategies

needed to achieve our goal (16). A possible solution derived from the surveys suggests an improvement action plan(s) for each of the results in Table 1. The left column includes the percentage and the number of people who answered Yes or No.

Q/A#	Participants Yes: Favorable No: Unfavorable %	Improvement Action Items
1	Yes: 5 (33%) No: 10 (67%)	Provide training for all members. A website could also be created with information about measurement and metric programs.
2	Yes: 6 (40%) No: 9 (60%)	Provide information about the goals of the RI system. A website constructed with that information could be a starting point.
3	Yes: 7 (47%) No: 8 (53%)	Keep users informed of their roles and expectations in relation to the RI system.
4	Yes: 5 (33%) No: 10 (67%)	Notify all users with information on how to access the RI system.
5	Yes: 5 (33%) No: 10 (67%)	Management can provide information on the main goals and the benefits of the RI system. A website could have all of this information available at all times.
6	Yes: 4 (27%) No: 11 (73%)	Management can provide information on the main goals and the benefits of the indicator modules. A website could have all this information available at all times.
7	Yes: 4 (27%) No: 11 (73%)	Notifications should be sent out to all participants any time new updates occurred on the RI system. A website could be set up with RSS so people could automatically receive updates.
8	Yes: 5 (33%) No: 10 (67%)	All team members should be encouraged to participate in the feedback process. A team discussion website could be available for this purpose.
9	Yes: 7 (47%) No: 8 (53%)	Designated team leads from each group unit could be in charge of providing feedback on regular basis.
10	Yes: 6 (40%) No: 9 (60%)	Provide contact information to all member and stakeholders involved in the process. In addition a team website could be used for this purpose.
11	Yes: 9 (60%) No: 6 (40%)	A blog, wiki page, or a team discussion in a website, as well as and regular meeting session with team leads of each group unit could be organized.

Table 1: Improvement Action Plan as Results of Survey #1.

EMAIL NOTIFICATION AFTER SURVEY #1

A thank you email notification for all the participants was sent after the surveys were collected. Information about how to access the RI system pilot program was sent to encourage the participants to try it out. The notification indicated that a second survey would follow-up in relation to the usability of the new request entry form and ease of use of the RI system.

Usability Survey #2 – System Embracement

Survey #2 helped identify the user's perspective in relation to the look and feel of the request entry form and its usability. This survey was distributed 14 business days after Survey #1 to all 15 participants. It was expected that, with that time frame between surveys, users would have had a chance to access the RI system. Figure 4 shows the four questions for this survey. Notice also that the questions are simple, but they can capture the first perception of end users' embracement toward system usability.

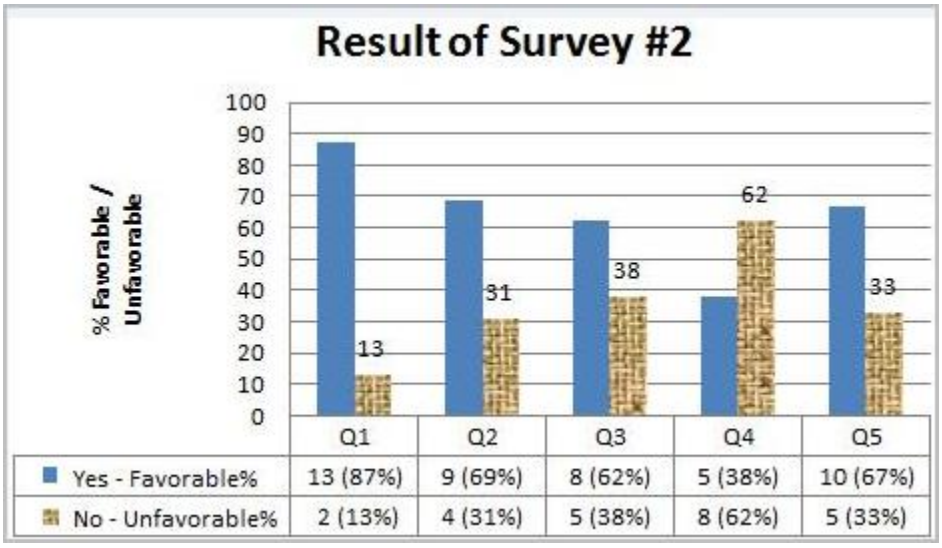
Your Initial Evaluation of the Request System

