# ANALYZING DISCREPANCIES IN A SOFTWARE DEVELOPMENT PROJECT CHANGE REQUEST (CR) ASSESSMENT PROCESS AND RECOMMENDATIONS FOR PROCESS IMPROVEMENTS

by

Kenneth James Cunningham

A Technical Management Research Project
Submitted to the Extended Campus
In Partial Fulfillment of the Requirements of the Degree of
Master of Science in Technical Management

Embry-Riddle Aeronautical University
Extended Campus
Houston Center
October 2003

# ANALYZING DISCREPANCIES IN A SOFTWARE DEVELOPMENT PROJECT CHANGE REQUEST (CR) ASSESSMENT PROCESS AND RECOMMENDATIONS FOR PROCESS IMPROVEMENTS

by

Kenneth James Cunningham

This Technical Management Research Project
was prepared under the direction of the candidate's Research Committee Member,
Dr. Marian C. Schultz, Associate Professor, The University of West Florida,
and the candidate's Research Committee Chair,
Dr. James T. Schultz, Associate Professor, Extended Campus, and has been
approved by the Project Review Committee. It was submitted
to the Extended Campus in partial fulfillment of
the requirements for the degree of
Master of Science in Technical Management

PROJECT REVIEW COMMITTEE:

__________________________________
Marian C. Schultz, Ed.D.
Committee Member

__________________________________
James T. Schultz, Ed.D.
Committee Chair

# ABSTRACT

Researcher: Kenneth James Cunningham

Title: Analyzing Discrepancies in a Software Development Project Change Request (CR) Assessment Process and Recommendations for Process Improvements

Institution: Embry-Riddle Aeronautical University

Degree: Master of Science in Technical Management

Year: 2003

The Change Request (CR) assessment process is essential in the display development cycle. The assessment process is performed to ensure that the changes stated in the description of the CR match the changes in the actual display requirements. If a discrepancy is found between the CR and the requirements, the CR must be returned to the originator for corrections. Data was gathered from each of the developers to determine the type of discrepancies and the amount of time spent assessing each CR. This study sought to determine the most common types of discrepancies, and the amount of time required to assessing those issues. The study found that even though removing discrepancy before an assessment would save half the time needed to assess an CR with a discrepancy, the number of CR's found to have a discrepancy was very small compared to the total number of CR's assessed during the data gathering period.

## TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# CHAPTER I

# INTRODUCTION

Errors cost time and money. The same is true for errors found in the display requirement used by the Display Development Team (DDT). Errors are to be expected during the development process, but for the DDT, a contractor for a large government organization, errors can be deadly. Therefore, it is necessary to remove all errors from laptop displays, developed under the contract, before delivery. "It is important that any system errors be eliminated before programming commences or they can cause havoc when discovered later" (Brown & Sampson, 1973, p. 18).

## Background of the Problem

The DDT develops displays that receives data from a specified source, translates it, and display it on a laptop. The requirements on how to develop displays are provided by an outside source known as the customer. A requirement for each display contains a picture of how the display is to look, and a text file which explains how the objects on the display are to function.

The customer submits new requirements for a new or existing display through a procedure known as a Change Request (CR). The CR is submitted with a primary form which contains information pertaining to the changes being requested. Attached to the CR form are new requirements for each display being updated or created. A CR can contain display requirements for multiple displays.

The DDT developer does not know the details about what the display is to control, or what information it is going to receive. One only knows what the requirements attached to the CR state. Failure to develop a display to requirements will result in inconsistencies between the documentation and the actually display. Displays being developed must match the requirements. The developer can not differ from the requirements submitted by the CR since they do not have the knowledge available on how a display is to function outside of the requirements. Therefore, if the display requirements provided in the CR are not correct, then the display will also be incorrect. It is important for the changes in the CR to be without errors so that the display will function correctly when delivered. Safety is of the highest priority, leaving no room for error.

There is a procedure in place, known as the assessment process, for the developer to check requirements for problems before the requirements are accepted and the displays are developed. This procedure is performed when a CR is received from the customer. The developer conducts a series of test to ensure the changes stated in the CR form are the only changes made to the display. The two problems a developer can find in a CR are: (a) If something changes in the requirements "not stated" in the CR form, or (b) the requirements do not make the changes stated in the CR form. If either problem is found, the CR is returned to the customer so that they may revise and correct either the CR form, or the attached requirements. Once corrected, then the customer resubmits the CR form and attached requirements for re-testing. When the developer has completed the CR procedure and is satisfied that the CR form matches the expected changes in the requirements, then he or she submits it for approval.

The CR then goes before a board where it is reviewed and either approved or disapprove. Once the board approves a CR, the displays are now developed and the attached requirements can no longer be changed. If problems are discovered with the requirements, then a new CR is required to update the requirements, and must transverse though the entire CR process once again.

## Researcher's Work Setting and Role

The researcher has functioned as a developer for the DDT for two years. Within the DDT, he is a subsystem lead and has the responsibility to perform all assessments on his subsystem CRs that are requested. He has become very proficient at the current assessment process with the DDT, and has had hands-on experience with the types of problems that can occur. Before working for the DDT, he attended Oklahoma Panhandle State University, and graduated in the class of 2000 with a Baccalaureate in Computer Information Systems, with dual minors in Business Administration and Fine Arts.

## Statement of the Problem

The customer submits a CR to CDDT. The CR is then tested by the CDDT developer for any errors. If an error is found, the CR is returned to the customer to revise. Each time the CR is revised, the developer must re-test the CR for errors. This cycle can occur numerous times and is very time consuming. The cost of development increases because of the time spent testing the same CR multiple times. This cost leads to the misuse of labor which, in turn, leads to a very inefficient system. If more of the errors could be located by the customer before the CR is forwarded to the developer, then the developer would spend less time on assessing a CR, and more time on development.

