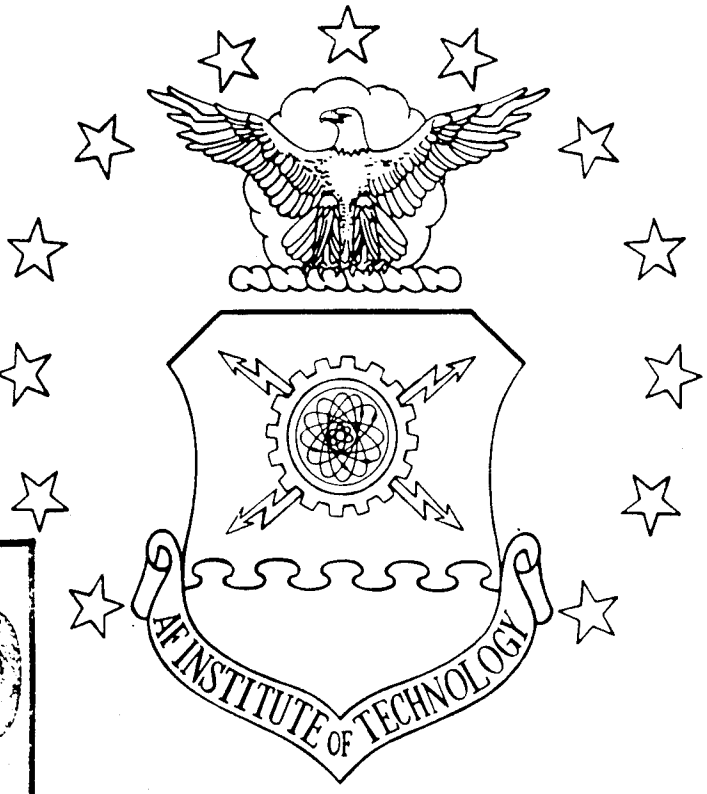





# A COMPARISON OF ERROR CATEGORIZATION SCHEMES FOR USE IN SOFTWARE SYSTEM SAFETY PROGRAMS

THESIS

Richard Escobedo, Captain, USAF
Jim Thomas, Captain, USAF

AFIT/GSS/LAR/94D-1

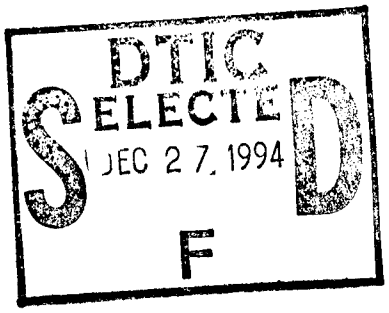




DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GSS/LAR/94D-1




# A COMPARISON OF ERROR CATEGORIZATION SCHEMES FOR USE IN SOFTWARE SYSTEM SAFETY PROGRAMS

THESIS

Richard Escobedo, Captain, USAF
Jim Thomas, Captain, USAF

AFIT/GSS/LAR/94D-1



Approved for public release; distribution unlimited

The opinions and conclusions in this paper are those of the authors and are not intended to represent the official position of the DOD, USAF, or any other government agency.

AFIT/GSS/LAR/94D-1

# A COMPARISON OF ERROR CATEGORIZATION SCHEMES FOR USE IN SOFTWARE SYSTEM SAFETY PROGRAMS

THESIS

Presented to the Faculty of the Graduate School of Logistics and Acquisition Management of The Air Force Institute of Technology AETC

In Partial Fulfillment of the Requirements for the Degree of Master of Science In Software Systems Management

Richard Escobedo, B.S.
Captain, USAF

Jim Thomas, B.S.
Captain, USAF

December 1994

# Preface

This study analyzes software error taxonomies for improving the development of safety critical software. A major goal was to identify or create an error categorization scheme that would be useful in the development of Department of Defense weapon systems. Although the proposed scheme is made up of elements of previous schemes within different domains, the resulting taxonomy has been deemed useful in DOD software development by the software experts surveyed.

Extensive work was performed to develop a telephone survey that would efficiently and concisely extract the opinions of software development experts. The use of an error taxonomy will be dependent upon an organization's software development process. Creating one for all DOD organizations is not our intention. Ideally, software development organizations would tailor the proposed scheme to fit their domain and defect prevention programs.

During our research, we received support from many people. Although we cannot list everyone who helped us, the contributions of some were so significant they demand special mention. We wish to thank Lieutenant Colonel Chris Arnold and Dr. Freda Stohrer, our advisors, for the time and patience they showed us as we developed a good idea into good research. We also want to thank Mr. Dan Reynolds, Professor of Statistics, for helping us identify the best statistical technique for our survey analysis. We also want to thank the many software experts who participated in the survey. Finally, we want to thank our wives, Hilda and Debbie, for their understanding, encouragement, and support—we will always be in their debt.

Richard Escobedo and Jim Thomas

# Table of Contents

Page

Page

# List of Figures

# List of Tables

# Abstract

Software safety is becoming increasingly important in the development of DOD advanced weapon systems. To make software safer, hazard conditions must be avoided along with the errors that accompany them. The first step in identifying errors is classifying error data. The area of software error classification is not as advanced as other software development areas. The technical literature lacks examples of comprehensive taxonomies that can be applied to various computer software domains and applications. The predominant approach is to organize errors into categories particular to the program currently in work. The typical error scheme is made of narrow categories that are not interrelated. Errors have been classified by symptom, by cause, by life-cycle phase, by severity, and by software control. The focus of this research was to determine the best way to classify errors in order to aid system safety in software development. The research identified common areas used in industry that aid in error classification. A telephone survey of experts in safety and software was used to obtain input on the most effective classification schemes. The research also proposed a taxonomy that will be ideal for DOD software development. Since software is becoming a larger part of advanced weapon systems, development of error-free and safe software to operate and support these weapon systems is increasingly important.

A COMPARISON OF ERROR CATEGORIZATION SCHEMES

FOR USE IN

SOFTWARE SYSTEM SAFETY PROGRAMS

# I. Introduction

## *General Issue*

The success of the United States military defense is linked to the technological superiority of its weapon systems. The technically advanced weapon systems employed in Operation Desert Storm significantly affected the outcome (AFP 63-115, 1993). To maintain technological superiority, the Department of Defense (DOD) is developing new weapon systems that exploit advances in computer technology. These new computer systems require software programs that have the responsiveness and the capacity to control time-critical devices or actual physical processes. This inevitable automation is becoming increasingly expensive, complex, and hazardous. Failures of these real-time mission-critical software systems can have devastating results as was evidenced by the discovery of an error in the timing system of the Patriot Missile after it was fielded for use during Operation Desert Storm. A software problem with the tracking system caused the computation of incorrect coordinates for incoming Scud missiles. The problem was traced to a clock error that grew with the total operating time of the tracking system. The eventual solution to this problem was to reset the clock error to zero after a short period

of operation. However, the solution was not implemented in time and the error may have been responsible for 28 deaths and 98 injuries (Wiener, 1993).

An analysis of the errors that contribute to the failure of a system will help prevent similar errors in future developments. As established above, the DOD is forced to rely heavily on computers to control critical decision making processes in weapon systems, and therefore must analyze past software errors to develop safe weapon systems.

Mishaps resulting directly from software control problems or errors in execution are documented in a cumulative index provided by Peter G. Neumann. These mishaps affect a wide range of areas including DOD weapon systems (Neumann, 1989). Two additional examples of software related mishaps involving DOD weapon systems are summarized below:

> A Navy F-18 fighter and crew nearly perished during a missile launch test. A wing mounted missile experienced a timing problem when the missile failed to separate from the wing after firing. Software was the main factor in the mishap, when it opened the clamp holding the missile to the wing, fired the missile, and then closed the clamp before the missile could leave the wing. The missile failed to develop enough thrust to leave the wing. The missile added an extra 3000 pounds of thrust to the wing and caused the aircraft to fall out of control. The pilot regained control after losing 20,000 feet of altitude. (Jorgens, 1988)

> A software error was suspected to be a contributing factor in the F-22 Advanced Tactical Fighter prototype crash at Edwards AFB, CA, during developmental test and evaluation. The flight control computer seemed unable to move the aircraft control surfaces fast enough to keep up with the pilot's commands. The pilot survived but the aircraft was severely damaged. The mishap is still under investigation. (Gellman, 1992)

## *Software System Safety*

Since software contributed to each of these DOD mishaps, the development of new weapon systems must address software system safety. That is, it must "...ensure that software executes within a system context without resulting in unacceptable risk" (Leveson, 1991).

The challenge to weapon system developers is to manage this risk. The risk of using a system must be balanced with the needs of the mission. The infinite number of error possibilities in a complex system makes ensuring absolute safety with large and complex computer-controlled systems impossible. It is the responsibility of the safety professional to formulate a reasonable, cost-effective plan that is consistent with a program's identified hazards and complexities. Due to limited resources, software system safety must focus only on those critical components and interfaces that have been identified to contain potential hazards (Piechota, 1992). Categorizing errors will help weapon system developers identify those areas that have caused problems in the past. Software developers can then apply their limited resources to those areas.

## *Specific Problem*

A comprehensive error categorization methodology will aid the DOD in implementing software safety. Categorizing the errors that contribute to mishaps will allow developers of software to concentrate limited resources to reduce these errors. This thesis identifies and evaluates error categorization schemes that can aid future software development associated with complex weapon systems.

## *Investigative Questions*

To solve the specific problem outlined in the last section, we must answer the following investigative questions:

1. What are the different methods for categorizing software errors?
2. What are the candidate error categorization schemes for software safety?
3. Which error categorization schemes are beneficial for software system safety?
4. Are different types of schemes preferable for different disciplines: for example, information systems, embedded systems or data bases?

## *Overview*

This thesis includes five chapters. Chapter I introduces our research area by stating the problem and the investigative questions that guided our efforts. Chapter II reviews the literature pertaining to software error classification. It presents several schemes that may be useful in categorizing errors as part of a software safety program during software development, including a scheme proposed by the authors. Chapter III describes the methodology used to answer the investigative questions. Chapter IV discusses the research findings and data analysis. Chapter V states the research recommendations. The appendix contains the telephone survey used to obtain the input from the software experts.

# II. Error Classification

## *Introduction*

An error classification scheme is a taxonomy that is used to categorize errors that occur during software development. Classification schemes are used to gather data on software errors in an effort to prevent them. Organizations have used different classification criteria in the construction of schemes. Therefore, deciding on the appropriate classification criteria for an organization is very important. Typical classification criteria are reviewed. Some existing software error classification schemes are presented for comparison. Finally, a proposed software error classification scheme is discussed.

## *Classification Criteria*

A comprehensive study of error classification schemes was performed by Collofello and Blumer in 1983. They identified several types of classification schemes which are useful in defect prevention and causal analysis. Errors have been classified by symptom, by cause, by life-cycle phase, by severity, and by software control. Most often a combination of types are used in a classification scheme (Collofello, 1983).

### Classification by Symptom.

The first attempts to categorize errors considered symptoms only. These classifications grouped errors according to their effects on the system and according to a general description of the error. This information is easily obtained from defect reports

during testing. The most popular classification by symptom scheme is by Endres. He produced a very detailed classification scheme in 1975. Based on internal testing of an operating system, the categories selected were very specific; for example, wrong register reference or incorrect resource allocation (Endres, 1975). Although Endres's scheme covered virtually all possible error symptoms for an operating system, the scheme falls short when used on a different project type, hardware system, or programming language. Essentially, the symptoms observed in these domains are vastly different. At the other end of the spectrum, broader symptom classification schemes were developed. These include Maxwell's error categories scheme and Lipow's software failures scheme (e.g., logic, data handling, and interface). While generalizations about the effectiveness of different error detection techniques can be made with these schemes, the biggest benefit is their application to various software domains (Maxwell, 1979; Lipow, 1979).

Beizer presents "The Taxonomy of Bugs" in his book, Software Testing Techniques. Four broad categories are used to classify software "bugs" (function bugs, system bugs, data bugs, and code errors) and methods to detect and prevent them are discussed. These remedies include formal specification languages, design methodologies, and sound documentation (Beizer, 1983). Ostrand and Weyuker developed a new classification approach called an attribute categorization scheme. In this scheme, errors are not assigned to a single category. The attributes of the error are captured in four interpretive areas that describe more fully the characteristics of the error. The areas are major category (e.g., data, decision, or system), type (e.g., address, loop, or branch), presence (how the fault was corrected), and use (operation being performed) (Ostrand, 1984). This symptomatic scheme does provide more insight into the errors than the other symptomatic schemes. However, the symptomatic schemes in general lack information on how to prevent the identified errors. More information on what should have been done to prevent these errors is needed (Buckland, 1982; Collofello, 1985).

Classification by Cause.

Causative error classification schemes provide this missing information. Processes, techniques, and tools can be evaluated more effectively with data from these types of schemes (Collofello, 1985). Basili classified error causes, not categories of symptoms. His causes include incorrect or misinterpreted requirements, incorrect or misinterpreted functional specifications, design errors (involving one or more than one component), misunderstanding of external environment, error in use of programming language or compiler, clerical error, or error due to previous miscorrection of an error. In his application of this error categorization scheme to the development of a medium scale satellite software project, Basili noted that a large portion of the errors were due to a misunderstanding of specifications or requirements (Basili, 1982). Jones asserts that the purpose of categorizing errors is to prevent their occurrence and that elaborate lists of categories are not required to do this. His scheme contains only four causes for any error: communication, education, oversight, and transcription. Jones proposes a process improvement methodology for performing causal analysis as part of the programming process. His general categories of root causes lead analysts directly to the software quality area of process improvement solutions (Jones, 1985). While the benefits to an error scheme based on cause are apparent, it must be noted that the difficulty with such a scheme is the effort required to determine the right cause or causes of an error.

Classification by Life-Cycle Phase.

Many error data collection processes consider the project phase, but not many include it directly in their classification scheme. Dunn proposed a classification scheme built around the phase of the software development life-cycle in which the error was produced. This information is key to the effectiveness of ongoing process improvement initiatives. Organizations have to know where in the software development life-cycle errors are prone to occur to improve the process. The categorization scheme has four

major time frames: definition of requirements, design phase, coding phase, documentation and installation. Each time frame has many subcategories (Dunn, 1987).

Classification by Severity.

The final type of scheme is based on the severity of the error. The severity may be defined by the time and cost to correct the error as well as by the effect on the software. Davis and Gantenbein recognized the value of severity classification schemes in the development of safety critical systems. Their paper describing techniques that can be used to design fault-tolerant software classifies errors three ways. *Internal* errors can be handled by the system where they were produced. *External* errors cannot be handled internally but the effects remain in that system. Finally, *pervasive* errors cause errors in other systems. The pervasive category is the most damaging. The severity classification of an error provides a measure of importance it has to the system (Davis, 1992).

Classification by Software Control.