1	Have you visited the new request system? [redacted]	**Yes**	**No**
If you answer Yes to question 1 then continue with the following questions.			
2	Is the new Request system user friendly and easy to navigate to create or receive a request for others or yourself?	**Yes**	**No**
3	Do you think that the system will help you improve your performance by helping you to keep track of your requested/assigned tasks?	**Yes**	**No**
4	As an owner of a request, do you think this new request system could be an accurate reflection of all the tasks assigned to you?	**Yes**	**No**
If you answer **No** to any of the question above would you please make suggestions or comments below:			

Figure 4: Survey #2 - Usability System Embracement - Request

THE RESULTS OF SURVEY #2

Figure 5 displays a graphic representation of the result of Survey #2 in term of percentages of favorable and unfavorable responses. "Q#" means question number, and there are five questions.



	Q1	Q2	Q3	Q4	Q5
Yes - Favorable%	13 (87%)	9 (69%)	8 (62%)	5 (38%)	10 (67%)
No - Unfavorable%	2 (13%)	4 (31%)	5 (38%)	8 (62%)	5 (33%)

Figure 5: Survey #2 - System Embracement Results

ANALYSIS FOR SURVEY #2

Survey #2 shows mostly favorable answers. However, the information can also be misleading if we do not have a threshold value for the minimum percentage we expect for favorable answers. For the purposes of this survey, a participation of 100%, and answer results of 80% and above as favorable answers, would be considered a good indicator of a strong environment process with a high embracement of the request system among all team members. The 80% threshold selected was due to the fact that three of the participants, as shown later in the table in "Results of the Request System Pilot

Program," never accessed the request system. Survey #2 shows that most favorable answers are below that expected 80% range which indicates higher areas of process improvement are needed. Table 2 shows a summary of the percentage of the participants able to respond to the respective questions, and the number of participants with their percentage of favorable and unfavorable answers.

Questions	**Participation**	**Participants (Favorable %) Yes**	**Participants (Unfavorable %) No**
Q1. Have you visited the request system? (URL site omitted)	15: 100%	13 (87%)	2 (13%)
Q2. Is the new Request system user friendly and easy to navigate to create or receive a request for others or yourself?	13: 87%	9 (69%)	4 (31%)
Q3. Do you think that the system will help you improve your performance by helping you to keep track of your requested/assigned tasks?	13: 87%	8 (62%)	5 (38%)
Q4. As an owner of a request, do you think this new request system could be an accurate reflection of all the tasks assigned to you?	13: 87%	5 (38%)	8 (62%)
Q 5. Participants with comments	15: 100%	10 (67%)	5 (33%)

Table 2: Summary of Survey #2 - System Embracement

Question 1 indicates that most people know how to access the RI system, but two people had not yet visited the site. Therefore, Survey #1 opened the request system indicator pilot program to everyone who was expected to be part of the pilot program, but some may not have been aware of it. Question 2 shows that areas of the navigation, look, and feel of the request entry forms still need improvement. Question 3 shows there are a large percentage of users who do not see the value of how this tool can help them

improve their performance. Questions 4 shows that eight (62%) users believed the new request system would not accurately reflect the work done by most of the team members. Question 4 will be explained in the Comments and Feedback section below.

COMMENTS AND FEEDBACK

Comments are certainly a key component to help define the areas of improvement in the process. The survey specifically stated that any "No" answers to the first four questions required the participant to include their suggestions or comments. A total of ten out of fifteen participants provided comments, and most comments were very descriptive about the rationale for answering Questions 2, 3 and 4. Some of the quoted comments are:

> Too tedious to open, track, close and complete the tickets for the amount of requests we receive each day and every week. There are so many small and time consuming requests that all add up but are difficult to always add each task into the team's overall request.
>
> So I just went to the new request system and filled out an order; my first impression? Constructed well, flows well and it is concise….We do so many different tasks that it would be hard to officially state all of them as tasks. Plus we don't use the request system that often for creating new items/tasks/requests.
>
> User Friendly? 50/50 -Help me improve performance? Not too much - For tasks where I am directly accountable to others, this makes sense. I would struggle integrating this on existence task management. -Also, all buttons/levels of hierarchy should be clickable…
>
> When you click "Submit new task": 1-"summary" should change to "request/Task Needed" or something along those lines 2-The "Category" structure is odd…
>
> Tabs stops don't seem to work when switching fields… Until all users requests go through this tool, (forced) it won't receive traction… Man hour selection does not truly compare/weight tasks.
>
> I just looked at it and never used it.

This looks like an AR Tracker. Not all task I do will show up on a list.