## Acronyms

| | |
|---|---|
| DDT | Display Development Team |
| CR | Change Request |
| ECLSS | Environmental Control & Life Support System |

| | |
|---|---|
| ETF | Error Tracking Form |
| GUI | Graphical User Interface |
| MyIssues | A time recording program |
| PUI | Program Unique Identifier |

## Limitations

The original purpose of this project was to assess the cost of error that are not detected in the assessment process. Since another CR must be created to correct an error in an approved CR, the goal was to track these errors and follow the CR trail. This data could be used to establish the cost of errors not detected in the assessment process. Unfortunately, no current information exists, and gathering the necessary data would require a substantial amount of time and would prove to be very costly. Additionally, the time required to return the CR to the customer is not tracked, so these costs cannot be qualified.

## Assumptions

It is assumed that cost was calculated by hours. This number can be used by management to apply a dollar value. It was also assumed that the data entered into the database by the developer, pertaining to the errors located in the CR during the assessment process, was entered correctly. Any incorrect entries or miss information will be unknown. The form was tested before the sample period to limit these types of errors. Procedures were also developed on how to enter the error data onto the form.

# CHAPTER II

# REVIEW OF RELEVANT LITERATURE AND RESEARCH

Developing displays requires detailed requirements. In the CR system, the customer provides these requirements to the developer. The CR process is the way the customer communicates with the developer and vice-versa. When a developer receives a CR, they assess it for errors. A better understanding of the assessment process is necessary so that common errors found can be discovered and corrected earlier in the process.

## The Software Development Process

Before examining the DDT assessment process, it is necessary to define the term "process". Merriam-Webster's Collegiate Dictionary (1993) states that a process is a series of actions or operations conducing to an end. A better definition can be found at Dictionary.com (n.d.) which states that a process is a series of operations performed in the making or treatment of a product.

## Customer Requirements and Errors

At the center of the assessment process is the display requirement. "Requirements are traditionally the cornerstone of most projects. They define the scope and effort, and ultimately determine the schedule. However, as a project takes shape, usually additional requirements are discovered. This "scope creep" typically impacts resource loading and slides the schedule" (Jarvis & Hayes, 1999, p. 3).

Depending on various viewpoints, errors are always someone else's fault: operators blame programmers, programmers blame systems analysts, analysts blame users and users blame all of the others (Brown & Sampson, 1973). "Assuming the data input is correct, there are five possible types of error 'internal' to a development programming system: hardware, software, operation, programming, and systems" (Brown & Sampson, 1973, p. 2). Table 1 shows the percentage of system failures due to the five types of error.

Table 1

*Percentage of System Failures due to Type of Errors*

| Percentage of system failures due to: | |
|---|---|
| Hardware | 1% |
| Software | 2% |
| Operating | 5% |
| Programming | 90% |
| Systems | 2% |

*Note.* from Brown & Sampson, 1973, p. 2.

Table 1 assumes the correct input data has been provided.

> However, not all causes of program errors are within the control of the programmer. There are much wider issues–-such as the quality, housing and organization of staff and the work standards they are expected to follow–-that have just as much effect on the incidence of bugs as the programming techniques adopted by the individual. (Brown & Sampson, 1973, p. 17)

"If the programmer does not clearly know what he is trying to achieve he is hardly likely to produce useful results" (Brown & Sampson, 1973, p. 18).

## Research on CR Process

The CDDT CR Assessment Process is a procedure the CDDT developer must perform on each display in a CR. This procedure requires the developer to initiate a series of tests using in-house developed tools on the display requirements. Most of the tools compare the new display requirements with previous display requirements. Other tests check the requirements for formatting errors. If the display submitted in the CR is a new display, only the formatting errors are checked. After all the display requirements in the CR have completed the CR assessment process, and no errors have been found, then the CR is submitted for approval by a board. If errors are found, the issues are forwarded to the appropriate customer, along with the CR (Robertson, n.d.).

## Statement of the Research Question

"It is important that any system errors be eliminated before programming commences or they can cause havoc when discovered later" (Brown & Sampson, 1973, p. 18). The first step to lowering the number of errors submitted by the customer is to ascertain the type of error that are found during the assessment process, and the cost associated with them. The research question is whether procedures for submitting and reviewing CRs by the customer have high error rates, and contribute to the cost of development.

# CHAPTER III

# RESEARCH METHODOLOGY

## Research Design

This is a descriptive study that analyzed the types of CR errors found throughout the assessment process. The developer utilized an electronic form called the Error Tracking Form (ETF) to track the number of each type of error. This data provided details on which errors are commonly being found in the assessment process. Analyzing this data revealed problem areas which can now be examined in order to develop more effective ways to locate and evaluate them earlier in the process.

## Research Model

During a development cycle, the developer performed the assessments as normal. However, the DDT developers were asked to keep track of errors by entering data into the ETF. This data was then be used to indicate which errors were the most common. The time it requires the developer for each assessment was also tracked by the ETF. The time it required, and the number of errors found, were then combined to indicate the assessments cost, in terms of development hours, for each type of error.

## Survey Data

The data utilized in the study was for the display CR's that are assessed by DDT developers during development cycle. A development cycle is the time frame in which the customer submits a CR, and the developer implements the changes. A development cycle can lasts as long as to six months. There are 12 developers in DDT, and twenty-three CR's were assessed during the sample period.