Another indicator of the impact a software error has on a system is the amount of autonomous control the software has over the system. The Department of Defense is developing a matrix that has two main categories pertaining to the type of control the error affects and severity of the error. This scheme is taken from the US Army Communications-Electronics Command Software System Safety Guide and is representative of other military categorization schemes. The control category consists of autonomous/time critical, autonomous/not time critical, information/time critical, operator control, and information decision algorithm control levels. The severity category is made up of catastrophic, critical, marginal, and negligible severity levels (Russo, 1992). This classification scheme attempts not only to capture the severity of an error but also its criticality to the system.

Combination Classifications.

Many developers of error classification schemes noted the importance of considering more than one type of classification; however, only a few have succeeded at combining two or more types in their categorization scheme. One of the earliest attempts was in 1976, when the Rome Air Development Center (RADC) sponsored a study of software reliability. The RADC successfully combined a detailed symptomatic scheme with causal categories and produced 164 error categories in 20 major classifications. This work was based on five software projects (Thayer, 1976). The symptomatic features of the scheme allow for easy data collection, while the causal components provide useful diagnostic insight. The error scheme is tailorable; however, it loses validity outside its project domain. Critics of the RADC study claim that the other classification types must be considered (Buckland, 1982; Collofello, 1985).

In the 1980s, two schemes were published that moved away from the symptomatic classifications and combined causal categories with other classification types. Buckland devised a three-dimensional error taxonomy for the purpose of statistical trend analysis of error data. The three dimensions are error category (cause), time of occurrence (life-cycle phase), and criticality (severity). The error categories defined by Buckland were similar to symptomatic categories (e.g., computational, logic, and interface); however, these error categories were broken down further by the research team to their root causes. Four time periods are used in this taxonomy: development, verification (integration), acceptance (formal testing), and transfer (operational use). Criticality was divided into three levels: A--critical error, B--dangerous situation, and C--minor problem. One of the major findings of her study of a space software application was that classification of errors should occur during development to be most effective. In her opinion many of the critical errors were produced in the requirements and design phases (Buckland, 1982). In 1985, Collofello proposed a two-dimensional error classification scheme. He concentrated on

the life-cycle phase in which the error was introduced and the cause of the error. The scheme was designed to apply to any development organization for comparison, regardless of the development process or activities used. The broad life-cycle phases in the scheme include requirements specification, high-level design, detailed design and implementation, and modifications. These were cross-referenced in a matrix against the error causes: communicational, conceptual, and clerical. Individual categories, specific to the application, were also provided in each matrix cell. Collofello adhered to five data collection principles while developing his scheme: 1) the data collection must be nonobtrusive, 2) the scope of the data collection must be large (collect as much as you can), 3) data must not be specific to one project, 4) both long- and short-term benefits for the organization must be maximized, and 5) the new data should complement previous efforts (Collofello, 1985).

More recent error classification schemes expand on the cause-effect relationships of errors in software development. Nakajo and Kume use causal error classification schemes to identify intermediate cause-effect relationships between types of human errors and the ultimate system failure. They recognize that most classification schemes produce either an originating cause or a final result (system failure). Nakajo and Kume are after the intervening "work system flaws" that contribute to the error. They suggest these flaws occur in the development process, in the individual programmer's work, or in the engineering environment. Identifying these cause-effect relationships are very important for long-term process improvement of software development (Nakajo, 1991).

The previous classification schemes were limited to new development efforts. Collofello expanded his 1985 error classification scheme to include follow-on support activities. In his new scheme, the ultimate goal for determining the cause of an error is process improvement. Collofello encourages using the scheme to develop cost-effective recommendations for elimination of the long-term causes of errors. The maintenance

causal categories include: system knowledge/experience, communication, software impacts, methods/standards, feature deployment, supporting tools, and human error. This scheme was applied to a modification of a large telephone system. One conclusion from the study is that almost 80 percent of all errors created during the modification activities were caused by insufficient knowledge/experience, communication problems, or software modification impacts (Collofello, 1993).

## *Relevant Schemes*

To determine the best classification scheme for use in software safety programs, we chose three published error taxonomies, as well as our own, for comparison. The comparison was accomplished through a telephone survey of software safety development experts. The three published schemes are provided by Collofello, Jones, and Russo and they are labeled Scheme A, Scheme B, and Scheme D, respectively. Our proposed scheme, Scheme C, is discussed in the subsequent section.

### Scheme A.

Collofello proposed this two-dimensional classification scheme in 1985. He developed a scheme that would apply to any development organization, regardless of the development process or activities used. Scheme A attempts to capture error data in terms of the cause of the error and the development phase in which it occurred. Three major causes are listed: Communicational, Conceptual, and Clerical. Communicational causes are breakdowns in communication among team members. Conceptual causes are difficulties in analyzing the problem and synthesizing a solution. Clerical causes are oversights or simple transcription problems. Software development is divided into four life-cycle phases: Requirements, High-Level Design, Detailed-Design and Coding, and Debugging and Maintenance (Collofello, 1985).

**Table 1. Collofello's Error Classification Scheme**

| | Requirements | High-Level Design | Detailed-Design and Coding | Debugging and Maintenance |
|---|---|---|---|---|
| Communicational | | | | |
| Conceptual | | | | |
| Clerical | | | | |

Scheme B.

Jones believed that large lists of causes were not necessary to prevent defects. He proposed a process improvement methodology for performing causal analysis. This methodology includes a causal scheme with only four error causes. The four major cause categories are Communications, Education, Oversight, and Transcription. Communications errors are errors due to breakdowns in lines of communication between team members. Education errors result from the software developer's failure in understanding or training. Oversight problems occur when all possibilities are not considered. Finally, transcription errors are simple clerical errors (Jones, 1985).

Jones's scheme differs from Collofello's scheme because it concentrates on only one classification category: cause. Jones's process improvement methodology is designed to use the error data for more than just defect prevention. Jones wants developers to deal with the real causes of errors and not overburden themselves with all the other details of the error. Therefore, corrective action is not limited to fixing the code. In addition, the flaws in the software process must be fixed.

**Table 2. Jones's Error Categorization Scheme**

| | |
|---|---|
| Communications | |
| Education | |
| Oversight | |
| Transcription | |

Scheme D.

Unlike the previous two schemes, Scheme D does not classify errors according to cause. Scheme D classifies error data according to the control that the software module has over the system and the effect of the error on the system. The software control is characterized by the independence of the module in the system and the real-time execution. Autonomous Time Critical refers to software exercising autonomous control over potentially hazardous hardware systems, subsystems, or components without the possibility of real time human intervention to preclude the occurrence of a hazard. Autonomous Not Time Critical refers to software exercising autonomous control over potentially hazardous hardware systems, subsystems, or components allowing time for human intervention by independent safety systems to mitigate the hazard. Information Time Critical refers to a software item displaying information requiring immediate operator action to mitigate a hazard. Operator Control refers to software items issuing commands over potentially hazardous hardware systems, subsystems, or components requiring human action to complete the control function. Information Decision Algorithm refers to software generating information of a safety critical nature used to make safety critical decisions.

The effects of the error on the system are divided into four major categories: Catastrophic, Critical, Marginal, and Negligible. Catastrophic errors result in system loss or life loss. Critical errors result in major system damage or severe injury. Marginal errors result in minor system damage or minor injury. Negligible errors result in less than minor system damage or less than subsystem loss.

**Table 3. Russo's Error Categorization Scheme**

| | Catastrophic | Critical | Marginal | Negligible |
|---|---|---|---|---|
| Autonomous Time Critical | | | | |
| Autonomous Not Time Critical | | | | |
| Information Time Critical | | | | |
| Operator Control | | | | |
| Information Decision Algorithm | | | | |

*Proposed Scheme C*

Scheme C is proposed by the authors to aid in the development of DOD weapon systems. The literature agrees on several points about error classification schemes in general:

- The purpose of categorizing errors is to aid in their detection and prevention.
- The data required for the scheme must be easily obtained.
- Not all schemes apply outside the project domain that they were based on.
- A consistent and comprehensive error data collection methodology must be implemented for maximum benefits (Basili, 1982; Beizer, 1983; Collofello, 1985 and 1993; Jones, 1985; Ostrand, 1984).

To devise a classification scheme for software development in the DOD, the characteristics of the previous effective schemes must be applied; in addition the severity of the error must be considered. Therefore, our error taxonomy will include the following dimensions:

- Error Cause
- Life-Cycle Phase
- Error Severity

Unlike Buckland's error category scheme where the same three dimensions (cause, life-cycle phase, and severity) were considered separately, our taxonomy will combine the dimensions in a matrix similar to Collofello's two dimensional matrix of cause and life-cycle phase (Buckland, 1982; Collofello, 1985).

Scheme C attempts to capture error data in terms of the cause of the error, the development phase in which it occurred, and the severity of the error. The major causes are Incorrect Requirements, Communications, Oversight, Interface, Incorrect Computations, and Transcription. Incorrect Requirements occur when the specification, understanding, or applicability of the requirements is insufficient. Communications errors are breakdown in the flow of information between team members. Oversight errors occur when all possible cases or conditions are not considered or handled. Interface errors can be anomalies in communication or interaction between systems or subunits. Incorrect computations are incorrect formulations of equations or functions used by the system. Transcription errors are clerical errors. The development effort is divided into four life-cycle phases: Requirements, Design, Coding, and Support. Severity is indicated by adding the code provided for each level of the errors effect on the system: Catastrophic, Critical, Marginal, and Negligible.

**Table 4. Proposed Error Categorization Scheme**

| | Requirements | Design | Coding | Support |
|---|---|---|---|---|
| Incorrect Rqmts | | | | |
| Communications | | | | |
| Oversight | | | | |
| Interface | | | | |
| Incrt Computns | | | | |
| Transcription | | | | |

Note: Indicate the severity of the error by code:

| | | | | | |
|---|---|---|---|---|---|
| C | - | Catastrophic | R | - | Critical |
| M | - | Marginal | N | - | Negligible |

# III. Methodology

## *Introduction*

This chapter describes the methodology we used to evaluate the four error categorization schemes identified in the previous section. All four of our investigative questions are discussed in this chapter. To refresh the reader's memory, the four questions are:

1. What are the different methods for categorizing software errors?
2. What are the candidate error categorization schemes for software safety?
3. Which error categorization schemes are beneficial for software system safety?
4. Are different types of schemes preferable for different disciplines: for example, information systems, embedded systems or data bases?

The first section of this chapter addresses the population from which our survey data was collected, the next section addresses the data collection process, and the final section addresses the data analysis process.

## *Population*

The target population for our research was software system safety experts within the DOD. There are several reasons for a small target population. First, the concept of software system safety appeared in the literature in the mid 1980's and thus is a relatively

new concept (Leveson, 1986). Second, as of 1992, over 96 percent of the Aeronautical System Center's System Safety Managers (SSMs) were not fully qualified in software system safety and few SSMs understand software safety analysis (Colan and Prouhet, 1992). These facts led us to believe that there are few software system experts within the DOD. To identify DOD software system safety experts, we contacted the safety headquarters of the Army, Air Force, and Navy. After working down many organizational levels, we sent our survey to 12 software system safety experts who were affiliated with the DOD. One DOD, two Army, three Navy, and four Air Force software system safety experts provided data via the survey discussed later in this chapter. The remaining two experts did not respond to the survey.

## *Data Collection*

The data collection for this research encompassed all four of our investigative questions. Table 5 shows a matrix of investigative questions and data collection methods. The first two investigative questions were satisfied by the literature review; the results were reported in Chapter II. Data for the third and fourth investigative questions were collected via a telephone survey of software system safety experts; the results are reported in Chapter IV.

**Table 5. Investigative Question Methodology Matrix**

| Investigative Question | Literature Review | Telephone Survey |
|---|---|---|
| 1 | X | |
| 2 | X | |
| 3 | | X |
| 4 | | X |

Literature Review.

We reviewed both DOD and civilian literature to determine the different ways of categorizing software errors. Our research identified three candidate software safety categorization schemes. We then devised a fourth candidate scheme based upon our review of the literature.

Telephone Survey.

The telephone survey is one of the quickest and most economical approaches to reach individuals (Emory, 1991). The telephone survey was used to determine the following about the candidate categorization schemes identified in the previous chapter:

1) How knowing the classification of errors relative to a scheme would lead to the development of safer systems.
2) How implementing a scheme would affect an organization's software development process.
3) How the costs associated with implementing a scheme vary.
4) Whether a common scheme is best applicable to different software platforms and/or applications.

Interviews were conducted using the set of predetermined questions located in Appendix A. The schemes and questions were provided to the respondent prior to the interview. This review of the candidate schemes and questions prior to the interview allowed respondents to become familiar with the candidate schemes, thus enabling the experts to formulate their opinions and recommendations about the proposed categorization schemes, and thereby minimizing the time respondents spent on the phone.

## *Survey Development*

Our survey was designed to obtain three different types of information and was therefore divided into three sections. First, the attitudes of those surveyed towards the candidate error categorization schemes were obtained. The majority of the questions in the first section consisted of closed-ended questions asking respondents to rank the schemes based upon various criteria. The remainder of the questions in the first section were open-ended to allow the respondent to discuss/identify the strengths and weaknesses of each individual scheme as well as identify whether data is currently being collected that could be used in each scheme.

The second type of information, and thus the second section of the survey, was used for classification and analysis of the data obtained from the survey. These questions pertained to the education and experience of those surveyed. This education and experience data allowed additional insight into the applicability of the schemes to different software domains and applications.

The third type of information involved data pertaining to the administration of the survey. This administrative information included the interviewer, respondent, date, time of the interview, and anonymity.

## *Data Analysis*

The survey provided two different types of data for analysis. The first type is the ranking data provided by questions 1 through 11. The second type is the open-ended data inputs provided by questions 12 through 18. This section discusses the different methods used to analyze both the ranking data and the open-ended inputs received from the respondents.

Ranking Data.