The request system may work for teams that execute tasks in an extremely controlled environment… It requires a lot of work to keep adding every single task we do in such a fast paced environment… Man-hours would be hard to track

We do so many different tasks that it would be hard to officially state all of them as tasks. Plus we don't use the request system that often for creating new items/tasks/requests.

For tasks where I am directly accountable to others, this makes sense. I would struggle integrating this in existing tasks managements.

POSSIBLE IMPROVEMENT ACTION PLAN

Table 3 presents some possible improvement action plans for questions 2 - 4.

Q/A#	Participants Favorable Unfavorable %	Improvement Action Items
2	Yes: 9 (69%) No: 4 (31%)	Set up a few meetings with each group unit for input on the look and feel of the site. Provide a team discussion site where people can make suggestions.
3	Yes: 8 (62%) No: 5 (38%)	Re-evaluate the functionalities and goals of the RI system with the team lead of each group unit. Re-define and prioritize the type of input fields of the request ticket that can be supported by the system.
4	Yes: 5 (38%) No: 8 (62%)	Re-evaluate the functionalities and goals of the RI system with the team lead of each group unit. Identify the type of constraints that each member of the RI system experienced based on their functions.

Table 3: Survey #2 - Improvement Action Items

EMAIL NOTIFICATION AFTER SURVEY #2

After completion of Survey #2, another thank you notification email was sent to all participants. This step was taken, with several objectives:

- To emphasize the importance of their participation.
- To announce, with the respective URL link, that a new SharePoint collaboration team site with information about the request system indicators pilot was now available, and that all the results for Survey #1 and Survey# 2 had been posted.
- A notification that a final Survey #3 would be distributed for capturing the usability of the Indicators modules from the Request System.

Survey #3 - Indicators Evaluation - Usability

Survey #3 focused on the usability of the performance indicators within the RI system from the perspective of the requester and/or owner of a task. For this survey we used a 5-point rating scale for value of importance, low to high, to each of the indicator modules. Survey #3 is shown in Figure 6 and it was distributed to the same fifteen people of Survey 1 and 2. Notice that an open question is added at the end of this survey for feedback.

Please select (circle) a value between 1 and 5 for each indicator listed below; the rating should be based on the added benefit that the indicator provides for improving performance for yourself or your team. Note, that this is not a complete list of all indicators available, but a short list to provide a better focus for feedback.

Value of Importance: A value of 1 indicates that it is of lowest value to you or your team, and a value of 5 is the highest value.

THERE IS NO GOOD OR BAD ANSWER.

Indicator	Value of Importance
1- Distribution /Owner	1 2 3 4 5
2- Man Hours Distribution /Team	1 2 3 4 5
3- Man Hours Distribution /Owners	1 2 3 4 5
4- By WW / Team	1 2 3 4 5
5- By WW /Owners	1 2 3 4 5
6- Man Hours By WW /Team	1 2 3 4 5
7- Man Hours By WW / Owners	1 2 3 4 5
8- Average Man Hours By WW /Team	1 2 3 4 5
9- Average Man hours By WW /Owner	1 2 3 4 5

Please provide any other comments feedback below (or in back of page):

Figure 6: Survey #3 - Usability of Indicators

As previously mentioned, the indicators are graphical drill-down charts that allow management and team members to view the distribution of requests, and the number of hours spent on fulfilling requests by group units as well as by the team members. A brief description of the purpose of each of those indicators is listed below.

- Distribution/Owner: A graphical chart, or indicator module, showing the percentage of tasks distributed by owner.
- Man Hours Distribution/Team: A graphical chart showing the percentage and number of hours spent on tasks for each of the group units.
- Man Hours Distribution/Owners: A graphical chart displaying the percentage and number of hours spent on all tasks by a team member.
- By WW/Team: A graphical chart displaying a percentage of the number of tasks per workweek and by group units.
- By WW/Owner: A graphical chart displaying a percentage of the number of tasks per week by owner and group units.
- Man Hours by WW/Owner: A graphical chart displaying the number of hours the owners of tasks have spent per work week.
- Average Man Hours by WW/Team: A graphical chart displaying the average man hours that each group unit has spent per work week.
- Average Man hours by WW/Owner: A graphical chart displaying the average hours each team member spent per work week.

Figure 7 shows an example of the Man Hours Distribution / Owners indicator. For the purpose of this paper, the user names have been marked out. By clicking on the associated user name, not shown in the picture, you can drill-down to the user's Man

Hours distribution across the group units and eventually drill-down to the users' detail request list.