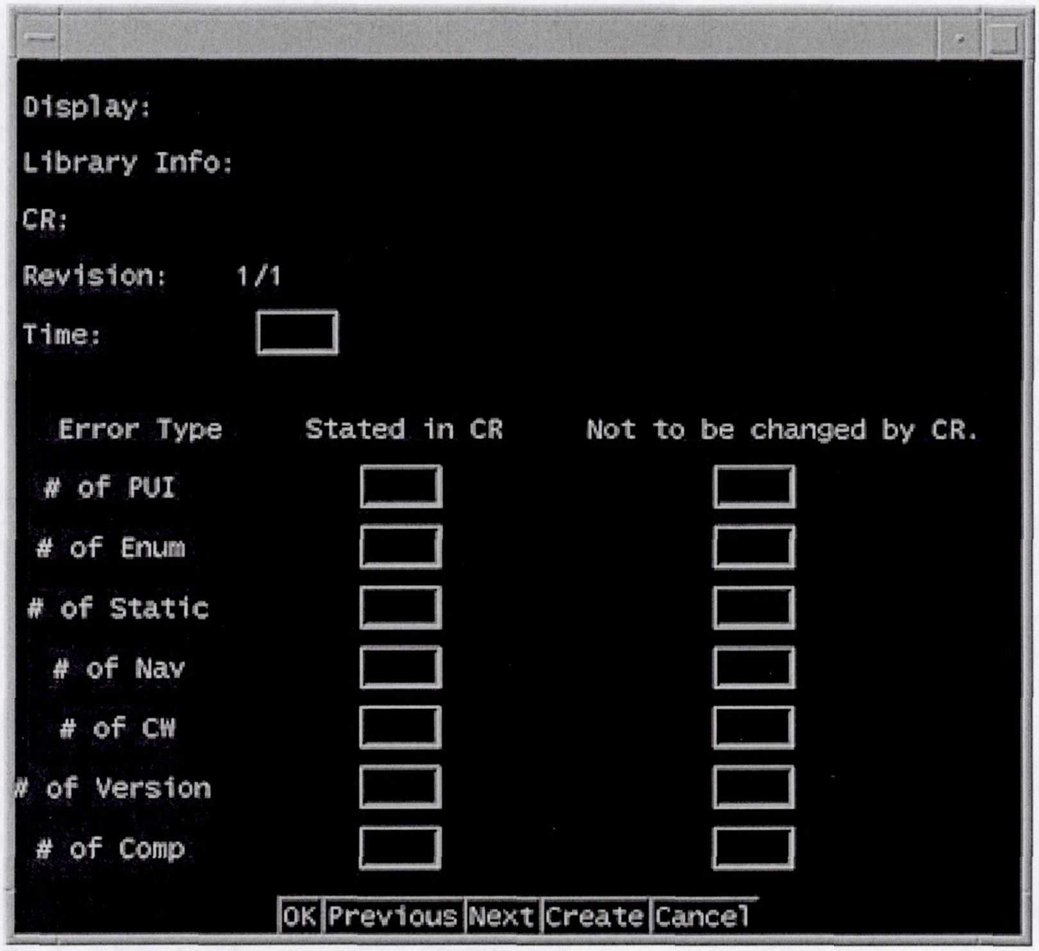
## The Data Gathering Device

Data was gathered from the developer during the assessment process through the ETF. Data pertaining to the CR assessment process was only entered if the CR was returned to the customer. If the CR passed with out issue, then it was not tracked by the ETF. The ETF be completed each time a developer forwarded the CR back to the customer. The ETF asked for a CR number, display name, revision number and had a list of errors to choose from. The ETF revision number kept track on the number of times the CR had traversed through the assessment process. It was necessary to keep track of how many revisions a display encountered in order to track of the time spent on each assessment. The types of errors that were discovered during an assessment were picture (also known as static), navigation, caution and warning, and telemetry.

Each type of error found in the CR falls into one of two cases. The first case is if the error found is from a change stated in the CR form, but not reflected in the attached requirements. This indicates the customer has not made the changes he or she wanted. Therefore, the requirement is incorrect. The second case is if there is a change in the requirement which is not stated in the CR form. This indicates that either the customer accidentally changed the wrong part of the requirement, or they neglected to state the change in the comments section of the CR. Categorizing the types of errors into these cases was important because this information was necessary in locating where an error was created.

The ETF was integrated with a time tracking program already used by the DDT, called “MyIssues”. MyIssues is used by the developer to keep track of assessment time.

Since the developer already utilizes this program, it made it easier for the developer to access and use the ETF. Figure 1 indicates how the ETF interface appears to the user.



Display:

Library Info:

CR:

Revision: 1/1

Time: □

| Error Type | Stated in CR | Not to be changed by CR. |
|---|---|---|
| # of PUI | □ | □ |
| # of Enum | □ | □ |
| # of Static | □ | □ |
| # of Nav | □ | □ |
| # of CW | □ | □ |
| # of Version | □ | □ |
| # of Comp | □ | □ |

*Figure 1*. Error Tracking Form (ETF) interface.

## Terminology

There are seven types of error for which an assessment can be returned. They are Program Unique Identifier (PUI), Enumeration (Enum), Static, Navigation (Nav), Caution and Warning (CW), Version, and Computation (Comp).

A PUI is like an mailing address. Data is sent to that address to control the object as a button or a text field. A PUI assessment error would mean that an object does not have an address attached to it. If this were to happen the data would go to the address, but since the address is not tied to the object, the object would not received the data.

An Enum is a list of text. A basic example would be a text field with two Enums. Enum one would be SIT and Enum two would be STAND. So if the data sent to the text field is 1, then the Enum "SIT" would be displayed in the text field. If the data sent is a 2, then the Enum "STAND" would be displayed. An object can have any number of Enums attached to it An Enum assessment error would be if the Enum does not have a number attached to it, or if the Enum it self was incorrect.

Static looks at the picture requirement provided along with the text requirement for a display. A Static assessment error is any issue with the picture requirement submitted with the requirements. The pictures shows how the display is to look, text fields, buttons, labels, etc. Each object on the picture must have a corresponding reference in the text requirement. The data in the text requirement is link to the picture using XY-coordinates. If an object that would display data, such as a text field, appears on the picture, then that object should have corresponding data in the text field to explain show what PUI, Nav, and CW it has. An error in the assessment would occur if there was no data in the text requirements. The opposite is also true. If there is data in the text requirement for an object, then that object should appear on the picture. Another Static error would be obvious graphical errors, such as overlapping objects.

A Nav is a the name of another display that should appear when a button is pressed. Each button on a display has a Nav tied to it. This tells the programmer where the button navigates to. A Nav error would be if the name given is incorrect or the display it is navigating to does not exist.