Each ranking question in the survey provided a set of data about the four proposed categorization schemes based upon different criteria. For each different criterion, we were interested in determining the experts' order of preference for the schemes. The schemes were ordered by the sum of the experts' rankings. For example, Table 6 shows the ranking by nine hypothetical experts:

**Table 6. Sample Ranking Data**

| | | Scheme | | | | |
|---|---|---|---|---|---|---|
| | | A | B | C | D | Sum |
| Expert | 1 | 4 | 1 | 2 | 3 | 10 |
| | 2 | 3 | 2 | 1 | 4 | 10 |
| | 3 | 3 | 1 | 4 | 2 | 10 |
| | 4 | 3 | 2 | 1 | 4 | 10 |
| | 5 | 4 | 3 | 1 | 2 | 10 |
| | 6 | 3 | 1 | 4 | 2 | 10 |
| | 7 | 4 | 2 | 1 | 3 | 10 |
| | 8 | 4 | 1 | 2 | 3 | 10 |
| | 9 | 3 | 2 | 4 | 1 | 10 |
| | Total | 31 | 15 | 20 | 24 | 90 |

The column totals show that the order of preference is scheme B, C, D, A. What we cannot easily interpret from these totals is if the ranking represents a consensus among the experts. To help interpret the agreement between the rankings, we used the

Kendall Coefficient of Concordance (Gibbons, 1976). This coefficient is a relative measure represented by a ratio of two different sums of squares. The Kendall Coefficient of Concordance can range between the values zero and one, with a value of one representing perfect agreement among rankings. As the value of the coefficient decreases, the strength of agreement between rankings decreases as well. According to Gibbons, the following equation is the simplest method for calculating the Kendall Coefficient, represented by the letter W.

$$W = \frac{12\sum_{j=1}^{n} R_j^2 - 3k^2 n(n+1)^2}{n\ k^2(n^2-1)}$$

where R represents the column totals
k represents the number of rankings
n represents the number of objects being ranked

Applied to our previous example:

$$W = \frac{12\sum_{j=1}^{4} R_j^2 - 3(9)^2 4(4+1)^2}{4\ (9)^2(4^2-1)} \quad \text{reduces to } W = 0.338$$

A coefficient value of 0.338 may seem so low as to indicate that there is no consensus. However, the sample size affects the significance of the coefficient value greatly. In order to determine if the coefficient value is significant, we conducted a test using the Q test statistic. According to Gibbons, the Q statistic is most effective for tests with a large sample size (large number of experts). The tables in Gibbons's book for use with the Kendall Coefficient go up to a sample size of eight, with the caveat that values for larger

sample sizes can be estimated using the chi-square distribution. The following formula is used to calculate Q:

$$Q = k(n-1)W$$

applied to our previous example:

$$Q = 9(3)0.338 \text{ reduces to } Q = 9.133$$

Two hypotheses will complete our test. The null hypothesis is that no significant agreement exists between the different rankings. The alternate hypothesis would thus be that a significant agreement exists. We will use an alpha level of significance of 0.05 for our test. The chi-square distribution with n-1 degrees of freedom is the appropriate distribution for a test using the Q statistic. We calculate a P-value in our test, which is defined to be the area under the chi-square distribution to the right of our calculated Q statistic. Our decision to accept or reject the null hypothesis depends upon the P-value and its relation to the chosen alpha level. If our P-value is greater than the alpha level, then we accept the null hypothesis and conclude that our Q statistic is from the same distribution as the null hypothesis. If our P-value is less than the alpha level, then we reject our null hypothesis and conclude our Q statistic is from a different distribution than our null hypothesis. The summary of the test involving our sample ranking data follows:

Null Hypothesis: No significant agreement exists between the rankings

Alternate Hypothesis: A significant agreement exists between the rankings

Distribution: Chi-Square with 4 - 1 = 3 degrees of freedom

Alpha level: .05

$Q = 9.133$ $P(Q \geq 9.133) = .028$

Test Conclusion:

Because our P-value is less than our alpha level of .05 then we reject the null hypothesis and conclude that a significant association exists between the rankings in our example. Thus, a Kendall Coefficient of Concordance value of 0.338 is significant and our inference that the order of preference B, C, D, A represents a consensus among the experts is supported. Our test is summarized graphically in Figure 1 with the shaded region representing the rejection region and the vertical dotted line representing Q.

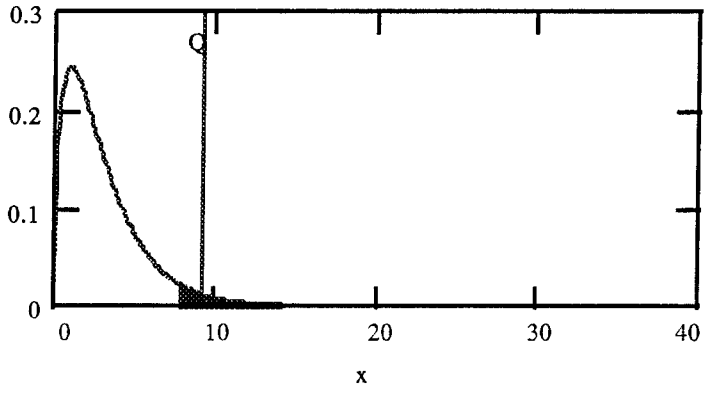



**Figure 1. Decision Graph for Sample Ranking Data**

We repeated the above analysis for questions 1 through 11. We then conducted a summary test to establish an overall order of preference for the schemes evaluated in our survey. For the questions that resulted in a significant association for the rankings, we used the order of preference from that question to come up with a set of rankings using the questions as the evaluators for the schemes. We then conducted a Q test to see if there was a significant association between the rankings provided by each individual question. The analysis of the open-ended data questions depended upon the outcome of the summary Q test.

Open-Ended Data.

The open-ended data provided by questions 12 through 14 was used to determine some of the advantages and disadvantages of using the schemes assessed in our survey. The analysis of the open-ended data provided by questions 15 through eighteen was driven by the result of the summary Q test. Because the summary Q test resulted in a significant association between the rankings provided by the questions, the information was used to assess the applicability of the schemes to different software platforms.

## *Summary*

This chapter identified the target population for our survey as software system safety experts affiliated with the DOD. Because software system safety issues have only recently been addressed in the literature, our target population was small. We described the purposes of the literature review and telephone survey, the two data collection methods used to satisfy our four investigative questions. A discussion of survey development and the two types of data obtained from the survey was completed. This chapter concluded with a discussion of the data analysis plan, including how the Kendall

Coefficient of Concordance was used to analyze the ranking data. The next chapter presents the results of our data analysis and relate those results to each investigative question.

# IV. Findings and Analysis

## *Overview*

This chapter details the findings of the research and analyzes the survey data used to support these findings. The chapter discusses the investigative questions, restating each investigative question, and presenting the findings. The chapter concludes with a discussion of general findings and a summary. A complete analysis of each ranking question in the survey is contained in Appendix B.

## *Investigative Questions*

### Investigative Question 1: What are the different methods for categorizing software errors?

The purpose of the first investigative question was to determine the current methods used to categorize software errors. By identifying the various ways software errors are categorized, we were able to study the best methods and to gain insight into their success. As discussed in the literature review, Chapter II, there are many ways to categorize software errors. These types of classification schemes, which are useful in defect prevention and causal analysis, are classification by symptom, by cause, by life-cycle phase, by severity and by software control. Most recently a combination of types has been used in classification schemes.

Investigative Question 2: What are the candidate categorization schemes for software safety?

Investigative question two sought to identify actual categorization schemes from the classification types listed under the first investigative question. These schemes were limited to a representative sample potentially applicable to software safety. They were then examined by software safety experts.

In Chapter II, we presented four candidate categorization schemes for software safety. The four schemes were included in the survey in Appendix A. All but one of the candidate schemes combine two or more of classification types identified above in investigative question one. Combination schemes were chosen as candidate schemes because they provide more information on software errors and reflect the inherent complexity of software errors. Table 7 shows which classification types were included in each scheme. All classification types were represented in the candidate schemes except for classification by symptom.

**Table 7. Classification Types Represented In Survey**

| | Scheme A | Scheme B | Scheme C | Scheme D |
|---|---|---|---|---|
| Symptom | | | | |
| Cause | X | X | X | |
| Life-cycle Phase | X | | X | |
| Severity | | | X | X |
| Software Control | | | | X |

Investigative Question 3: Which categorization schemes are beneficial for software system safety?

Investigative question three attempted to evaluate the candidate schemes and determine the ones that apply to software safety. As discussed in Chapter III, the survey questions were developed to answer this question by obtaining information in four different areas. These areas are restated below with reference to the survey questions that were designed to obtain the information:

1) How knowing the classification of errors relative to a scheme would lead to the development of safer systems. (Survey Questions 1, 2, 3, and 4)
2) How implementing a scheme would affect an organization's software development process. (Survey Questions 5, 6, and 9)
3) How the costs associated with implementing a scheme vary. (Survey Questions 7, 8, and 10)
4) Whether a common scheme is best applicable to different software platforms and/or applications. (Survey Questions 11, 15, 17, and 18)

Table 8 summarizes all the survey question results by providing the relative rankings of the candidate schemes. Only results that demonstrate agreement among the experts are included.

**Table 8. Summary of Significant Scheme Ranking Results By Question**

| Rank | SQ 2 | SQ 3 | SQ 4 | SQ 5 | SQ 9 | SQ 10 | SQ 11 | Sum Q |
|---|---|---|---|---|---|---|---|---|
| 1st | C & D | C | C | C | C | C | C | C |
| 2nd | | A | A | A & D | A | D | D | A & D |
| 3rd | A & B | D | D | | B & D | A | A | |
| 4th | | B | B | B | | B | B | B |

Note: SQ # refers to survey question #

The first question in the survey (see Appendix A) presents the types of classification schemes, and asks the experts to rank them in order of usefulness to software safety. Our analysis showed strong agreement among the experts, with a P-Value of 0.007. They ranked severity first, which is understandable from a software safety perspective. The errors with the highest severity potential must be given the most attention. Cause and software control were tied for second. These two types of classification schemes appear to have equal importance to the experts. The type of classification that ranked fourth was symptom, which beat out last-ranked life-cycle phase. This ranking was surprising inasmuch as literature review (page 6) indicates that current software error categorization methods depart from the traditional practice of noting symptoms and are moving to methodologies that consider life-cycle phase.

Survey question two addressed error collection. With a P-Value of 0.00022, the rankings provided by the experts were significant. Schemes C and D tied for first, with schemes A and B tied for third. Both schemes C and D capture the severity of an error, which the experts deemed very important.

Survey question three addressed error correction. A P-Value of 0.024 indicated significant rankings by the experts. Scheme C was ranked first by the experts, followed by schemes A, D, and B being ranked second, third, and fourth respectively. Both schemes A and C categorize errors by cause and, as discussed in the literature review, determining the cause of an error is a significant step in correcting that error. Scheme B contains cause categories only, but it was dismissed by the experts as too simplistic to provide useful information.

Survey question four addressed error prevention. A P-Value of 0.006 indicated consensus among the experts. Scheme C was ranked first by the experts, followed by schemes A, D, and B in respective order. Again cause was the common thread between

the top two schemes for preventing errors. Both schemes A and C categorized errors by the life-cycle phase; however, as discussed earlier in this chapter with investigative question one, the experts ranked this aspect as least important for categorization. This led us to believe that the experts did not consider life-cycle phase in determining their preferences.

Survey question five addresses modification of the development process. A P-Value of 0.05 was just within our rejection region and thus represented significant agreement between the experts on the ranking of schemes . Scheme C was ranked first by the experts, followed by a tie for second for schemes A and D. Scheme B was ranked last by the experts. We believe that the combination of the cause and life-cycle phase is the reason for Scheme C ranking first.

Survey question six addresses changes required in the software process. A P-Value of 0.392 indicated the association between the experts' rankings was insignificant. Thus no ranking conclusions could be made for this question. One possible explanation for this lack of agreement is that each organization utilizes a different software process and thus the scheme requiring the fewest changes in one organization may require many in another organization.

Survey question seven addresses error categorization training. When a process is changed, training is required to make sure benefits are realized from the change. This training typically translates to an additional expense. A P-Value of 0.178 represented no association in the experts' rankings, and thus no ranking conclusions could be made for this question.

Survey question eight addresses the cost to implement an error categorization scheme. It was not our intention to imply that this was true; however, we felt it was important to obtain the experts' opinion on the relative cost of implementing the different schemes. A P-Value of 0.718 represented no agreement by the experts on the rankings,

and thus no ranking conclusions could be made for this question. Again, one possible conclusion is that each organization may utilize a different software process and thus the cost of one scheme may be the cheapest for one organization and the most expensive for another. A different conclusion could be the method by which an organization chooses to implement the schemes, i.e., manually or computer-aided.

Survey question nine addresses software process improvement. A P-Value of 0.012 represented significant rankings by the experts. Scheme C was ranked first by the experts; Scheme A ranked second; and schemes B and D tied for third. The combination of cause and life-cycle phase was the reason for Schemes A and C ranking above the other two. This information fit well with the process improvement methodologies we have studied in our classes at AFIT.

Survey question ten addresses the cost and benefits of implementing an error categorization scheme. Qualifiers were included to instruct the expert to make sure implementation costs as well as savings from long-term use were to be considered in their answer. A P-Value of 0.019 represented significant agreement by the experts. Scheme C was ranked first by the experts, followed by schemes D, A, and B in order. The top two schemes incorporate severity, which was deemed most important by the experts in an earlier question.

The results of survey question eleven showed that the experts preferred Scheme C over the other schemes for use with software system safety. Survey question eleven asked the experts to consider all of their previous answers when ranking the schemes. A P-Value of 0.001 represented significant agreement by the experts. Scheme C was followed by schemes D, A, and B in that order.

A summary test verified the rankings from survey question eleven. We used the rankings from questions two through ten (questions 2, 3, 4, 5, 9, 10) in which the null hypothesis rejected to calculate a Kendall Coefficient and a corresponding P-Value. With

a P-Value of 0.005, we concluded that the rankings derived from the individual questions were significant. More importantly, the rankings are similar to those obtained in question eleven, thus our cross-check was successful.

Investigative Question 4: Are different types of schemes useful for different disciplines: such as information systems, embedded systems or data bases?

This question sought to determine if safety experts with diverse software experience would prefer different categorization schemes. The information necessary to answer this question came from both the summary test ranking and the open-ended data. The ranking data discussed above in investigative question three concluded that Scheme C was preferred by the experts. The results from open-ended question eighteen indicated that our experts did have diverse software experience. Table 9 contains the platform results from question eighteen.

**Table 9. Summary of Experts' Platform Experience**

| | Expert | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Platform | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Avionics | x | | x | x | x | x | x | x | x | x |
| Business | | | | x | | | x | | x | |
| Ground | x | | | | | | x | | | |
| Manned Space | | | | | | x | | | | |
| Missile | x | x | x | | | x | x | | x | |
| Mobile | | | | | | | | | | |
| Ship | | | x | | | | x | | x | |
| Unmanned Space | | | | | | | | | x | |
| Other | | x | | x | | | | | | |

The mobile platform is the only platform where our experts did not have some experience. Both the avionics and missile platforms were well represented by our experts. Based upon the results of our summary statistical test and that our experts have diverse platform experience, we concluded that one scheme is applicable to different software domains.

## *General Findings*

This section discusses general findings drawn from some of the open-ended questions in our survey. The issue of data collection for the candidate schemes is discussed, followed by some general strengths and weaknesses of the schemes.

When we initially searched for experts to participate in our survey, several different sources were contacted concerning the availability of data to test the candidate schemes. Actual DOD mishap data was preferred, but after several sources failed to uncover specific relevant data, we would have welcomed any applicable data. All sources contacted indicated that data was not readily available, and thus the data search ended without success. Survey question twelve was included in the survey to determine if the experts were aware of data being collected for each scheme. After analyzing the results from survey question twelve, we concluded that over fifty percent of the experts were aware of data being collected that could be classified by each of our candidate schemes. This issue will be addressed in the next chapter.

Survey question thirteen addressed the weaknesses of each of the schemes included in our survey. One nearly unanimous comment was that Scheme B was too simplistic to capture much meaningful data. Also, several experts stated that Schemes A and B did not capture the severity of an error. Many of the experts commented that classification of errors into the software control categories in Scheme D was very difficult.