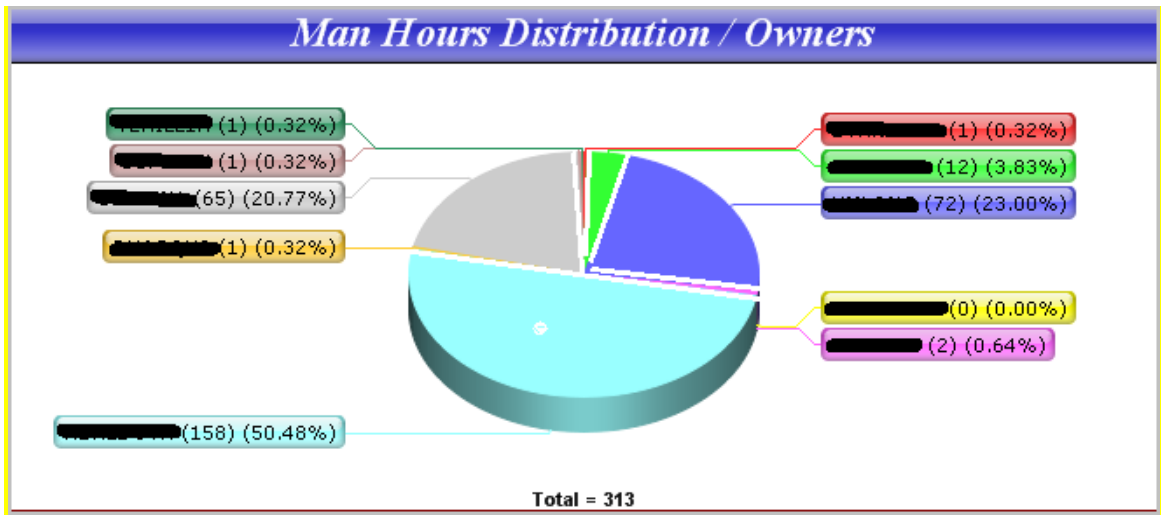


Figure 7: Man Hours Distribution/Owners

THE RESULTS OF SURVEY #3

The results of Survey #3, as they appeared in the SharePoint Survey team site of the RI system, are shown in Figure 8 below.