Caution and Warning (CW) is yet another piece of data tied to an object, usually a button. CW is a unique name tied to an object. If a problem were to happen with the system, data would be sent to that unique name and the button would go into alarm. A button goes into alarm by changing color, red for warning and yellow for alarm. A CW assessment error would be for example, if the wrong name was tied to an object.

The Version is the number given to the requirements submitted. Each time a requirement is submitted for a display, the version of the requirement should increase. For example, if the display "homepage" has requirement Version 1 and the customer wishes to update the display, then they must submit requirements Version 2. The CR used to submitted the requirements states the version number. A Version assessment error occurs if the number did not increment, or if the version number in the requirements does not match the version number stated in the CR.

A Computation (Comp) is a mathematical equation performed on the data submitted through the PUI. The raw data goes through the equation and the result is displayed on the laptop. A Comp error would be if the mathematical equation is incorrect.

When performing a assessment on display requirements, the errors explained above are separated into two categories, errors discovered in an object that had changes stated CR, and errors discovered for an object that did not stated any kind of change.

An example of "Stated" error would be if the CR states a particular change, but the requirement do not match this expected change. The change is stated but done incorrectly. This example would go into the "Stated" category.

An example of a “Not Stated” error would be if, during an assessment, a PUI was found to have changed in the requirements, but that change was not stated in the CR. This change would not be expected and would need to be addressed by the customer. This example would go into the “Not Stated” category.

## Instrument Pretest

The ETF was implemented before the data gather period. This allowed the developers to become proficient entering data, and also to allow time to fine-turn the ETF.

## Instrument Reliability and Validity

The program measured the time spent assessing CR and the types and amount of errors found during each CR revision, as well as the number of revisions the CR had before being complete and ready for development. It is important for this study that the developer enters the correct data into the ETF regarding the type and number of errors. The validity of the data received from the ETF can come into question if the developer does not enter the data correctly. Procedures were provided for the developer to follow in order to maintain correct data entry. Also, making the ETF simple and easy to use improve the data received from the developer.

## Treatment of Data and Procedures

The data gathered provided a descriptive data summary that was used to create charts. These charts revealed the most common errors, and can be used to study the process utilized by the customer in submitting a CR. Table 6 depicts to show how much each type of error cost in development hours. With this information, the processed used by the customer was examined to ascertain where and when these errors were created.

# CHAPTER IV

## RESULTS

Table 2 shows number of errors for each error type.

Table 2

*Total Errors by Error Type*

| Error Type | Number of Errors |
|---|---|
| PUI | 8 |
| Enum | 0 |
| Static | 132 |
| Nav | 10 |
| CW | 0 |
| Version | 2 |
| Comp | 8 |

Figure 2 shows the data Table 2 in the form of a pie chart.

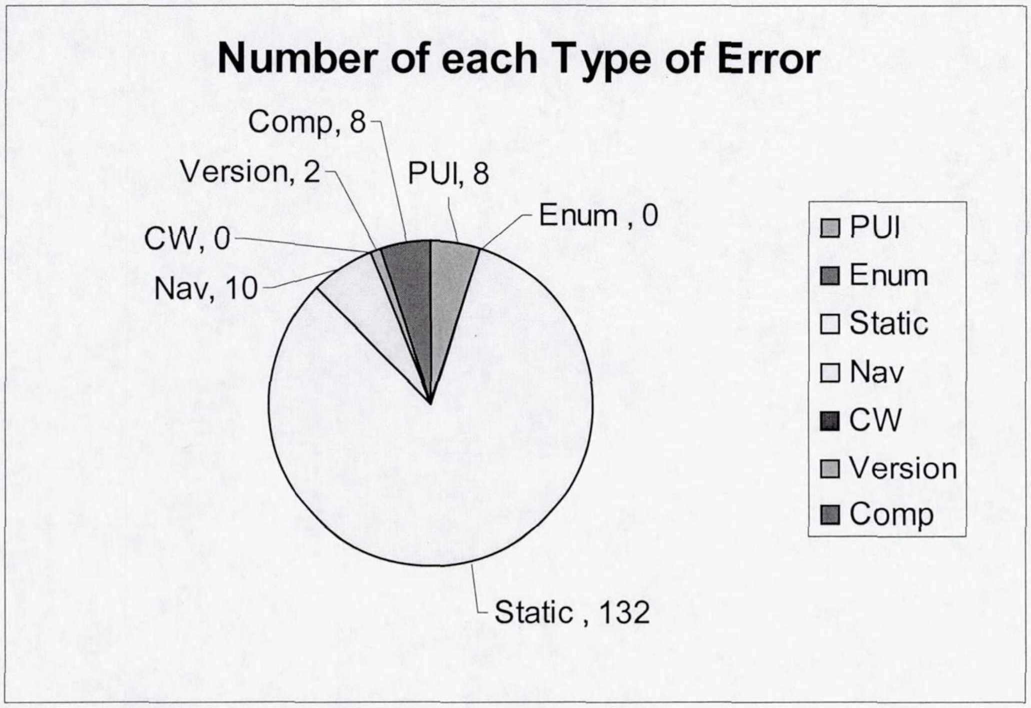


*Figure 2.* Number of each type of error.

Table 3 shows the number of "stated" errors for each error type.

Table 3

*Total "Stated" Errors by Error Type*

| Error Type | Number of Stated Errors |
|---|---|
| PUI | 4 |
| Enum | 0 |
| Static | 23 |
| Nav | 5 |
| CW | 0 |
| Version | 1 |
| Comp | 2 |

Figure 3 shows the data in Table 3 in the form of a pie chart.