A weakness noted by a couple of experts was that none of the schemes incorporated hardware hazards. While we realize this is important from a total system safety aspect, the focus of our research was categorizing software errors pertaining to software system safety.

Survey question fourteen addressed the strengths of each of the schemes included in our survey. A majority of the experts approved that severity was captured in both schemes C and D. This correlates well with the results of survey question one, where severity was the highest ranked classification scheme. The only other consistent comment pertained to Schemes A and C. The experts felt that both of these schemes would contribute significantly to process improvement initiatives. One theme common to both schemes is that they include cause and life-cycle phase. As mentioned earlier, classification by life-cycle phase ranked last in the first question of the survey. This apparent conflict will be discussed in the next chapter.

## *Summary*

This chapter presented the research findings and survey data analysis. The first and second investigative questions indicated that there are many ways to categorize errors. The method of choice is dependent upon the individual development organization and the processes employed by that organization. The research indicates that more organizations are using combination schemes to deal with the complexity of software errors. The third investigative question indicated that Scheme C was determined to be the most beneficial to software system safety by the experts surveyed. The fourth investigative question indicated that one scheme is applicable to a variety of different software disciplines.

# V. Conclusions and Recommendations

## *Overview*

This chapter presents the conclusions and recommendations of our research effort. The chapter begins with a discussion of the conclusions we made from the investigative question results. The chapter concludes with suggestions for future research.

## *Conclusions*

The research suggested some interesting conclusions about software error categorization. We were unable to locate data to compare or validate the candidate schemes; however, the experts reported that they were aware of data being collected pertaining to the different schemes contained in the survey. This data is being collected, but it is not being consolidated into a public repository or made available in any way for use by software safety researchers. Availability of a data repository of such data would have helped our research greatly.

The research found opinions about the cost of the schemes to be inconsistent. No agreement could be reached about the relative cost of implementing one scheme over another; however, the experts did agree that Scheme C would be more cost beneficial than the other schemes. We concluded from this that even if Scheme C was the most costly to implement, its use would also result in the greatest amount of savings. Implications for future research from this conclusion will be discussed in the next section.

The research also suggests that an error categorization scheme for software safety should include cause and severity as major categories. The experts strongly agreed with the top ranking of Scheme C, which combined cause, severity, and life-cycle phase in one scheme. Life-cycle phase as an individual categorization method was ranked lowest by the experts, thus causing some doubt as to the effectiveness of its inclusion in Scheme C. Therefore, a two-dimensional scheme, categorizing errors according to cause and severity, would be an excellent start for an organization beginning a software safety related error data collection process.

## *Future Research*

There is a great need for research in the area of error categorization pertaining to software system safety. Areas for research include:

- Validation of the effectiveness of schemes
- Usefulness of life-cycle phase data
- Applicability of schemes to non-DOD system development
- Incorporation of hardware and software into one scheme for overall system safety.

Our research discovered that data was being collected that could be used with schemes in our survey. Worthwhile future research could be to apply data to the schemes and assess their effectiveness at capturing and summarizing this data. An effort in this area would provide a transition from theory to reality.

Our research also suggests that error classification by life-cycle phase was the least important to include in a proposed error categorization scheme. Life-cycle phase ranked lowest when experts were asked about the usefulness of collecting error data by various methods. This is counterintuitive to the notion of process improvement, where

phase of development is important. A study into the benefits of including life-cycle phase in an error categorization scheme should be explored.

Another area for future research is to evaluate the schemes using experts from outside the DOD. Such research will indicate whether the same scheme would be applicable to commercial software development. A tremendous amount of software work is being accomplished in the private sector and many companies have surely realized the benefits of error data collection. However, the experts should be selected from organizations doing work where safety is important.

The last recommendation for future research is a direct result of comments by the survey experts pertaining to hardware hazards. Experts indicated that hardware hazards need to be incorporated into the error categorization schemes. A worthwhile effort would be to devise a scheme that could incorporate both software and hardware safety hazards simultaneously.

# Appendix A: Software Error Categorization Survey

Below are software error categorization schemes proposed by various researchers. The specific goals of each researcher are different, but they all relate to improving the software development process. We have labeled the schemes Scheme A, Scheme B, Scheme C, and Scheme D. An introduction to each scheme is provided along with an explanation of terms and an illustrated example of how an error would be placed in a scheme.

## Scheme A

Scheme A attempts to capture error data in terms of the cause of the error and the development phase in which it occurred. Three major causes are listed: Communicational, Conceptual, and Clerical. The development effort is divided into four life-cycle phases: Requirements, High-Level Design, Detailed-Design and Coding, and Debugging and Maintenance.

Explanations:

**Communicational** - Breakdown in communications among team members.

**Conceptual** - Difficulties in analyzing the problem and synthesizing a solution.

**Clerical** - Oversights or simple transcription problems.

| | Requirements | High-Level Design | Detailed-Design and Coding | Debugging and Maintenance |
|---|---|---|---|---|
| Communicational | | | | |
| Conceptual | | | | |
| Clerical | | | | |

Example:

During testing, an error is discovered in a flight control system module. The user did not fully specify that he wanted a backup display updated continuously in-flight. The problem is an annoyance but will cost a bundle to correct.

This error occurred during the requirements phase of development and is a result of miscommunication between the user and the system designers. The error would fall in the cell located under Requirements and across from Communicational.

## Scheme A Example

| | Requirements | High-Level Design | Detailed-Design and Coding | Debugging and Maintenance |
|---|---|---|---|---|
| Communicational | Incorrect display | | | |
| Conceptual | | | | |
| Clerical | | | | |

# Scheme B

Scheme B classifies errors according to their cause. Four major cause categories are used: Communications, Education, Oversight, and Transcription.

Explanations:

**Communications** - Breakdown in communications between team members.

**Education** - Team member's failure to understand something due to inadequate training or education. Errors of this type can be further divided into understanding new or old functions.

**Oversight** - All possible cases or conditions are not considered or handled.

**Transcription** - Simple error.

| | |
|---|---|
| Communications | |
| Education | |
| Oversight | |
| Transcription | |

Example:

During testing, an error is discovered in a flight control system module. The user did not fully specify that he wanted a backup display updated continuously in-flight. The problem is an annoyance but will cost a bundle to correct.

This error is a result of miscommunication between the user and the system designers. The error would fall in the cell located across from Communications.

# Scheme B Example

| | |
|---|---|
| Communications | Incorrect display |
| Education | |
| Oversight | |
| Transcription | |

# Scheme C

Scheme C attempts to capture error data in terms of the cause of the error, the development phase in which it occurred, and the severity of the error. The major causes are: Incorrect Requirements, Communications, Oversight, Interface, Computational, and Transcription. The development effort is divided into four life-cycle phases: Requirements, Design, Coding, and Support. Severity is indicated by adding the code provided for each level of the errors effect on the system: Catastrophic, Critical, Marginal, and Negligible.

Explanations:

**Cause Categories**

**Incorrect Requirements** - The specification, understanding, or applicability of the requirements is insufficient.
**Communications** - Breakdown in communications between team members.
**Oversight** - All possible cases or conditions are not considered or handled.
**Interface** - Anomalies in communication or interaction between systems or subunits.
**Computational** - Incorrect formulation of equations or functions used by the system.
**Transcription** - Simple error.

**Severity Categories**

**Catastrophic** - Results in system loss or life loss.
**Critical** - Results in major system damage or severe injury.
**Marginal** - Results in minor system damage or minor injury.
**Negligible** - Results in less than minor system damage or less than subsystem loss.

| | Requirements | Design | Coding | Support |
|---|---|---|---|---|
| Incorrect Rqmts | | | | |
| Communications | | | | |
| Oversight | | | | |
| Interface | | | | |
| Computational | | | | |
| Transcription | | | | |

Note: Indicate the severity of the error by code:

| | | | |
|---|---|---|---|
| C | Catastrophic | R | Critical |
| M | Marginal | N | Negligible |

Example:

During testing, an error is discovered in a flight control system module. The user did not fully specify that he wanted a backup display updated continuously in-flight. The problem is an annoyance but will cost a bundle to correct.

This error occurred during the requirements phase of development and is a result of miscommunication between the user and the system designers. The error would fall in the cell located under Requirements and across from Communicational. The severity is captured by coding the entry. The problem is minor so the severity is Negligible.

## Scheme C Example

| | Requirements | Design | Coding | Support |
|---|---|---|---|---|
| Incorrect Rqmts | | | | |
| Communications | N- Inc display | | | |
| Oversight | | | | |
| Interface | | | | |
| Computational | | | | |
| Transcription | | | | |

Note: Indicate the severity of the error by code:

| | |
|---|---|
| C | Catastrophic |
| R | Critical |
| M | Marginal |
| N | Negligible |

# Scheme D

Scheme D classifies error data according to the control software module has over the system and the effect of the error on the system. The software control is characterized by the independence of the module in the system and the real-time execution. The effect of the error on the system is divided into four major categories: Catastrophic, Critical, Marginal, and Negligible.

Explanations:

**Control Categories**

**Autonomous Time Critical** - Software exercises autonomous control over potentially hazardous hardware systems, subsystems or components without the possibility of real time human intervention to preclude the occurrence of a hazard.

**Autonomous Not Time Critical** - Software exercises autonomous control over potentially hazardous hardware systems, subsystems or components allowing time for human intervention by independent safety systems to mitigate the hazard.

**Information Time Critical** - Software item displays information requiring immediate operator action to mitigate a hazard.

**Operator Control** - Software items issue commands over potentially hazardous hardware systems, subsystems or components requiring human action to complete the control function.

**Information Decision Algorithm** - Software generates information of a safety critical nature used to make safety critical decisions.

| | Catastrophic | Critical | Marginal | Negligible |
|---|---|---|---|---|
| Autonomous Time Critical | | | | |
| Autonomous Not Time Critical | | | | |
| Information Time Critical | | | | |
| Operator Control | | | | |
| Information Decision Algorithm | | | | |

Example:

During testing, an error is discovered in a flight control system module. The user did not fully specify that he wanted a backup display updated continuously in-flight. The problem is an annoyance but will cost a bundle to correct.

The error is not time critical and only provides information to the operator. Since the problem is minor the severity is Negligible. The error would be placed in the scheme under Negligible and across from Information Decision Algorithm.

## Scheme D Example

| | Catastrophic | Critical | Marginal | Negligible |
|---|---|---|---|---|
| Autonomous Time Critical | | | | |
| Autonomous Not Time Critical | | | | |
| Information Time Critical | | | | |
| Operator Control | | | | |
| Information Decision Algorithm | | | | Incorrect display |

## CONFIDENTIALITY

Although your responses are not anonymous, only the researchers will be able to match your responses with your identity.

Section 1 - This section will collect rank information on the schemes.

1. Many different classification methods of categorizing software errors have been used in previous software work. Please rank from one to five (one being the best, no ties please) the following classification methods from the one most useful to the one least useful in collecting data about software system safety.

_____ Classification by Symptom (effect of error on the system)
_____ Classification by Cause (why the error occurred)
_____ Classification by Life-cycle Phase (when the error was injected)
_____ Classification by Severity (of the error on the system)
_____ Classification by S/W Control (level of control over the system)

2. Rank the provided schemes from one to four (one being the best, no ties please) based upon which scheme would best identify those errors likely to have an impact on the safety of a system.

_____ Scheme A
_____ Scheme B
_____ Scheme C
_____ Scheme D

3. Rank the provided schemes from one to four (one being the best, no ties please) based upon which scheme provides categorization information about critical errors most likely to be of use in **correcting** those errors.

_____ Scheme A
_____ Scheme B
_____ Scheme C
_____ Scheme D

4. Rank the provided schemes from one to four (one being the best, no ties please) based upon which scheme provides categorization information about critical errors most likely to be of use in **preventing** those errors.

_____ Scheme A
_____ Scheme B
_____ Scheme C
_____ Scheme D

5. Rank the provided schemes from one to four (one being the best, no ties please) based upon which scheme provides categorization information about critical errors most likely to be of use in modifying the development process to prevent occurrence of those errors in future systems.

_____ Scheme A
_____ Scheme B
_____ Scheme C
_____ Scheme D

6. Rank the provided schemes from one to four (one being the best, no ties please) based upon which scheme would require the fewest changes to your software process to implement.

_____ Scheme A
_____ Scheme B
_____ Scheme C
_____ Scheme D

7. Rank the provided schemes from one to four (one being the best, no ties please) based upon which scheme would require the least amount of training to implement in your organization.

_____ Scheme A
_____ Scheme B
_____ Scheme C
_____ Scheme D

8. Rank the provided schemes from one to four (one being the best, no ties please) based upon which scheme would be the least costly to implement.

_____ Scheme A
_____ Scheme B
_____ Scheme C
_____ Scheme D

9. Rank the provided schemes from one to four (one being the best, no ties please) based upon which scheme would fit best with ongoing software process improvement initiatives in your organization.

_____ Scheme A
_____ Scheme B
_____ Scheme C
_____ Scheme D

10. Considering all costs associated with implementing a scheme and likely savings accrued from its long-term use, rank the provided schemes from one to four (one being the best, no ties please) based upon which scheme you would expect to be most cost beneficial.

_____ Scheme A
_____ Scheme B
_____ Scheme C
_____ Scheme D

11. Considering your previous replies, rank the provided schemes from one to four (one being the best, no ties please) based upon which scheme you would most favor for use in conjunction with a software safety program.

_____ Scheme A
_____ Scheme B
_____ Scheme C
_____ Scheme D

The following questions will be asked about each individual scheme.

12. Is data being collected that could be used in conjunction with this scheme?

____ Yes
____ No

13. What is the biggest drawback of using this categorization scheme?

14. What is this the best feature of this scheme?

Section 2 - This section will collect information about the interviewed expert

15. What is the highest level of education you have completed? In what disciplines?

Bachelor's degree in _____

Master's degree in _____

Doctorate in _____

16. Have you attended any short courses/seminars in which software system safety was addressed? If yes, describe.

17. What is your current position and how does it relate to Software System Safety?

The following four lists will be used to code any software experience you have:

Platform:

| | | |
|---|---|---|
| 1. Avionics | 4. Manned Space | 7. Ship |
| 2. Business | 5. Missile | 8. Unmanned Space |
| 3. Ground | 6. Mobile | 9. Other |

Application:

| | | |
|---|---|---|
| a. CAD | h. MIS | o. Report Generation |
| b. Command/Control | i. Mission Planning | p. Simulation |
| c. Data Base | j. MMI | q. SW Development Tools |
| d. Diagnostics | k.Office Automation | r. Test |
| e. Flight | l. OS/Executive | s. Training |
| f. Graphics | m. Process Control | t. Utilities |
| g. Message Switching | n. Radar | u. Other |

Type of Involvement:

1. S/W Development
2. Managing S/W Development
3. Operating S/W
4. S/W Acquisition

Length of Experience:

a. Less than 1 year
b. More than 1 year but less than 3 years
c. More than 3 years but less than 5 years
d. More than 5 years

Here is an example of how the experience coding will be accomplished. For six months at an Air Force Program Office, a captain was responsible for the oversight of a contractor that was developing control software for a communication satellite that had developed a problem after launch. In one of the set of blanks provided, the captain would write the platform number (8 for unmanned space) in the first blank, the application (b for command/control) in the second blank, his involvement (4 for S/W Acquisition) in the third blank, and his length of experience (a for 6 months) in the last blank. It would appear like this:

8 b 4 a

18. Please fill out a set of blanks for each type of experience you have with software:

| | | |
|---|---|---|
| ____ ____ ____ ____ | ____ ____ ____ ____ | ____ ____ ____ ____ |
| ____ ____ ____ ____ | ____ ____ ____ ____ | ____ ____ ____ ____ |
| ____ ____ ____ ____ | ____ ____ ____ ____ | ____ ____ ____ ____ |
| ____ ____ ____ ____ | ____ ____ ____ ____ | ____ ____ ____ ____ |

Section 3 - This section will be completed by the interviewer to capture administrative data about the survey.