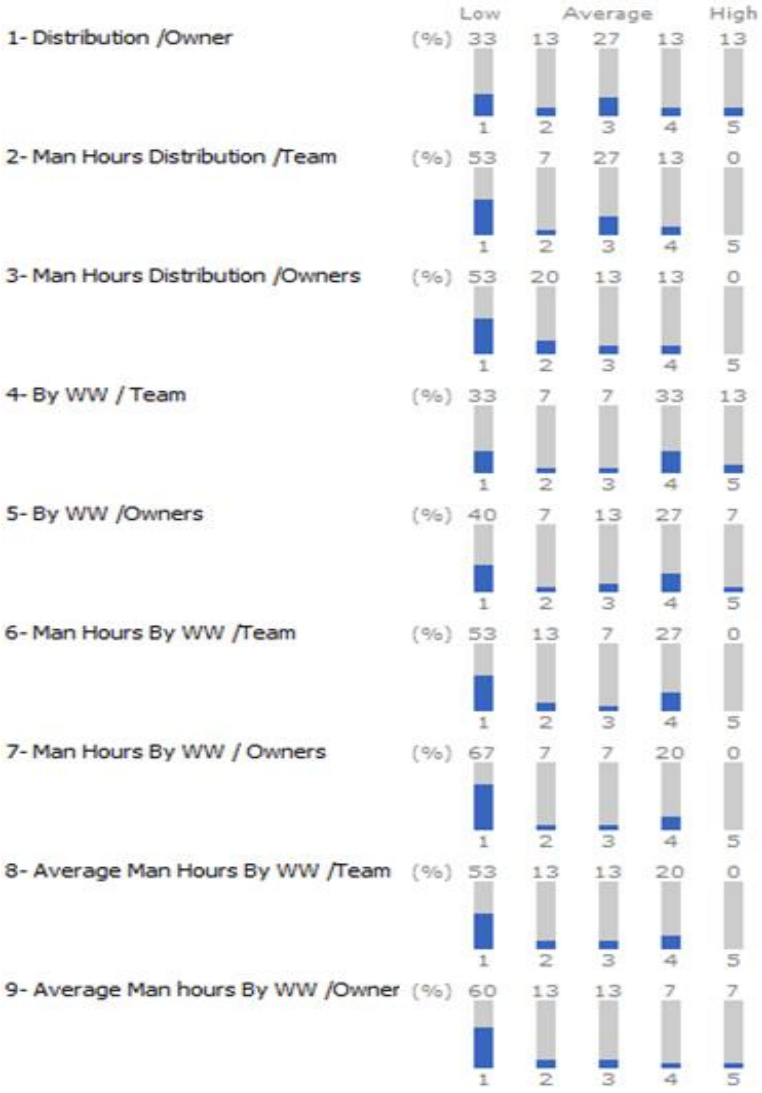


Figure 8: Survey #3 - Results - Usability of Indicators

The results of Survey #3 were very discouraging since it showed that the majority of the end users did not embrace the indicators as management would have expected. Most of the indicators that measure man-hours, such as indicator 2, 3, 6, 7, 8, and 9, had a higher disapproval rate. From discussion with some of the team members, a few showed signs of resistance, as expected. Nevertheless, most of the respondents' disapprovals

were related to the constraints imposed by the operational working environment for entering requests into the system, and not the actual indicator.

COMMENTS AND FEEDBACK

Providing comments was an option for Survey 3. Out of the fifteen participants, seven of them provided some very valuable feedback, one of them expanded on the difficulty of capturing some of the data, as follows,

> Response 1 - We are a customer driven team and our workload is somewhat random and it is sometimes impossible to predict. As we received requests, they are distributed to the next available person. Therefore, by team is the only decent indicator to capture the entire workload. It is too difficult to track by individuals and owner is usually myself so it is figure out the distributed team performance… Response 8: Average man hours are hard to track and encompass all tasks involved in completing any one requests.
>
> None of these Indicators apply to Inventory Control Techs.
>
> Need a way back to main page once you are sent to indicator page.
>
> Indicators from a GUI perspective are very linear and they are not user-friendly. I cannot find any definition about them!… It took me a while to make sense out of them. They need to stay in the same frame and within the site. Using the back button to return to a previous page is very confusing and annoying! It is very easy to lose track of what you are looking at.
>
> They seem very vICE specific. I am not sure how they can be applied to a broader customer base.
>
> None of these apps are applicable to green badge inventory control employees.
>
> We care about distribution of tasks by teams, but everything else does not apply.

POSSIBLE IMPROVEMENT ACTION PLAN

For each of the indicators evaluated in Survey #3, we have some improvement action plans, in addition to the ones mentioned for Survey #1 and #2,:

- Re-evaluate each of the indicators from the perspective of each of the team leads and managers of the group unit.
- Prepare, plan and execute new GQM sessions with each of the team leads and managers, to discover their goals and sub goals, of what they want to know, to determine what metrics are needed by each of the group units in order to achieve their performance indicators.
- Implement a new training program for performance indicator measurements.
- Identify the roles, responsibilities, and expectations of each of the participants.

EMAIL NOTIFICATION AFTER SURVEY #3

An email notification was sent to all participants once Survey #3 was posted in the collaboration SharePoint team site for the request system indicators. The team site was indicated as the place for posting recommendations.

Development of a Collaboration Team Site

To support the distribution of information in relation to the pilot program and the usability case study, a Microsoft SharePoint website was built to fill the gap for the lack of a centralized documentation repository. The idea was to build collections of sub-sites for each group unit of the VTCG. For the application team this was a great starting point for process improvement. Therefore, team sites for each of the group units were created. A sub-site for each of the applications in development was also created. The Request System application team web site was set up, with Team Discussion, Wiki, Calendar, Events Notification, and Survey's Feature Components as shown in Figure 9.