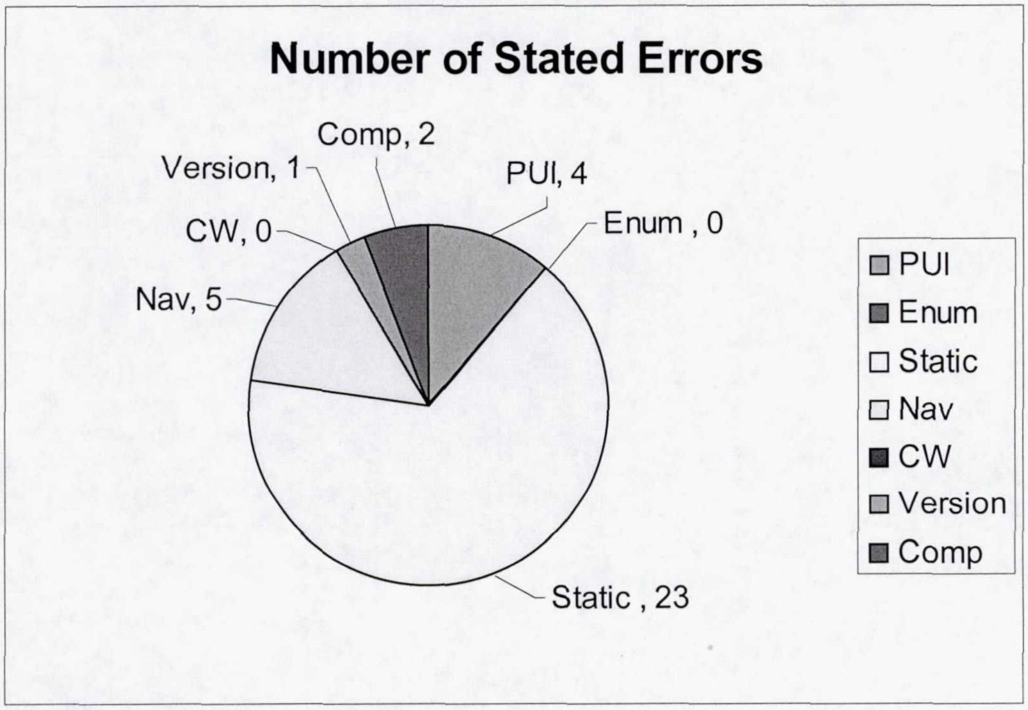


*Figure 3*. Number of "stated" errors.

Table 4 shows the number of "not stated" errors for each error type.

Table 4

*Total "Stated" Errors by Error Type*

| Error Type | Number of "Not Stated" Errors |
|---|---|
| PUI | 4 |
| Enum | 0 |
| Static | 109 |
| Nav | 5 |
| CW | 0 |
| Version | 1 |
| Comp | 6 |

Figure 4 shows the data in Table 4 in the form of a pie chart.

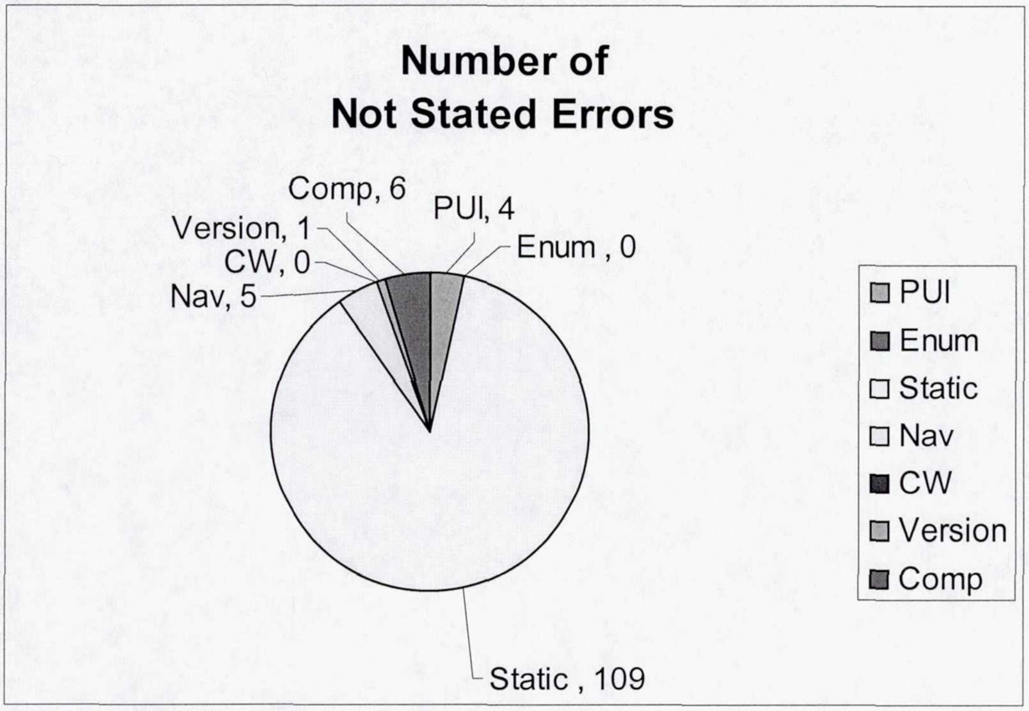


*Figure 4*. Number of "not stated" errors.

Table 5 shows the number of hours spent errors separated by version of assessment.

Table 5

*Total Time Spent on Each Version of Assessment*

| Assessment Version | Time Spent (in hrs) |
|---|---|
| First Assessment | 31.9 |
| Second Assessment | 1 |
| Final Assessment | 24.5 |

Figure 5 shows the data in Table 5 in the form of a pie chart.