Person Contacted:

Date/Time:

DOD Affiliation:

List in Appendix: Yes or No

Interviewer:

# Appendix B: Detailed Ranking Data Analysis

Question 1 concerning different classification methods in collecting data about software system safety

Classification Methods

$$Rank := \begin{bmatrix} 4 & 2 & 5 & 3 & 1 \\ 3 & 1 & 2 & 4 & 5 \\ 3 & 4 & 5 & 2 & 1 \\ 2 & 3 & 4 & 1 & 5 \\ 4 & 3 & 5 & 1 & 2 \\ 5 & 3 & 4 & 1 & 2 \\ 2 & 4 & 5 & 1 & 3 \\ 4 & 2 & 3 & 1 & 5 \\ 4 & 3 & 5 & 2 & 1 \\ 2 & 4 & 5 & 1 & 3 \end{bmatrix}$$ Experts

Define a ranking matrix with the columns representing the objects being ranked and the rows being the ranks the objects received from each expert

$Rank := Rank^T$ Matrix is transposed for calculation purposes.

$n := rows(Rank)$ $n = 5$ number of objects being ranked

$k := cols(Rank)$ $k = 10$ number of experts ranking objects

$R := \sum_{j=1}^{k} Rank^{\langle j \rangle}$ Sum of the columns representing the total score received by each object being ranked (the lower the total, the more popular the object)

$R := R^T$ Transpose the vector back into the familiar original

Column sums:

$R = (\ 33 \quad 29 \quad 43 \quad 17 \quad 28\ )$

Define the Kendall Coefficient of Concordance (value ranges between 0 and 1)

$$W := \frac{12 \cdot \sum_{z=1}^{cols(R)} R^{\langle z \rangle\ 2} - 3 \cdot k^2 \cdot n \cdot (n+1)^2}{n \cdot k^2 \cdot n^2 - 1} \qquad W = 0.352$$

Define the Probability Density Function and Cumulative Distribution Function for our Statistical Test:

Chi-Squared PDF:

$$f(x, v) := \frac{1}{2^{\frac{v}{2}} \cdot \Gamma\left(\frac{v}{2}\right)} \cdot x^{\frac{v}{2}-1} \cdot e^{-\frac{x}{2}}$$

Cumulative Chi-Squared Distribution Function

$$F(hi, v) := \int_0^{hi} f(x, v)\, dx$$

Define the necessary parts for our statistical test:

$H_o$ : no significant association exists between the different experts rankings

$H_a$ : a significant association exists between the different experts rankings

$\alpha := .05$ Level of significance for our test

$v := cols(R) - 1$ $v = 4$ Degrees of Freedom for our test (number of objects ranked -1)

Determine the Critical Value for our Test

| | |
|---|---|
| $y := 10$ | initial guess at value |
| $crit := root(F(y, v) - (1 - \alpha), y)$ | Formula to find value where CDF = 1 - sig. level |
| $crit = 9.488$ | Critical value calculated above |

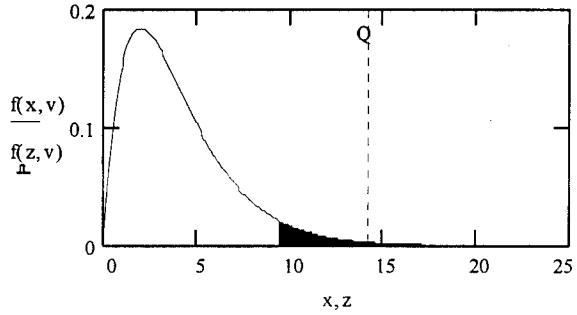
$x := 0, \delta .. 5 \cdot v$ range variable used for Chi-Squared Graph below

$z := crit, crit + \delta .. 5 \cdot v$ range variable used to shade the rejection region in the Graph below

Calculate the Q value that will fall either in our acceptance or rejection region:

| | |
|---|---|
| $k = 10$ | number of experts ranking objects |
| $v = 4$ | Degrees of Freedom (number of objects being ranked minus one) |
| $W = 0.352$ | Kendall Coefficient of Concordance for the set of rankings |
| $Q := k \cdot v \cdot W$ | Formula for calculating Q $Q = 14.08$ |

Now we can graphically assess whether or not we should accept or reject our null hypothesis:




P-Value:

Assuming the null hypothesis is true, The P-value for our calculated Q value can be obtained by subtracting the Q value's cumulative probability from 1:

$1 - F(Q, v) = 0.007$

## Figure 2. Decision Graph for Survey Question 1

Because the Q value is well within our rejection region also substantiated by a P-value significantly less than .05, we reject the null hypothesis and conclude that there is a significant association between the different experts rankings. We can further conclude that the order of preference for gathering data about Software System Safety is as follows:

$R = ( 33 \quad 29 \quad 43 \quad 17 \quad 28 )$ The column totals calculated above

1) Severity
2) Cause and S/W Control (No apparent preference for one of these over the other)
3) Symptom
4) Life Cycle Phase

Question 2 concerning scheme that would best identify those errors likely to have an impact on the safety of a system.

Schemes

$$\begin{matrix} & (A & B & C & D) & \\ \text{Rank} := & \begin{bmatrix} 4 & 3 & 2 & 1 \\ 2 & 3 & 1 & 4 \\ 4 & 3 & 2 & 1 \\ 4 & 3 & 1 & 2 \\ 3 & 4 & 2 & 1 \\ 4 & 3 & 2 & 1 \\ 3 & 4 & 2 & 1 \\ 3 & 4 & 1 & 2 \\ 4 & 3 & 2 & 1 \\ 4 & 3 & 1 & 2 \end{bmatrix} & \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \end{bmatrix} & \begin{matrix} E \\ x \\ p \\ e \\ r \\ t \\ s \end{matrix} \end{matrix}$$

Define a ranking matrix with the columns representing the objects being ranked and the rows being the ranks the objects received from each expert

$\text{Rank} := \text{Rank}^T$ Matrix is transposed for calculation purposes.

$n := \text{rows}(\text{Rank})$ $n = 4$ number of objects being ranked

$k := \text{cols}(\text{Rank})$ $k = 10$ number of experts ranking objects

$R := \sum_{j=1}^{k} \text{Rank}^{\langle j \rangle}$ Sum of the columns representing the total score received by each object being ranked (the lower the total, the more popular the object)

$R := R^T$ Transpose the vector back into the familiar original

Column sums:

A B C D

$R = (35 \quad 33 \quad 16 \quad 16)$

Define the Kendall Coefficient of Concordance (value ranges between 0 and 1)

$$W := \frac{12 \cdot \sum_{z=1}^{\text{cols}(R)} R^{\langle z \rangle^2} - 3 \cdot k^2 \cdot n \cdot (n+1)^2}{n \cdot k^2 \cdot (n^2 - 1)} \qquad W = 0.652$$

Define the Probability Density Function and Cumulative Distribution Function for our Statistical Test:

Chi-Squared PDF:

$$f(x, v) := \frac{1}{2^{\frac{v}{2}} \cdot \Gamma\left(\frac{v}{2}\right)} \cdot x^{\frac{v}{2}-1} \cdot e^{-\frac{x}{2}}$$

Cumulative Chi-Squared Distribution Function:

$$F(hi, v) := \int_0^{hi} f(x, v)\, dx$$

Define the necessary parts for our statistical test:

$H_o$ : no significant association exists between the different experts rankings

$H_a$ : a significant association exists between the different experts rankings

$\alpha := .05$ Level of significance for our test

$v := cols(R) - 1 \quad v = 3$ Degrees of Freedom for our test (number of objects ranked -1)

Determine the Critical Value for our Test

$y := 10$ initial guess at value

$crit := root(F(y,v) - (1 - \alpha), y)$ Formula to find value where CDF = 1 - sig. level

$crit = 7.817$ Critical value calculated above

$x := 0, \delta .. 10 \cdot v$ range variable used for Chi-Squared Graph below

$z := crit, crit + \delta .. 5 \cdot v$ range variable used to shade the rejection region in the Graph below

Calculate the Q value that will fall either in our acceptance or rejection region:

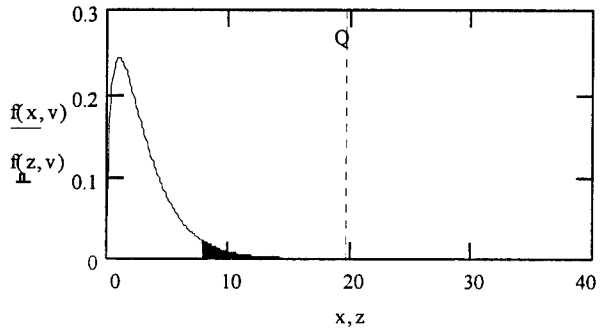
$k = 10$ number of experts ranking objects

$v = 3$ Degrees of Freedom (number of objects being ranked minus one)

$W = 0.652$ Kendall Coefficient of Concordance for the set of rankings

$Q := k \cdot v \cdot W$ Formula for calculating Q $Q = 19.56$

Now we can graphically assess whether or not we should accept or reject our null hypothesis:




P-Value:

Assuming the null hypothesis is true, The P-value for our calculated Q value can be obtained by subtracting the Q value's cumulative probability from 1:

$1 - F(Q,v) = 0.00022$

**Figure 3. Decision Graph for Survey Question 2**

Because the Q value is well within our rejection region also substantiated by a P-value significantly less than .05, we reject the null hypothesis and conclude that there is a significant association between the different experts rankings. We can further conclude that the order of preference for the schemes that would best identify those errors likely to have an impact on the safety of a system is as follows:

A B C D

$R = ( 35 \quad 33 \quad 16 \quad 16 )$ The column totals calculated on previous page

1) C and D
2) A and B

Question 3 concerning scheme that provides categorization information about critical errors most likely to be of use in correcting those errors.

Schemes

$$Rank := \begin{bmatrix} 1 & 4 & 3 & 2 \\ 3 & 2 & 1 & 4 \\ 2 & 4 & 1 & 3 \\ 2 & 1 & 3 & 4 \\ 3 & 4 & 2 & 1 \\ 4 & 3 & 2 & 1 \\ 2 & 4 & 1 & 3 \\ 2 & 4 & 1 & 3 \\ 2 & 3 & 1 & 4 \\ 2 & 4 & 1 & 3 \end{bmatrix}$$

Experts

Define a ranking matrix with the columns representing the objects being ranked and the rows being the ranks the objects received from each expert

$Rank := Rank^T$ Matrix is transposed for calculation purposes.

$n := rows(Rank)$ $n = 4$ number of objects being ranked

$k := cols(Rank)$ $k = 10$ number of experts ranking objects

$R := \sum_{j=1}^{k} Rank^{\langle j \rangle}$ Sum of the columns representing the total score received by each object being ranked (the lower the total, the more popular the object)

$R := R^T$ Transpose the vector back into the familiar original

Column sums:

$R = ( 23 \quad 33 \quad 16 \quad 28 )$

Define the Kendall Coefficient of Concordance (value ranges between 0 and 1)

$$W := \frac{12 \cdot \sum_{z=1}^{cols(R)} R^{\langle z \rangle^2} - 3 \cdot k^2 \cdot n \cdot (n+1)^2}{n \cdot k^2 \cdot (n^2 - 1)} \qquad W = 0.316$$

Define the Probability Density Function and Cumulative Distribution Function for our Statistical Test:

Chi-Squared PDF:

$$f(x, v) := \frac{1}{2^{\frac{v}{2}} \cdot \Gamma\left(\frac{v}{2}\right)} \cdot x^{\frac{v}{2}-1} \cdot e^{-\frac{x}{2}}$$

Cumulative Chi-Squared Distribution Function:

$$F(hi, v) := \int_0^{hi} f(x, v)\, dx$$

Define the necessary parts for our statistical test:

$H_o$ : no significant association exists between the different experts rankings

$H_a$ : a significant association exists between the different experts rankings

$\alpha := .05$ Level of significance for our test

$v := cols(R) - 1 \quad v = 3$ Degrees of Freedom for our test (number of objects ranked -1)

Determine the Critical Value for our Test

$y := 10$ initial guess at value

$crit := root(F(y,v) - (1 - \alpha), y)$ Formula to find value where CDF = 1 - sig. level

$crit = 7.817$ Critical value calculated above

$x := 0, \delta .. 10 \cdot v$ range variable used for Chi-Squared Graph below

$z := crit, crit + \delta .. 5 \cdot v$ range variable used to shade the rejection region in the Graph below

Calculate the Q value that will fall either in our acceptance or rejection region:

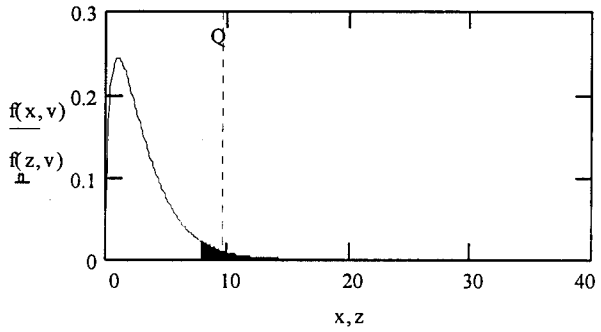
$k = 10$ number of experts ranking objects

$v = 3$ Degrees of Freedom (number of objects being ranked minus one)

$W = 0.316$ Kendall Coefficient of Concordance for the set of rankings

$Q := k \cdot v \cdot W$ Formula for calculating Q $Q = 9.48$

Now we can graphically assess whether or not we should accept or reject our null hypothesis:




P-Value:

Assuming the null hypothesis is true, The P-value for our calculated Q value can be obtained by subtracting the Q value's cumulative probability from 1:

$1 - F(Q,v) = 0.024$

**Figure 4. Decision Graph for Survey Question 3**

Because the Q value is well within our rejection region also substantiated by a P-value significantly less than .05, we reject the null hypothesis and conclude that there is a significant association between the different experts rankings. We can further conclude that the order of preference for the schemes that provide categorization information about critical errors most likely to be of use in correcting those errors is as follows:

$R = ( 23 \quad 33 \quad 16 \quad 28 )$ The column totals calculated on previous page

1) C
2) A
3) D
4) B

Question 4 concerning scheme that provides categorization information about critical errors most likely to be of use in preventing those errors.