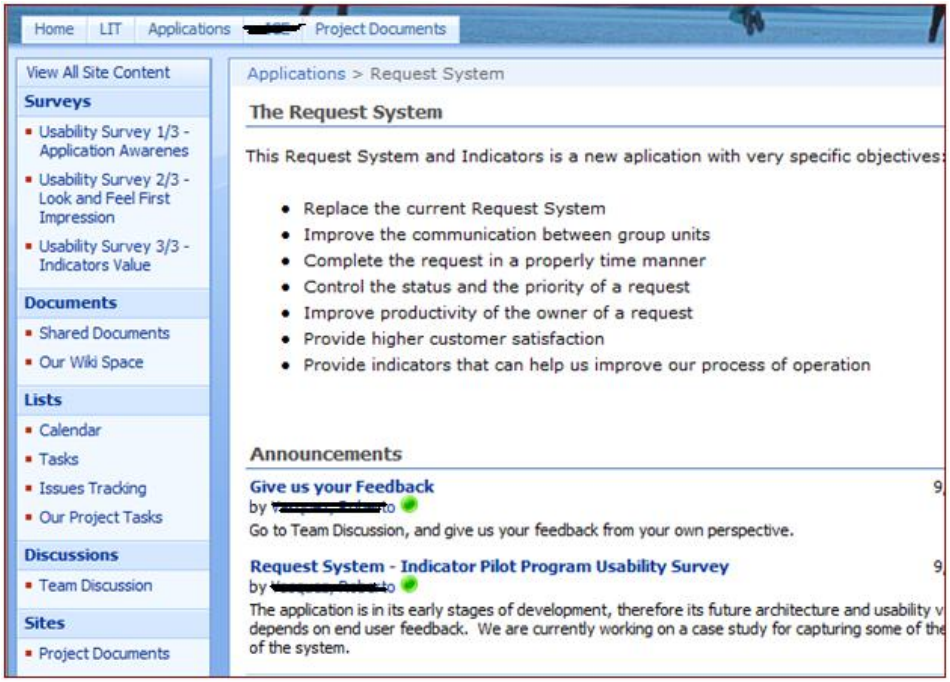


Figure 9: The Request System Collaboration Web Site - Surveys

Results of the Request System Pilot Program

The Request System Indicators Pilot Program ran for three consecutive months. Table 4 displays the results of the total number of requests created and received by the same 15 participants during the execution of the survey usability case study. The first column shows the general functional role that those participants had within the organization.

		June		July		August		September	
Role	User	Created Requests	Received Requests	Created Request	Received Requests	Created Request	Received Requests	Created Request	Received Requests
Inventory Control (IC) Tech	User1								1
IC Tech	User2								1
IC Tech	User3								1
IC Team Lead	User4		1				1		
App Team Lead	User5			16	18		1		2
Manager	User6			2	2		1		1
Manager	User7								
Team Lead	User8			2				2	
Manager	User9			1					
App Team Lead	User10					1		8	
Lab Engineer	User11							1	
IC Tech	User12								
Lab Engineer	User13								
Team Lead	User14			1				1	
Team Lead	User15			2		2			1

Table 4: Number of Total Requests Created and Received per Month by Users

From the results, User5 was the only user who created 16 requests for himself and two other requests for others. All other users created or received requests from other users. User6 created two requests and received two requests from other users. Some of the users received requests from users that were not part of this study. Three of the

survey participants, User7, User12, and User13 never created or received a request. Further analysis indicated that User7 was one of the users who gave a high rate of importance to each of the indicators used in survey #3 but never created or received a task, and has never been part of a measurement program. The results of the survey and the data collected speak for themselves. With such a low rate of usability of the system, and the unfavorable answers of the survey, this project requires a great deal of improvement and a re-evaluation of the application as indicated in each of the sections of the "Possible Improvement Action Plans" outlined after each survey.

4. CONCLUSION

The results of the usability survey case study and the data collected clearly show that the three months of the Request-System Indicators Pilot Program failed to meet most, if not all, of the expectations of the VTGC manager. First, survey participants did not become active users of the system. Second, the idea of capturing man-hours from the RI system was not embraced, and was considered rather inappropriate by most of the participants, including team leads and managers. Third, the system is not ready for production and has a long way to go to be useful as a performance indicator tool by most group units. This application had been developed without any proper requirement documentation, prioritization of the functionality, and exclusion of other team leads' stakeholders for each group unit. These results clearly indicate the failure of this pilot program. The application was also developed from one stakeholder's point of view and went primarily from verbal requirements directly into development mode. This process, which is wrongly referred to as the 'agile' methodology by some developers, is really an ad-hoc prototyping style, which is typically not sustainable and usually leads to systemic application development failure.

The original intent and development of the RI system may have been, and may still be, the right thing to do from the perspective of the VTGC manager and the RI software developer, but without the buy-in and active participation of all the stakeholders, we could easily predict that this application will not have a happy ending. Currently, major process improvements are needed to move this application in the right direction.

The surveys were a successful tool since they provided the answers, be it unfavorable, to the manager's questions, as mentioned in the Questions section of this paper. Preparing the right environment to present those answers by building the

SharePoint site is believed to be an effective method to keep everybody informed of the survey results, and provide a central location for the RI development team to promote the features and the new system changes. Yet, after all these efforts, the SharePoint collaboration solution site did not create the expected results that it was intended to do. First, the site did not encourage the development team of the request system to add any new information about the features and goals of the RI system. Second, although nine (60%) of the participants agreed that a team discussion site would be a good place to provide feedback during the development cycle, at four weeks after the completion of the pilot program, no one has yet added any new suggestions. Also, after four weeks, only a total of two requests had been created by two of the fifteen original participants.