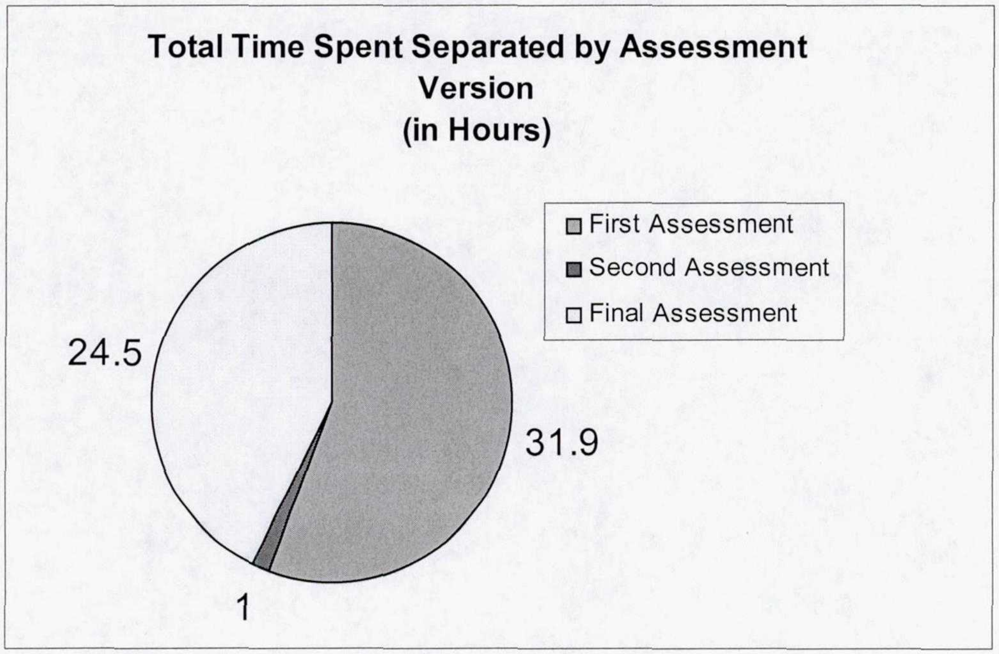


*Figure 5.* Total time spent separated by assessment version.

Table 6 shows the number of hours spent errors for each error type separated by version of assessment.

Table 6

*Total Time Spent Separated by Error Type*

| Error Type | First Assessment (in hrs) | Second Assessment (in hrs) | Final Assessment (in hrs) |
|---|---|---|---|
| PUI | 6.5 | 0 | 6 |
| Enum | 0 | 0 | 0 |
| Static | 11.3 | 1 | 8.4 |
| Nav | 2.6 | 0 | 1.8 |
| CW | 0 | 0 | 0 |
| Version | 0.5 | 0 | 2.5 |
| Comp | 11 | 0 | 5.8 |

Figure 6 shows the data in Table 6 in the form of a bar graph.

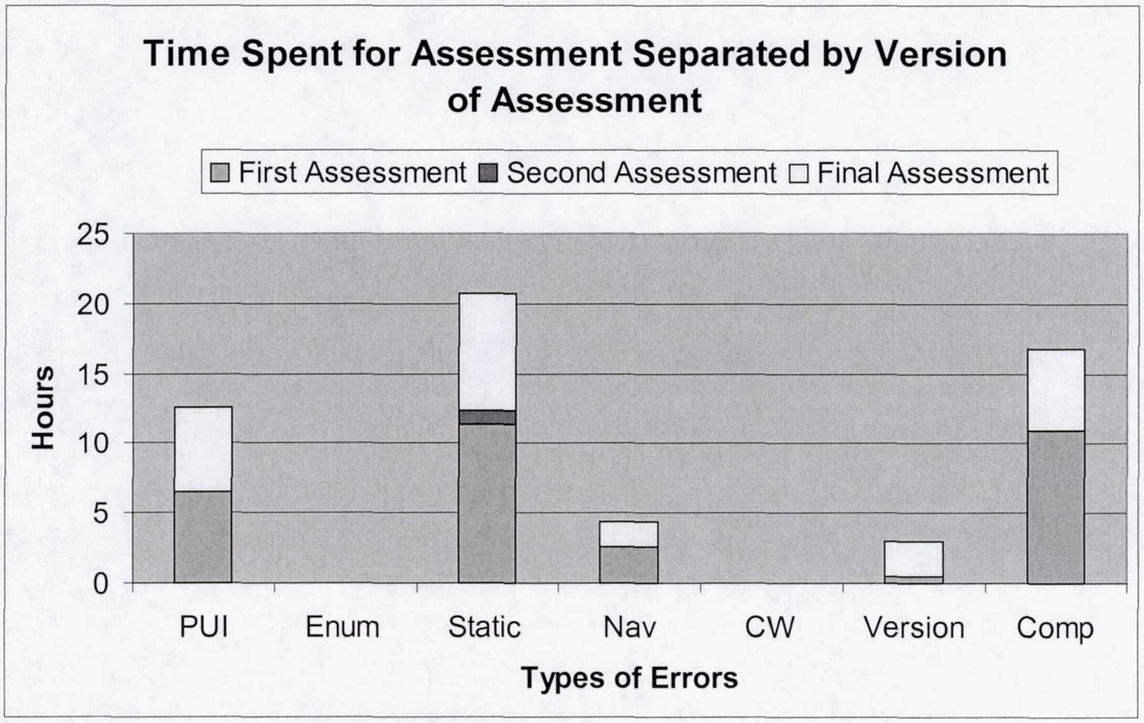


*Figure 6.* Time spent for assessment separated by version of assessment.

Table 7 shows the number of CR's assessed during the data gather period and how much total time was spent.

Table 7

*Total CR's Assessed During Data Gather Period*

| Number of CR's assessed | Total Time Spent (in hrs) |
|---|---|
| 286 | 163.95 |

Table 8 shows the number of CR's assessed during the data gather period that had to be sent back to the customer and how much total time was spent. The extra time needed to assess shows how much time was needed for the second and/or final assessments.

Table 8

*Total CR's Assessed With Errors During Data Gather Period*

| Number of CR's assessed with errors | Total Time Spent (in hrs) | Extra Time Needed to Assess (in hrs) |
|---|---|---|
| 23 | 54.7 | 25.5 |

# CHAPTER V

# DISCUSSION

Table 2 and Figure 2 show that most errors were located in the static error type. There are a number of reason a CR can have a static error. The first is because a CR must explain in detail any change to the display layout in the CR description. If a change is made that is not stated in the CR description, then the CR must be sent back so that the customer can either fix the display or add a description of the change to the CR. The second reason a static error may occur is that the CR described a change, but the picture attached to the CR did not match the description. Finally, the most likely reason a CR would have a static issue is if the customer removed an object from a display, but failed to removed the PUI attached to it in the text file. This would leave data in the text file that would point to an object on the picture that does not exist. This CR would be sent back to the customer to remove the left-over data in the text file.