Schemes

$$Rank := \begin{bmatrix} 3 & 2 & 1 & 4 \\ 2 & 3 & 1 & 4 \\ 4 & 3 & 2 & 1 \\ 1 & 3 & 2 & 4 \\ 3 & 4 & 2 & 1 \\ 4 & 3 & 1 & 2 \\ 2 & 4 & 1 & 3 \\ 2 & 3 & 1 & 4 \\ 2 & 3 & 1 & 4 \\ 3 & 4 & 1 & 2 \end{bmatrix}$$

Experts

Define a ranking matrix with the columns representing the objects being ranked and the rows being the ranks the objects received from each expert

$Rank := Rank^T$ Matrix is transposed for calculation purposes.

$n := rows(Rank)$ $n = 4$ number of objects being ranked

$k := cols(Rank)$ $k = 10$ number of experts ranking objects

$R := \sum_{j=1}^{k} Rank^{\langle j \rangle}$ Sum of the columns representing the total score received by each object being ranked (the lower the total, the more popular the object)

$R := R^T$ Transpose the vector back into the familiar original

Column sums:

$R = (26 \quad 32 \quad 13 \quad 29)$

Define the Kendall Coefficient of Concordance (value ranges between 0 and 1)

$$W := \frac{12 \cdot \sum_{z=1}^{cols(R)} R^{\langle z \rangle^2} - 3 \cdot k^2 \cdot n \cdot (n+1)^2}{n \cdot k^2 \cdot (n^2 - 1)} \qquad W = 0.42$$

Define the Probability Density Function and Cumulative Distribution Function for our Statistical Test:

Chi-Squared PDF:

$$f(x, v) := \frac{1}{2^{\frac{v}{2}} \cdot \Gamma\left(\frac{v}{2}\right)} \cdot x^{\frac{v}{2}-1} \cdot e^{-\frac{x}{2}}$$

Cumulative Chi-Squared Distribution Function:

$$F(hi, v) := \int_0^{hi} f(x, v)\, dx$$

Define the necessary parts for our statistical test:

$H_o$ : no significant association exists between the different experts rankings

$H_a$ : a significant association exists between the different experts rankings

$\alpha := .05$ Level of significance for our test

$v := cols(R) - 1$ $v = 3$ Degrees of Freedom for our test (number of objects ranked -1)

Determine the Critical Value for our Test

$y := 10$ initial guess at value

$crit := root(F(y,v) - (1 - \alpha), y)$ Formula to find value where CDF = 1 - sig. level

$crit = 7.817$ Critical value calculated above

$x := 0, \delta .. 10 \cdot v$ range variable used for Chi-Squared Graph below

$z := crit, crit + \delta .. 5 \cdot v$ range variable used to shade the rejection region in the Graph below

Calculate the Q value that will fall either in our acceptance or rejection region:

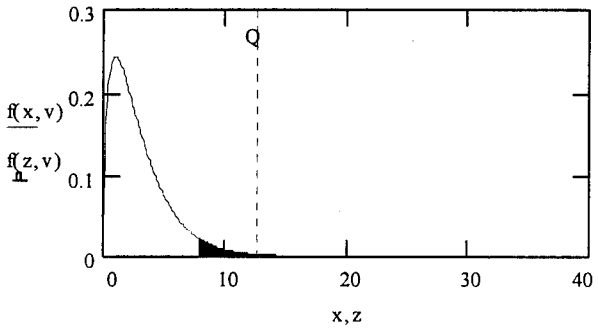
$k = 10$ number of experts ranking objects

$v = 3$ Degrees of Freedom (number of objects being ranked minus one)

$W = 0.42$ Kendall Coefficient of Concordance for the set of rankings

$Q := k \cdot v \cdot W$ Formula for calculating Q $Q = 12.6$

Now we can graphically assess whether or not we should accept or reject our null hypothesis:




P-Value:

Assuming the null hypothesis is true, The P-value for our calculated Q value can be obtained by subtracting the Q value's cumulative probability from 1:

$1 - F(Q,v) = 0.006$

**Figure 5. Decision Graph for Survey Question 4**

Because the Q value is well within our rejection region also substantiated by a P-value significantly less than .05, we reject the null hypothesis and conclude that there is a significant association between the different experts rankings. We can further conclude that the order of preference for the schemes that provides categorization information about critical errors most likely to be of use in preventing those errors is as follows:

$R = ( 26 \quad 32 \quad 13 \quad 29 )$ The column totals calculated on previous page

1) C
2) A and D
3) B

Question 5 concerning scheme that provides categorization information about critical errors most likely to be of use in modifying the development process to prevent occurrence of those errors in future systems.

Schemes

$$Rank := \begin{bmatrix} 1 & 3 & 4 & 2 \\ 2 & 4 & 1 & 3 \\ 3 & 4 & 2 & 1 \\ 1 & 2 & 3 & 4 \\ 3 & 4 & 2 & 1 \\ 4 & 3 & 1 & 2 \\ 2 & 3 & 1 & 4 \\ 2 & 3 & 1 & 4 \\ 4 & 3 & 1 & 2 \\ 2 & 4 & 1 & 3 \end{bmatrix}$$

Experts

Define a ranking matrix with the columns representing the objects being ranked and the rows being the ranks the objects received from each expert

$Rank := Rank^T$ Matrix is transposed for calculation purposes.

$n := rows(Rank)$ $n = 4$ number of objects being ranked

$k := cols(Rank)$ $k = 10$ number of experts ranking objects

$R := \sum_{j=1}^{k} Rank^{\langle j \rangle}$ Sum of the columns representing the total score received by each object being ranked (the lower the total, the more popular the object)

$R := R^T$ Transpose the vector back into the familiar original

Column sums:

$R = ( 24 \quad 33 \quad 17 \quad 26 )$

Define the Kendall Coefficient of Concordance (value ranges between 0 and 1)

$$W := \frac{12 \cdot \sum_{z=1}^{cols(R)} R^{\langle z \rangle^2} - 3 \cdot k^2 \cdot n \cdot (n+1)^2}{n \cdot k^2 \cdot n^2 - 1} \qquad W = 0.26$$

Define the Probability Density Function and Cumulative Distribution Function for our Statistical Test:

Chi-Squared PDF:

$$f(x, v) := \frac{1}{2^{\frac{v}{2}} \cdot \Gamma\left(\frac{v}{2}\right)} \cdot x^{\frac{v}{2}-1} \cdot e^{-\frac{x}{2}}$$

Cumulative Chi-Squared Distribution Function:

$$F(hi, v) := \int_0^{hi} f(x, v)\, dx$$

Define the necessary parts for our statistical test:

$H_o$ : no significant association exists between the different experts rankings

$H_a$ : a significant association exists between the different experts rankings

$\alpha := .05$ Level of significance for our test

$v := cols(R) - 1 \quad v = 3$ Degrees of Freedom for our test (number of objects ranked -1)

Determine the Critical Value for our Test

$y := 10$ initial guess at value

$crit := root(F(y,v) - (1 - \alpha), y)$ Formula to find value where CDF = 1 - sig. level

$crit = 7.817$ Critical value calculated above

$x := 0, \delta .. 10 \cdot v$ range variable used for Chi-Squared Graph below

$z := crit, crit + \delta .. 5 \cdot v$ range variable used to shade the rejection region in the Graph below

Calculate the Q value that will fall either in our acceptance or rejection region:

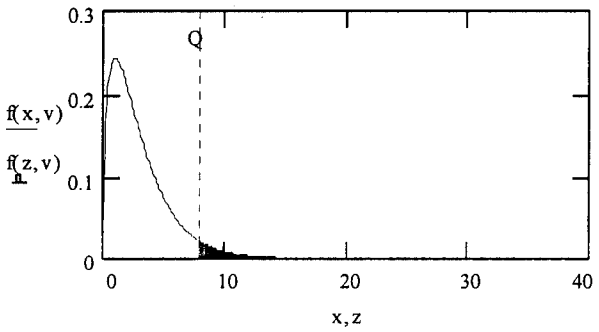
$k = 10$ number of experts ranking objects

$v = 3$ Degrees of Freedom (number of objects being ranked minus one)

$W = 0.26$ Kendall Coefficient of Concordance for the set of rankings

$Q := k \cdot v \cdot W$ Formula for calculating Q $Q = 7.8$

Now we can graphically assess whether or not we should accept or reject our null hypothesis:




P-Value:

Assuming the null hypothesis is true, The P-value for our calculated Q value can be obtained by subtracting the Q value's cumulative probability from 1:

$1 - F(Q,v) = 0.05$

**Figure 6. Decision Graph for Survey Question 5**

Because the Q value is within our rejection region also substantiated by a P-value equal to .05, we reject the null hypothesis and conclude that there is a significant association between the different experts rankings. We can further conclude that the order of preference for the schemes that provide categorization information about critical errors most likely to be of use in modifying the development process to prevent occurrence of those errors in future systems is as follows:

$R = ( 24 \quad 33 \quad 17 \quad 26 )$ The column totals calculated on previous page

1) C
2) A and D
3) B

Question 6 concerning scheme that would require the fewest changes to your software process to implement.

Schemes

$$Rank := \begin{bmatrix} 1 & 3 & 2 & 4 \\ 3 & 2 & 1 & 4 \\ 2 & 1 & 3 & 4 \\ 3 & 2 & 4 & 1 \\ 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 3 & 1 & 4 & 2 \\ 1 & 3 & 2 & 4 \\ 4 & 3 & 1 & 2 \\ 3 & 4 & 1 & 2 \end{bmatrix}$$

Experts

Define a ranking matrix with the columns representing the objects being ranked and the rows being the ranks the objects received from each expert

$Rank := Rank^T$ Matrix is transposed for calculation purposes.

$n := rows(Rank)$ $n = 4$ number of objects being ranked

$k := cols(Rank)$ $k = 10$ number of experts ranking objects

$R := \sum_{j=1}^{k} Rank^{\langle j \rangle}$ Sum of the columns representing the total score received by each object being ranked (the lower the total, the more popular the object)

$R := R^T$ Transpose the vector back into the familiar original

Column sums:

$R = (22 \quad 23 \quad 24 \quad 31)$

Define the Kendall Coefficient of Concordance (value ranges between 0 and 1)

$$W := \frac{12 \cdot \sum_{z=1}^{cols(R)} R^{\langle z \rangle^2} - 3 \cdot k^2 \cdot n \cdot (n+1)^2}{n \cdot k^2 \cdot (n^2 - 1)} \qquad W = 0.1$$

Define the Probability Density Function and Cumulative Distribution Function for our Statistical Test:

Chi-Squared PDF:

$$f(x, v) := \frac{1}{2^{\frac{v}{2}} \cdot \Gamma\left(\frac{v}{2}\right)} \cdot x^{\frac{v}{2}-1} \cdot e^{-\frac{x}{2}}$$

Cumulative Chi-Squared Distribution Function:

$$F(hi, v) := \int_0^{hi} f(x, v)\, dx$$

Define the necessary parts for our statistical test:

$H_o$ : no significant association exists between the different experts rankings

$H_a$ : a significant association exists between the different experts rankings

$\alpha := .05$ Level of significance for our test

$v := cols(R) - 1 \quad v = 3$ Degrees of Freedom for our test (number of objects ranked -1)

Determine the Critical Value for our Test

$y := 10$ initial guess at value

$crit := root(F(y,v) - (1 - \alpha), y)$ Formula to find value where CDF = 1 - sig. level

$crit = 7.817$ Critical value calculated above

$x := 0, \delta .. 10 \cdot v$ range variable used for Chi-Squared Graph below

$z := crit, crit + \delta .. 5 \cdot v$ range variable used to shade the rejection region in the Graph below

Calculate the Q value that will fall either in our acceptance or rejection region:

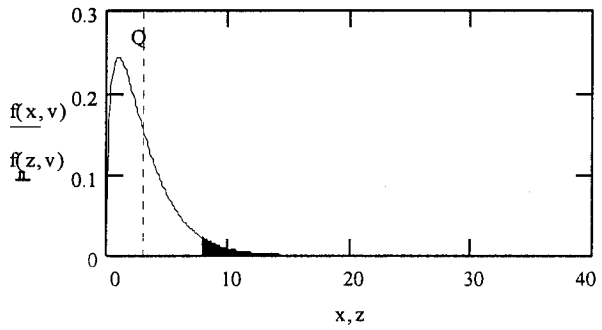
$k = 10$ number of experts ranking objects

$v = 3$ Degrees of Freedom (number of objects being ranked minus one)

$W = 0.1$ Kendall Coefficient of Concordance for the set of rankings

$Q := k \cdot v \cdot W$ Formula for calculating Q $Q = 3$

Now we can graphically assess whether or not we should accept or reject our null hypothesis:




P-Value:

Assuming the null hypothesis is true, The P-value for our calculated Q value can be obtained by subtracting the Q value's cumulative probability from 1:

$1 - F(Q, v) = 0.392$

**Figure 7. Decision Graph for Survey Question 6**

Because the Q value is well within our acceptance region also substantiated by a P-value significantly greater than .05, we accept the null hypothesis and conclude that there is no significant association between the different experts rankings. We cannot make any conclusions concerning the order of preference for the schemes that would require the fewest changes to your software process to implement.

Question 7 concerning scheme that would require the least amount of training to implement.

Schemes

$$Rank := \begin{bmatrix} 1 & 3 & 2 & 4 \\ 2 & 1 & 3 & 4 \\ 2 & 1 & 3 & 4 \\ 3 & 2 & 4 & 1 \\ 2 & 1 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 2 & 1 & 4 & 3 \\ 1 & 3 & 2 & 4 \\ 3 & 4 & 1 & 2 \\ 4 & 3 & 1 & 2 \end{bmatrix}$$

Experts

Define a ranking matrix with the columns representing the objects being ranked and the rows being the ranks the objects received from each expert

$Rank := Rank^T$ Matrix is transposed for calculation purposes.