On the positive side of systematic process improvement, the results of the surveys were discussed with the team and the VTGC manager. To move the project forward, it was decided that the future RI system would limit its focus to one specific group unit and associated stakeholders. In addition, a new survey is planned for distribution for another customized application. Another solution that should be considered is to look for third party tools that could provide very similar functionalities needed by the request indicators system using the requirements derived from the failed pilot program.

The origins of VTCG were not intended to create internal customized applications but instead to build and support the needs of the entire lab infrastructure, including Lab, Inventory, Shipping, and Purchasing. Also, the VTCG team tries to do whatever needs to be done, as priority arrives, to get the job done. And they have been very successful in doing so for several years. But in terms of building applications, this is not the same case. From discussions with VTCG members and observation of their business processes, there are two very significant problems that appear to be at the root why the RI system, and applications like it, are suffering. Both problems are directly related to the

origins of what the VTCG operations used to be, a small supportive group, as opposed to what the VTCG has now become, a large supportive group. First, from the perspective of the VTCG manager, the application group unit is not a software development team. Therefore, any software development processes that involve formal requirement documentation, as well as architecture of the system, are not seen as important or as key elements to expedite the building process. Second, from the perspective of the application team, they are still operating in that heroic mode, CMM level 1, of trying to please the VTC manager by trying to do whatever they are asked to do, even when the development targets for building a tool are completely unrealistic. Therefore, when a new deadline is not met, a new priority takes place, and the cycle keeps repeating. Unfortunately, this framework of operation has very negative impacts to any development group, as proved in the result of this study. For as long as the VTCG application group units are continuously being asked to build customized applications, it is an unquestionable fact that the application group unit has become a development group, and it is in the interest of everybody in the VTCG to unite and work toward improving their own development process. But also, for as long as the application team continues to work in that heroic mode of "I can do it" and ignores the reality of software good practice development processes, they will continue to find that their application will not meet the expectation of the stakeholders. In the long run, and as long as this group continues to work in this processed operation, the application team will not evolve to the next level of productivity and it's applications will be nothing more than prototypes, unreliable and un-scalable pieces of software. Otherwise, if the VTCG is not a software development group, then a solution is to look for third-party tools that could provide very similar or better functionalities for the applications needed to support the core of their business in terms of shipping, inventory, purchasing, and ticket request applications. At

least by taking this step and trying to find third party tools, that will free the team from the burden of trying to build so many types of applications that lack any of the good qualities expected from any application tool, including scalability, ease use, reliability, and maintainability.

BIBLIOGRAPHY

1. **Ebert, Christof, et al.** *Best Practices in Software Measurment.* Berlin : Springer, 2005. ISBN 3-540-20867-4.

2. **McAndrews, Donald R.** *Establishing a Software Measurement Process.* Pittsburgh : Software Engineering Institute, 1993. CMU/SEI-93-TR-16.

3. **Ebert, Christof and Dumke, Reiner.** Making Measurement a Success - A Primer. *Software Measurement.* New York : Springer, 2007, pp. 19-23.

4. **Wikipedia.** Software development methodology. [Online] [Cited: October 2, 2010.] http://en.wikipedia.org/wiki/Software_development_methodology.

5. **Wideman, R. Max.** Software Development and Linearity (Or, why some porject management methodologies don't work). *ICFAI PRESS.* March 2003, p. 18.

6. **Krasner, Herb, Curtis, Bill and Iscoe, Neil.** Communication Breakdowns and Boundary Spanning Activities on Large Programming Projects. *Empirical Studies of Programmers, Ch. 4.* s.l. : Ablex Publishing, 1987, pp. 47-64.

7. Gauging Developers Acceptance of Software Metrics. *Metrics Acceptance.* [Online] [Cited: July 24, 2010.] http://metricsacceptance.net/.

8. **Begnoche, Luc, Abran, Alain and Buglione, Luigi.** A Measurement Approach Integrating ISO 15939, CMMI and the ISBSG. *Software Engineering Research Laboratories.* [Online] 2009. [Cited: July 29, 2010.] http://www.gelog.etsmtl.ca/publications/pdf/1060.pdf.