The second most common error, coming in at a distant second to the number of static errors, is navigation errors. Navigation errors, like static errors, are cause by the CR failing to describe the change accurately. An example of this would be when no navigation was listed in the CR as being change, but when the CR is analyzed, navigation changes were found. The CR would have to be sent back to the customer to either change the navigation back to the previous version, or for the change to be added to the CR description. These types of discrepancies between the CR description, and the text file and pictures provided, is the major reason for a CR to be sent back to the customer. This is how the developer, with their limited knowledge on how the display system works, can recognized potential errors is the requirements.

Table 3 and Figure 3 show the number of errors that were caused when the CR description stated a change to an object, but that change to that object was different in the requirements. Static errors were found to be the most common, followed by navigation and then PUI's.

Table 4 and Figure 4 show the number of errors that were caused when an object changed in the requirements, but the CR description did not state any change to that object. Static errors were found to be the most common, followed by computation and then navigation.

Comparing Table 3 to Table 4 shows that Table 3 had the greatest number of errors for each type. The greatest jump can be seen in static errors. In Table 3 there were only 23 stated errors found, whereas in Table 4, there were 109 static errors. That indicates that 17%of static error found were a discrepancies between the description in the CR and the requirements, and that 83% of the errors were changes to the requirements that were not stated in the CR description.

Table 5 and Figure 5 show the total amount of time spent assessing each version of a CR. The first assessment was when the developer would first check the CR for errors. If an error is found during the assessment, then the CR is sent back to the customer to be fixed. The fixed version of the CR would then be sent back to the developer, who would in turn re-assess. If no errors were found during the second assessment, as was with the majority of the CR's in the data set, then that would be the final assessment. If errors were found again, then the CR would be return to the customer to be corrected. This process would continue till the CR had no errors. CR's were returned no more than three times during this study, as can be seen in Table 5. Of the time spent assessing CR's, 55.57% was spent on first assessment, followed by 1.74% for the second assessment, and 42.65% for the final assessment. This shows that even though the same assessment process was used for each version of the CR, most of the overhead was done in the first assessment.

Table 6 and Figure 6 show how time was spent for each assessment version separated out by error type. This provides a detailed picture of where time was spent for each error. The less about of time spent was in the second assessment. This was because most CR's did not require more than two assessments. For the majority of assessments, the second version was the final assessment. Table 6 shows that CR's with static errors took the longest to assess. This is followed by CR's with computation changes and then CR's with PUI changes.

Table 7 shows the number of CR's assessed during the data gather period as well as the total time spent. Comparing this data with Table 8 reveals that out of all the CR's assessed only 8.04% of the CR's assessed had errors. However, the 8% of assessments that had errors accounted for 15.55% of the total assessment time.

# CHAPTER VI

## CONCLUSIONS

After analyzing the data, a number of conclusions can be made from this study.

Static errors were the most frequent error type found during the assessment. This indicates that the customer may not be able to accurately match the text requirement to the picture requirement. Since the developer only has a limit knowledge of the system, it is interesting that they find as many errors as they do. It is interesting how errors, that are so simple to find by someone with limit knowledge of the system, can be submitted into CR's. It would seem logical that the person that developed the requirements would be more suited to assess the requirement.

The greatest amount of time spent of assessment that had errors was the first assessment. The first assessment accounted for 55.75% of the time required to assess CR's that contained errors. This indicates that the first assessment assumes most of the overhead of performing an assessment. The data also shows that CR's that had error took almost twice as long to assess.

The majority of errors found were changes made to the requirements that were not stated in the CR description. For this to occur there one of two things are occurring, either the customer is forgetting to state changes made to requirements in the CR description, or the customer is some how changing things in the requirements that are not intended. Since the developer's only knowledge of how the requirements are going to change is from the CR description, then it is important that the CR description be very detailed. This would inform the developer of exactly what to expect.

Removing 8.04% of errors found in CR's would save 15.55% time for doing assessments. The amount of errors found was small compared to the total number of assessment made, and the amount of extra time needed to re-assess was relatively small as well. This shows that number of CR's that had errors was not that large nor very costly to the organization.

# CHAPTER VII

## RECOMMENDATIONS

The problem with having the developer check the CR's for errors is that they have limited knowledge of the system. They only know what's written the requirements and CR description. If the change and the description of the change don not match, then the developer assumes that there is an error. This also means that if the change is described in the CR and is change, the developer would not know if the change was correct or not, only that a change as implemented.

It would be better if the customer had a system for checking errors internally before submitting the CR since they have a greater understating of how the system works. The first recommendation would be to have the customer run a more though analysis of there requirements before submitting them. Static error were the most common error type found. If the customer had a better system of comparing the static requirement to the text requirement, 82.5% of errors found could be removed, which would in turn save time.

Since the data gathering period began for the paper, a new system of developing displays has been implemented. This new system provides a tool the customer that allows them to development their displays by using a GUI interface. This system removes the need for the customer to submit requirement and as a result, the developer does not need to assess any requirements. Under the new system, the customer submits a completed display to the developer who only has to run a single program to verify there are no errors in the structure of the code used to create the display. Complier errors are code in the display, that break one or more of the rules used to show the display, and run correctly on the laptop. This new system has been a huge success in the area of removing errors found in assessment, not only because there are no longer requirements to assess, but because the customer has full control of the display. This allows them to create a display without having to explain the system to an outsider such as the developer. However, the customer needs to be more careful with the displays, than they were with the requirements submitted in the CR, since the developers are no longer checking them for logical errors, only functional errors.

# REFERENCES

Brown, A. R., & Sampson, W. A. (1973). *Program debugging: The prevention and cure of program errors.* London: Macdonald & Co. Ltd.