$n := rows(Rank)$ $n = 4$ number of objects being ranked

$k := cols(Rank)$ $k = 10$ number of experts ranking objects

$R := \sum_{j=1}^{k} Rank^{\langle j \rangle}$ Sum of the columns representing the total score received by each object being ranked (the lower the total, the more popular the object)

$R := R^T$ Transpose the vector back into the familiar original

Column sums:

$R = ( 21 \quad 21 \quad 26 \quad 32 )$

Define the Kendall Coefficient of Concordance (value ranges between 0 and 1)

$$W := \frac{12 \cdot \sum_{z=1}^{cols(R)} R^{\langle z \rangle^2} - 3 \cdot k^2 \cdot n \cdot (n+1)^2}{n \cdot k^2 \cdot n^2 - 1} \qquad W = 0.164$$

Define the Probability Density Function and Cumulative Distribution Function for our Statistical Test:

Chi-Squared PDF:

$$f(x, v) := \frac{1}{2^{\frac{v}{2}} \cdot \Gamma\left(\frac{v}{2}\right)} \cdot x^{\frac{v}{2}-1} \cdot e^{-\frac{x}{2}}$$

Cumulative Chi-Squared Distribution Function:

$$F(hi, v) := \int_0^{hi} f(x, v)\, dx$$

Define the necessary parts for our statistical test:

$H_o$ : no significant association exists between the different experts rankings

$H_a$ : a significant association exists between the different experts rankings

$\alpha := .05$ Level of significance for our test

$v := cols(R) - 1 \quad v = 3$ Degrees of Freedom for our test (number of objects ranked -1)

Determine the Critical Value for our Test

| | |
|---|---|
| $y := 10$ | initial guess at value |
| $crit := root(F(y, v) - (1 - \alpha), y)$ | Formula to find value where CDF = 1 - sig. level |
| $crit = 7.817$ | Critical value calculated above |

$x := 0, \delta .. 10 \cdot v$ range variable used for Chi-Squared Graph below

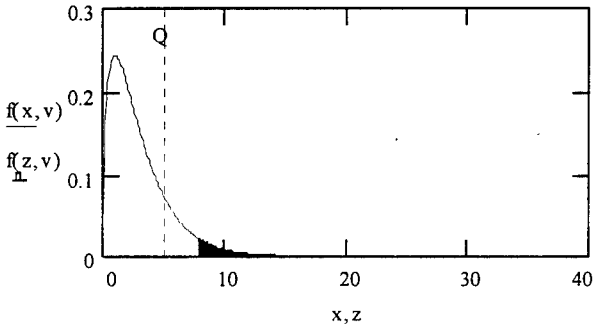
$z := crit, crit + \delta .. 5 \cdot v$ range variable used to shade the rejection region in the Graph below

Calculate the Q value that will fall either in our acceptance or rejection region:

| | |
|---|---|
| $k = 10$ | number of experts ranking objects |
| $v = 3$ | Degrees of Freedom (number of objects being ranked minus one) |
| $W = 0.164$ | Kendall Coefficient of Concordance for the set of rankings |

$Q := k \cdot v \cdot W$ Formula for calculating Q $\quad Q = 4.92$

Now we can graphically assess whether or not we should accept or reject our null hypothesis:




P-Value:

Assuming the null hypothesis is true, The P-value for our calculated Q value can be obtained by subtracting the Q value's cumulative probability from 1:

$1 - F(Q, v) = 0.178$

## Figure 8. Decision Graph for Survey Question 7

Because the Q value is well within our acceptance region also substantiated by a P-value significantly greater than .05, we accept the null hypothesis and conclude that there is no significant association between the different experts rankings. We cannot make any conclusions concerning the order of preference for the schemes that would require the least amount of training to implement.

Question 8 concerning scheme that would be the least costly to implement.

Schemes

$$\text{Rank} := \begin{bmatrix} 1 & 2 & 4 & 3 \\ 2 & 3 & 1 & 4 \\ 3 & 4 & 2 & 1 \\ 3 & 2 & 4 & 1 \\ 2 & 1 & 3 & 4 \\ 2 & 1 & 3 & 4 \\ 2 & 1 & 4 & 3 \\ 2 & 1 & 3 & 4 \\ 3 & 4 & 1 & 2 \\ 3 & 4 & 1 & 2 \end{bmatrix}$$

Experts

Define a ranking matrix with the columns representing the objects being ranked and the rows being the ranks the objects received from each expert

$\text{Rank} := \text{Rank}^T$ Matrix is transposed for calculation purposes.

$n := \text{rows}(\text{Rank})$ $n = 4$ number of objects being ranked

$k := \text{cols}(\text{Rank})$ $k = 10$ number of experts ranking objects

$R := \sum_{j=1}^{k} \text{Rank}^{\langle j \rangle}$ Sum of the columns representing the total score received by each object being ranked (the lower the total, the more popular the object)

$R := R^T$ Transpose the vector back into the familiar original

Column sums:

$R = (\ 23 \quad 23 \quad 26 \quad 28\ )$

Define the Kendall Coefficient of Concordance (value ranges between 0 and 1)

$$W := \frac{12 \cdot \sum_{z=1}^{\text{cols}(R)} R^{\langle z \rangle\ 2} - 3 \cdot k^2 \cdot n \cdot (n+1)^2}{n \cdot k^2 \cdot n^2 - 1} \qquad W = 0.036$$

Define the Probability Density Function and Cumulative Distribution Function for our Statistical Test:

Chi-Squared PDF:

$$f(x, v) := \frac{1}{2^{\frac{v}{2}} \cdot \Gamma\left(\frac{v}{2}\right)} \cdot x^{\frac{v}{2}-1} \cdot e^{-\frac{x}{2}}$$

Cumulative Chi-Squared Distribution Function:

$$F(hi, v) := \int_0^{hi} f(x, v)\, dx$$

Define the necessary parts for our statistical test:

$H_o$ : no significant association exists between the different experts rankings

$H_a$ : a significant association exists between the different experts rankings

$\alpha := .05$ Level of significance for our test

$v := cols(R) - 1 \quad v = 3$ Degrees of Freedom for our test (number of objects ranked -1)

Determine the Critical Value for our Test

| | |
|---|---|
| $y := 10$ | initial guess at value |
| $crit := root(F(y,v) - (1 - \alpha), y)$ | Formula to find value where CDF = 1 - sig. level |
| $crit = 7.817$ | Critical value calculated above |

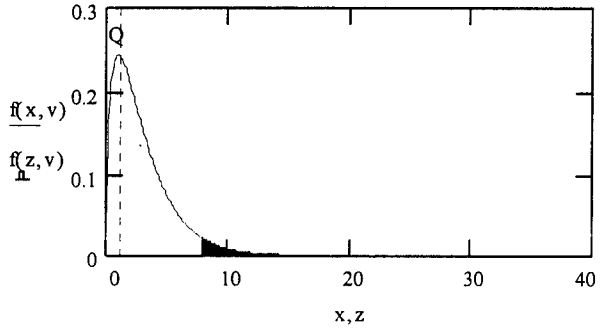
$x := 0, \delta .. 10 \cdot v$ range variable used for Chi-Squared Graph below

$z := crit, crit + \delta .. 5 \cdot v$ range variable used to shade the rejection region in the Graph below

Calculate the Q value that will fall either in our acceptance or rejection region:

| | |
|---|---|
| $k = 10$ | number of experts ranking objects |
| $v = 3$ | Degrees of Freedom (number of objects being ranked minus one) |
| $W = 0.036$ | Kendall Coefficient of Concordance for the set of rankings |
| $Q := k \cdot v \cdot W$ | Formula for calculating Q $\quad Q = 1.08$ |

Now we can graphically assess whether or not we should accept or reject our null hypothesis:




P-Value:

Assuming the null hypothesis is true, The P-value for our calculated Q value can be obtained by subtracting the Q value's cumulative probability from 1:

$1 - F(Q,v) = 0.782$

## Figure 9. Decision Graph for Survey Question 8

Because the Q value is well within our acceptance region also substantiated by a P-value significantly greater than .05, we accept the null hypothesis and conclude that there is no significant association between the different experts rankings. We cannot make any conclusions concerning the order of preference for the schemes that would be the least costly to implement.

Question 9 concerning scheme that would fit best with ongoing software process improvement initiatives in your organization.

Schemes

$$Rank := \begin{bmatrix} 2 & 3 & 1 & 4 \\ 2 & 3 & 1 & 4 \\ 3 & 4 & 1 & 2 \\ 2 & 4 & 1 & 3 \\ 4 & 3 & 2 & 1 \\ 2 & 1 & 3 & 4 \\ 2 & 3 & 1 & 4 \\ 1 & 3 & 2 & 4 \\ 3 & 4 & 1 & 2 \\ 4 & 3 & 1 & 2 \end{bmatrix}$$

Experts

Define a ranking matrix with the columns representing the objects being ranked and the rows being the ranks the objects received from each expert

$Rank := Rank^T$ Matrix is transposed for calculation purposes.

$n := rows(Rank)$ $n = 4$ number of objects being ranked

$k := cols(Rank)$ $k = 10$ number of experts ranking objects

$R := \sum_{j=1}^{k} Rank^{<j>}$ Sum of the columns representing the total score received by each object being ranked (the lower the total, the more popular the object)

$R := R^T$ Transpose the vector back into the familiar original

Column sums:

$R = ( 25 \quad 31 \quad 14 \quad 30 )$

Define the Kendall Coefficient of Concordance (value ranges between 0 and 1)

$$W := \frac{12 \cdot \sum_{z=1}^{cols(R)} R^{<z>\ 2} - 3 \cdot k^2 \cdot n \cdot (n+1)^2}{n \cdot k^2 \cdot n^2 - 1} \qquad W = 0.364$$

Define the Probability Density Function and Cumulative Distribution Function for our Statistical Test:

Chi-Squared PDF:

$$f(x, v) := \frac{1}{2^{\frac{v}{2}} \cdot \Gamma\left(\frac{v}{2}\right)} \cdot x^{\frac{v}{2}-1} \cdot e^{-\frac{x}{2}}$$

Cumulative Chi-Squared Distribution Function:

$$F(hi, v) := \int_0^{hi} f(x, v)\, dx$$

Define the necessary parts for our statistical test:

$H_o$ : no significant association exists between the different experts rankings

$H_a$ : a significant association exists between the different experts rankings

$\alpha := .05$ Level of significance for our test

$v := cols(R) - 1 \quad v = 3$ Degrees of Freedom for our test (number of objects ranked -1)

Determine the Critical Value for our Test

| | |
|---|---|
| $y := 10$ | initial guess at value |
| $crit := root(F(y, v) - (1 - \alpha), y)$ | Formula to find value where CDF = 1 - sig. level |
| $crit = 7.817$ | Critical value calculated above |

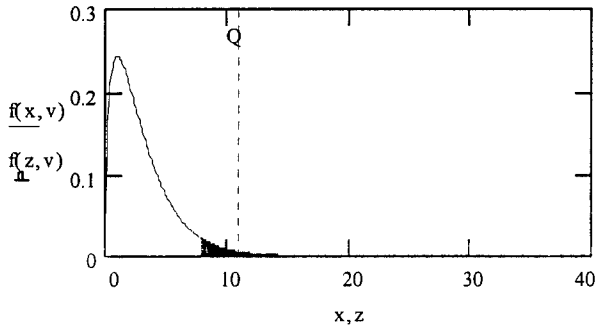
$x := 0, \delta .. 10 \cdot v$, range variable used for Chi-Squared Graph below

$z := crit, crit + \delta .. 5 \cdot v$ range variable used to shade the rejection region in the Graph below

Calculate the Q value that will fall either in our acceptance or rejection region:

| | |
|---|---|
| $k = 10$ | number of experts ranking objects |
| $v = 3$ | Degrees of Freedom (number of objects being ranked minus one) |
| $W = 0.364$ | Kendall Coefficient of Concordance for the set of rankings |
| $Q := k \cdot v \cdot W$ | Formula for calculating Q $\quad Q = 10.92$ |

Now we can graphically assess whether or not we should accept or reject our null hypothesis:




P-Value:

Assuming the null hypothesis is true, The P-value for our calculated Q value can be obtained by subtracting the Q value's cumulative probability from 1:

$1 - F(Q, v) = 0.012$

**Figure 10. Decision Graph for Survey Question 9**

Because the Q value is well within our rejection region also substantiated by a P-value significantly less than .05, we reject the null hypothesis and conclude that there is a significant association between the different experts rankings. We can further conclude that the order of preference for the schemes that would fit best with ongoing software process improvement initiatives in your organization is as follows:

$R = ( 25 \quad 31 \quad 14 \quad 30 )$ The column totals calculated on previous page

1) C
2) A
3) B and D

Question 10 concerning scheme that they expect would be most cost beneficial.

Schemes

$$\text{Rank} := \begin{bmatrix} 1 & 3 & 4 & 2 \\ 2 & 3 & 1 & 4 \\ 4 & 3 & 2 & 1 \\ 1 & 4 & 3 & 2 \\ 4 & 3 & 2 & 1 \\ 4 & 3 & 2 & 1 \\ 2 & 4 & 1 & 3 \\ 2 & 4 & 1 & 3 \\ 3 & 4 & 1 & 2 \\ 3 & 4 & 1 & 2 \end{bmatrix}$$

Experts

Define a ranking matrix with the columns representing the objects being ranked and the rows being the ranks the objects received from each expert

$\text{Rank} := \text{Rank}^T$ Matrix is transposed for calculation purposes.

$n := \text{rows}(\text{Rank})$ $n = 4$ number of objects being ranked

$k := \text{cols}(\text{Rank})$ $k = 10$ number of experts ranking objects

$R := \sum_{j=1}^{k} \text{Rank}^{\langle j \rangle}$ Sum of the columns representing the total score received by each object being ranked (the lower the total, the more popular the object)

$R := R^T$ Transpose the vector back into the familiar original

Column sums:

$R = (\ 26 \quad 35 \quad 18 \quad 21\ )$

Define the Kendall Coefficient of Concordance (value ranges between 0 and 1)

$$W := \frac{12 \cdot \sum_{z=1}^{\text{cols}(R)} R^{\langle z \rangle\ 2} - 3 \cdot k^2 \cdot n \cdot (n+1)^2}{n \cdot k^2 \cdot n^2 - 1} \qquad W = 0.332$$

Define the Probability Density Function and Cumulative Distribution Function for our Statistical Test:

Chi-Squared PDF:

$$f(x, v) := \frac{1}{2^{\frac{v}{2}} \cdot \Gamma\left(\frac{v}{2}\right)} \cdot x^{\frac{v}{2}-1} \cdot e^{-\frac{x}{2}}$$

Cumulative Chi-Squared Distribution Function:

$$F(hi, v) := \int_0^{hi} f(x, v)\, dx$$

Define the necessary parts for our statistical test:

$H_o$ : no significant association exists between the different experts rankings

$H_a$ : a significant association exists between the different experts rankings

$\alpha := .05$ Level of significance for our test

$v := cols(R) - 1 \quad v = 3$ Degrees of Freedom for our test (number of objects ranked -1)

Determine the Critical Value for our Test

| | |
|---|---|
| $y := 10$ | initial guess at value |
| $crit := root(F(y,v) - (1 - \alpha), y)$ | Formula to find value where CDF = 1 - sig. level |
| $crit = 7.817$ | Critical value calculated above |

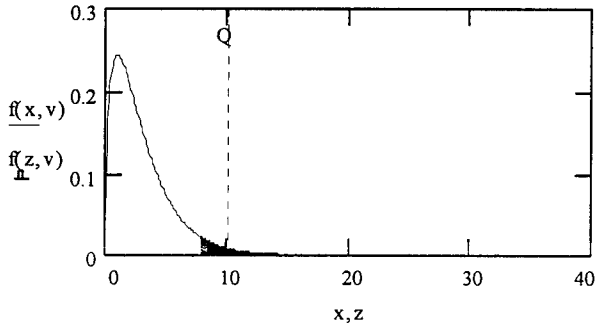
$x := 0, \delta .. 10 \cdot v$ range variable used for Chi-Squared Graph below

$z := crit, crit + \delta .. 5 \cdot v$ range variable used to shade the rejection region in the Graph below

Calculate the Q value that will fall either in our acceptance or rejection region:

| | |
|---|---|
| $k = 10$ | number of experts ranking objects |
| $v = 3$ | Degrees of Freedom (number of objects being ranked minus one) |
| $W = 0.332$ | Kendall Coefficient of Concordance for the set of rankings |
| $Q := k \cdot v \cdot W$ | Formula for calculating Q $\quad Q = 9.96$ |

Now we can graphically assess whether or not we should accept or reject our null hypothesis:




P-Value:

Assuming the null hypothesis is true, The P-value for our calculated Q value can be obtained by subtracting the Q value's cumulative probability from 1:

$1 - F(Q, v) = 0.019$

**Figure 11. Decision Graph for Survey Question 10**

Because the Q value is well within our rejection region also substantiated by a P-value significantly less than .05, we reject the null hypothesis and conclude that there is a significant association between the different experts rankings. We can further conclude that the order of preference for the scheme that the experts expect would be most cost beneficial is as follows:

$R = ( 26 \quad 35 \quad 18 \quad 21 )$ The column totals calculated on previous page

1) C
2) D
3) A
4) B

Question 11 concerning scheme that they would most favor for use in conjunction with a software safety program.