9. **Paulk, Mark C, et al.** *Capability Maturity Model for Software (Version 1.1).* Pittsburgh : Software Engineering Institute, 1993. CMU/SEI-93-TR-24.

10. **Perrin, Burt.** Effective use and misuse of performance meausurement. *American Journal of Evaluation.* 1998, Vol. 19, 3, pp. 367-379.

11. **Sultanoglu, Senser.** Sofware Measurement, Software Measurements Home Page, metrics, quality, cost estimation, size estimation, COCOMO, fun. [Online] [Cited: June 6, 2010.] http://yunus.hacettepe.edu.tr/~sencer/research.html.

12. **Pandian, C.** *Software Metrics.* Florida : CRC Press LLC, 2004. p. 1.

13. **Florac, William A and Carleton, Anita D.** *Measuring the Software Process.* Berkley : Addison-Wesley, 1999. ISBN 0-201-60444-2.

14. **Swanson, Richard A.** *Analysis for Improving Performance.* San Francisco : Berrett-Koehler Publishers, 2007.

15. IEEE Std. 1061-1998. *Standard for a Software Quality Metrics Methodology, revision.* Piscataway, N.J. : IEEE Standards Dept., 1998.

16. **Basili, Victor R, Caldiera, Gianluigi and Rombach, H Dieter.** The Goal Question Metric Approach. *Goal-Question-Metric (GQM).* [Online] [Cited: October 2, 2010.] https://goldpractice.thedacs.com/practices/gqm/.

17. **Park, Robert E, Goether, Wolfhart B and Florac, William A.** *Goal Driven Measurement - A Guidebook.* s.l. : Software Engineering Institute/Cernegie Mellon University, 1996. Technical Report CMU/SEI-96-HB-002.

18. **Schneidewind, Norman F.** New Software-Quality Metrics Methodology Standard fills measurement need. April 1993, Vol. 26, 4.

19. **Schneidewind, Norman F.** Body of Knowledge for Software Quality Measurement. February 2002, Vol. 35, 2, pp. 77-83.

20. International Organization for Standardisation. (ISO) (1991). *ISO/IEC: 9126 Information technology- Software Product Evaluation - Quality Characteristics and guidelines for their use - 1991.* [Online] [Cited: October 2, 2010.] http://www.cse.dcu.ie/essiscope/sm2/9126ref.html.

21. **Abran, Alain, Khelifi, Adel and Seffah, Ahmed.** Usability Meanings and Interpretations in ISO Standards. *Software Quality Journal.* Vol. 11, 4, pp. 325-338.

22. Usability - Wikipedia, the free encyclopedia. *Wikipedia The Free Encyclopedia.* [Online] [Cited: October 4, 2010.] http://en.wikipedia.org/wiki/Usability.

23. **Bevan, Nigel.** Measuring usability as quality of use. *Software Quality Journal.* 1995, Vol. 4, 2, pp. 115-130.

24. **9241-11, ISO.** Ergonomic requirements for office work with visual display terminals (VDTs) - Part 11:Guidence on usability. s.l. : International Organization for Standardization, 1998.

25. **Meyer, Bertrand.** The Unspoken Revolution in Software Engineering. *Computer.* January 2006, pp. 124-123.

26. **Leffingwell, Dean and Widrig, Don.** Managing Sowftware Requirements. *Managing Software Requirements.* Boston : Addison Wesley, 2008.

27. Business Collaboration Platform for the Enterprise and the Internet. *Microsoft SharePoint 2010.* [Online] Microsoft Corporation. [Cited: October 8, 2010.] http://sharepoint.microsoft.com/en-us/Pages/default.aspx.

Vita

Roberto Mario Vasquez was born and raised in El Salvador, the son of Roberto Mario Vasquez and Luz Margarita de Vasquez. He came to the United States in 1981, and lived in New Jersey while attending Kean College of New Jersey for almost two years. He transferred to The University of Texas at Austin in 1983 to pursue a major in Electrical and Computer Engineering, and graduated in December of 1987 with a Bachelor of Science in Electrical Engineering. He started working with a consulting engineering firm prior to his graduation where he developed one of the first RS-232 data acquisition applications for the Macintosh. Since then, he has worked for many corporations as a software consultant and has played many different roles in the life cycle of the software development process. In January of 2009, he entered The Executive Software Engineering Graduate Program at the Cockrell School of Engineering at The University of Texas at Austin to pursue the degree of Master of Science in Engineering with a major in Electrical and Computer Engineering and a specialization in software engineering.

Email: rmvasquez@austin.rr.com

This report was typed by the author.