Dictionary.com (n.d.). Retrieved May 13, 2002, from Dictionary.com Web site: http://www.dictionary.com/search?q=process

Jarvis, A., & Hayes, L. (Eds., 1999). *Dare to be excellent: Case studies of software engineering practices that worked.* Upper Saddle River, NJ: Prentice Hall.

Mish, F. C. (Ed.). (1993). *Merriam-Webster's Collegiate Dictionary* (10th ed.). Springfield, MA: Merriam-Webster.

Robertson, D. W. (n.d.). *Common Display Development Team Change Request Assessment Process.* Manuscript submitted for publication.

Valentine, S. (2001). *Personal computer system display developer delivery procedures.* Houston, TX: Lyndon B. Johnson Space Center.

# APPENDIX A

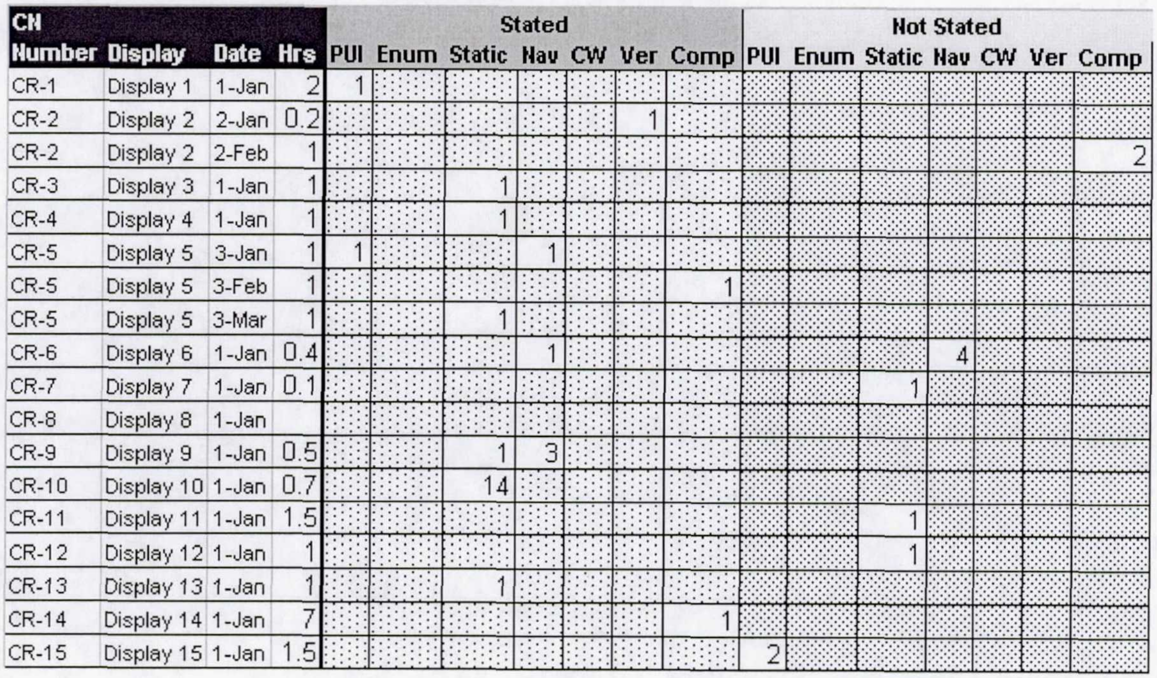
# BIBLIOGRAPHY

American Psychological Association. (2001). *Publication manual of the American Psychological Association* (5th ed.). Washington, DC: Author.

Rosado, A., Dammier, E., Clark, R., & Rosenhammer, F. (Eds.). (2001). *Guide to the graduate research project* (5th ed.). Daytona Beach, FL: Embry-Riddle Aeronautical University, Extended Campus.

# APPENDIX B

## DATA GATHERING INSTRUMENT or DATA SET



| CN | | | | Stated | | | | | | | Not Stated | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Number | Display | Date | Hrs | PUI | Enum | Static | Nav | CW | Ver | Comp | PUI | Enum | Static | Nav | CW | Ver | Comp |
| CR-1 | Display 1 | 1-Jan | 2 | 1 | | | | | | | | | | | | | |
| CR-2 | Display 2 | 2-Jan | 0.2 | | | | | | 1 | | | | | | | | |
| CR-2 | Display 2 | 2-Feb | 1 | | | | | | | | | | | | | | 2 |
| CR-3 | Display 3 | 1-Jan | 1 | | | 1 | | | | | | | | | | | |
| CR-4 | Display 4 | 1-Jan | 1 | | | 1 | | | | | | | | | | | |
| CR-5 | Display 5 | 3-Jan | 1 | 1 | | | 1 | | | | | | | | | | |
| CR-5 | Display 5 | 3-Feb | 1 | | | | | | | 1 | | | | | | | |
| CR-5 | Display 5 | 3-Mar | 1 | | | 1 | | | | | | | | | | | |
| CR-6 | Display 6 | 1-Jan | 0.4 | | | | 1 | | | | | | | 4 | | | |
| CR-7 | Display 7 | 1-Jan | 0.1 | | | | | | | | | | 1 | | | | |
| CR-8 | Display 8 | 1-Jan | | | | | | | | | | | | | | | |
| CR-9 | Display 9 | 1-Jan | 0.5 | | | 1 | 3 | | | | | | | | | | |
| CR-10 | Display 10 | 1-Jan | 0.7 | | | 14 | | | | | | | | | | | |
| CR-11 | Display 11 | 1-Jan | 1.5 | | | | | | | | | | 1 | | | | |
| CR-12 | Display 12 | 1-Jan | 1 | | | | | | | | | | 1 | | | | |
| CR-13 | Display 13 | 1-Jan | 1 | | | 1 | | | | | | | | | | | |
| CR-14 | Display 14 | 1-Jan | 7 | | | | | | | 1 | | | | | | | |
| CR-15 | Display 15 | 1-Jan | 1.5 | | | | | | | | 2 | | | | | | |