Schemes

$$\text{Rank} := \begin{bmatrix} 2 & 4 & 1 & 3 \\ 2 & 3 & 1 & 4 \\ 4 & 3 & 2 & 1 \\ 3 & 4 & 1 & 2 \\ 4 & 3 & 2 & 1 \\ 4 & 3 & 2 & 1 \\ 2 & 4 & 1 & 3 \\ 2 & 4 & 1 & 3 \\ 3 & 4 & 1 & 2 \\ 4 & 3 & 1 & 2 \end{bmatrix}$$

Experts

Define a ranking matrix with the columns representing the objects being ranked and the rows being the ranks the objects received from each expert

$\text{Rank} := \text{Rank}^T$ Matrix is transposed for calculation purposes.

$n := \text{rows}(\text{Rank})$ $n = 4$ number of objects being ranked

$k := \text{cols}(\text{Rank})$ $k = 10$ number of experts ranking objects

$R := \sum_{j=1}^{k} \text{Rank}^{\langle j \rangle}$ Sum of the columns representing the total score received by each object being ranked (the lower the total, the more popular the object)

$R := R^T$ Transpose the vector back into the familiar original

Column sums:

$R = ( 30 \quad 35 \quad 13 \quad 22 )$

Define the Kendall Coefficient of Concordance (value ranges between 0 and 1)

$$W := \frac{12 \cdot \sum_{z=1}^{\text{cols}(R)} R^{\langle z \rangle^2} - 3 \cdot k^2 \cdot n \cdot (n+1)^2}{n \cdot k^2 \cdot (n^2 - 1)} \qquad W = 0.556$$

Define the Probability Density Function and Cumulative Distribution Function for our Statistical Test:

Chi-Squared PDF:

$$f(x, v) := \frac{1}{2^{\frac{v}{2}} \cdot \Gamma\left(\frac{v}{2}\right)} \cdot x^{\frac{v}{2}-1} \cdot e^{-\frac{x}{2}}$$

Cumulative Chi-Squared Distribution Function:

$$F(hi, v) := \int_0^{hi} f(x, v)\, dx$$

Define the necessary parts for our statistical test:

$H_o$ : no significant association exists between the different experts rankings

$H_a$ : a significant association exists between the different experts rankings

$\alpha := .05$ Level of significance for our test

$v := cols(R) - 1 \quad v = 3$ Degrees of Freedom for our test (number of objects ranked -1)

Determine the Critical Value for our Test

| | |
|---|---|
| $y := 10$ | initial guess at value |
| $crit := root(F(y, v) - (1 - \alpha), y)$ | Formula to find value where CDF = 1 - sig. level |
| $crit = 7.817$ | Critical value calculated above |

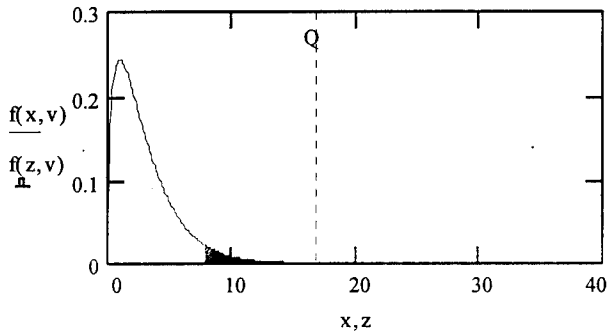
$x := 0, \delta .. 10 \cdot v$ range variable used for Chi-Squared Graph below

$z := crit, crit + \delta .. 5 \cdot v$ range variable used to shade the rejection region in the Graph below

Calculate the Q value that will fall either in our acceptance or rejection region:

| | |
|---|---|
| $k = 10$ | number of experts ranking objects |
| $v = 3$ | Degrees of Freedom (number of objects being ranked minus one) |
| $W = 0.556$ | Kendall Coefficient of Concordance for the set of rankings |
| $Q := k \cdot v \cdot W$ | Formula for calculating Q $\quad Q = 16.68$ |

Now we can graphically assess whether or not we should accept or reject our null hypothesis:




P-Value:

Assuming the null hypothesis is true, The P-value for our calculated Q value can be obtained by subtracting the Q value's cumulative probability from 1:

$1 - F(Q, v) = 0.001$

**Figure 12. Decision Graph for Survey Question 11**

Because the Q value is well within our rejection region also substantiated by a P-value significantly less than .05, we reject the null hypothesis and conclude that there is a significant association between the different experts rankings. We can further conclude that the order of preference for the schemes that they would most favor for use in conjunction with a software safety program is as follows:

$R = ( 30 \quad 35 \quad 13 \quad 22 )$ The column totals calculated on previous page

1) C
2) D
3) A
4) B

Overall Kendall Coefficient test to see if an association exists using the rankings established in the different questions where our tests showed a significant association exists among the expert rankings.

Schemes

$$Rank := \begin{bmatrix} 4 & 3 & 2 & 1 \\ 2 & 4 & 1 & 3 \\ 2 & 4 & 1 & 3 \\ 2 & 4 & 1 & 3 \\ 2 & 4 & 1 & 3 \\ 3 & 4 & 1 & 2 \end{bmatrix} \quad \begin{bmatrix} 2 \\ 3 \\ 4 \\ 5 \\ 9 \\ 10 \end{bmatrix} \text{ Questions}$$

Define a ranking matrix with the columns representing the objects being ranked and the rows being the ranks the objects received from each expert

$Rank := Rank^T$ Matrix is transposed for calculation purposes.

$n := rows(Rank)$ $n = 4$ number of objects being ranked

$k := cols(Rank)$ $k = 6$ number of experts ranking objects

$R := \sum_{j=1}^{k} Rank^{\langle j \rangle}$ Sum of the columns representing the total score received by each object being ranked (the lower the total, the more popular the object)

$R := R^T$ Transpose the vector back into the familiar original

Column sums:

$R = (15 \quad 23 \quad 7 \quad 15)$

Define the Kendall Coefficient of Concordance (value ranges between 0 and 1)

$$W := \frac{12 \cdot \sum_{z=1}^{cols(R)} R^{\langle z \rangle^2} - 3 \cdot k^2 \cdot n \cdot (n+1)^2}{n \cdot k^2 \cdot n^2 - 1} \qquad W = 0.711$$

Define the Probability Density Function and Cumulative Distribution Function for our Statistical Test:

Chi-Squared PDF:

$$f(x, v) := \frac{1}{2^{\frac{v}{2}} \cdot \Gamma\left(\frac{v}{2}\right)} \cdot x^{\frac{v}{2}-1} \cdot e^{-\frac{x}{2}}$$

Cumulative Chi-Squared Distribution Function:

$$F(hi, v) := \int_0^{hi} f(x, v)\, dx$$

Define the necessary parts for our statistical test:

$H_o$ : no significant association exists between the different experts rankings

$H_a$ : a significant association exists between the different experts rankings

$\alpha := .05$ Level of significance for our test

$v := cols(R) - 1 \quad v = 3$ Degrees of Freedom for our test (number of objects ranked -1)

Determine the Critical Value for our Test

$y := 10$ initial guess at value

$crit := root(F(y, v) - (1 - \alpha), y)$ Formula to find value where CDF = 1 - sig. level

$crit = 7.817$ Critical value calculated above

$x := 0, \delta .. 10 \cdot v$ range variable used for Chi-Squared Graph below

$z := crit, crit + \delta .. 5 \cdot v$ range variable used to shade the rejection region in the Graph below

Calculate the Q value that will fall either in our acceptance or rejection region:

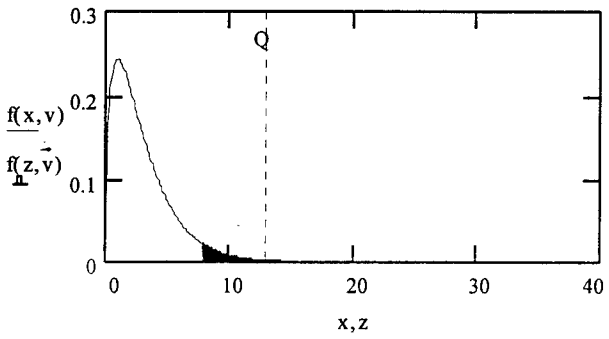
$k = 6$ number of experts ranking objects

$v = 3$ Degrees of Freedom (number of objects being ranked minus one)

$W = 0.711$ Kendall Coefficient of Concordance for the set of rankings

$Q := k \cdot v \cdot W$ Formula for calculating Q $Q = 12.8$

Now we can graphically assess whether or not we should accept or reject our null hypothesis:




P-Value:

Assuming the null hypothesis is true, The P-value for our calculated Q value can be obtained by subtracting the Q value's cumulative probability from 1:

$1 - F(Q, v) = 0.005$

**Figure 13. Decision Graph for Summary Test**

Because the Q value is well within our rejection region also substantiated by a P-value significantly less than .05, we reject the null hypothesis and conclude that there is a significant association between the different experts rankings. We can further conclude that the order of preference for the schemes taking the rankings from all questions indicating a significant association between expert's rankings is:

$R = ( 15 \quad 23 \quad 7 \quad 15 )$ The column totals calculated on previous page

1) C
2) A and D
3) B

# Appendix C: List of Experts Surveyed

This appendix contains an alphabetical list of the experts that participated in the survey. The order of this list does not correspond directly with the order of the experts' rankings for each question analyzed in Appendix B. One expert wished to remain anonymous and thus is not included in this list.

| | |
|---|---|
| Expert: | Dr. Stephen Cha |
| Organization: | Aerospace Corporation |
| DOD Affiliation: | Air Force |
| Expert: | Frank Foley |
| Organization: | Northrop Corporation |
| DOD Affiliation: | Air Force |
| Expert: | Dr. Michael Friedman |
| Organization: | Hughes |
| DOD Affiliation: | DOD |
| Expert: | Dr. Ross Grable |
| Organization: | Army Missile Command |
| DOD Affiliation: | Army |
| Expert: | William J. Kauffman |
| Organization: | Army Missile Command |
| DOD Affiliation: | Army |
| Expert: | Mitchell Lustig |
| Organization: | ASC System Safety Office |
| DOD Affiliation: | Air Force |
| Expert: | Captain Steve Mattern |
| Organization: | WL/XPN |
| DOD Affiliation: | Air Force |
| Expert: | Dr. Tim Shimeall |
| Organization: | Naval Post Graduate School |
| DOD Affiliation: | Navy |
| Expert: | Eileen Takach |
| Organization: | Naval Air Warfare Center |
| DOD Affiliation: | Navy |

THIS PAGE INTENTIONALLY LEFT BLANK.

# Bibliography


AF Pamphlet 63-115. Guidelines for Successful Acquisition and Management of Software Intensive Systems, Final Draft. November 1993.

Basili, V. "Software Errors and the Complexity, An Empirical Investigation," Proceedings of the Seventh Annual Software Engineering Workshop. Goddard Space Flight Center. December 1, 1982.

Beizer, B. Software Testing Techniques. New York: Van Nostrand Reinhold, 1983.

Colan, Peter W. and Robert W. Prouhet. An Assessment Of Software Safety As Applied to the Department of Defense Software Development Process. MS Thesis. AFIT/GSS/ENG/92D-2, School of Acquisition and Logistics Management, Air Force Institute of Technology, Dayton OH, 1992 (AD-A258155).

Collofello, J. S. and B. P. Gosalia. "An Application of Causal Analysis to the Software Modification Process," Software-Practice and Experience, 23(10): 1095-1105 (October 1993).

Collofello, J. S. and L. B. Blumer. "A Proposed Software Error Categorization Scheme," Proceedings of the National Computer Conference. 537-545. AFIPS Press, 1985.

Collofello, J. S. and L. B. Blumer. A General Scheme for Software Error Data Collection. Arizona State University Computer Science Technical Report, June 1983.

Davis, F. and R. Gantenbein. "Responding to Catastrophic Errors: A Design Technique for Fault-Tolerant Software," Journal of Systems Software, 17: 243-251 (1992).

Dunn, Robert H. "The Quest for Software Reliability," Handbook of Software Quality Assurance. New York: Van Nostrand Reinhold, 1987.

Emory, C. William and Donald R. Cooper. Business Research Methods (Fourth Edition). Homewood IL: Richard D. Irwin, Inc., 1991.

Endres, A. "An Analysis of Errors and Their Causes in System Programs," IEEE Transactions on Software Engineering, SE-1(2):140-149 (June 1975).

Gellman, Barton. "Computer Problem Cited in Crash of F-22 Prototype," Washington Post, 115: A3 (30 April 1992).

Gibbons, Jean Dickinson. Nonparametric Methods for Quantitative Analysis. Atlanta GA: Holt, Rinehart, and Winston, 1976.

Jones, C. L. "A Process-Integrated Approach to Defect Prevention," IBM Systems Journal, 24(2): 150-166 (1985).

Jorgens III, J. and J. Greenbaum. "Software Quality Assurance and System Safety," Journal of Clinical Engineering, 13: 196 (1988).

Leveson, N.G. "Software Safety: Why, What, and How", ACM Computing Surveys, 18: 25-69 (June 1986).

Leveson, Nancy G. "Software Safety in Embedded Computer Systems," Communications of the ACM, 34: 35-46 (February 1991).

Lipow, M. "Prediction of Software Failures," Journal of Systems Software, 1: 71-75 (1979).

Maxwell, F. D. The Determination of Measures of Software Reliability. Final Report. The Aerospace Corporation. 1979 (NASA-CR-158960).

Nakajo, T. and H. Kume. "A Case History Analysis of Software Error Cause-Effect Relationships," IEEE Transactions on Software Engineering, 17: 830-837 (August 1993).

Neumann, Peter G. "RISKS: Cumulative Index of Software Engineering Notes," ACM SIGSOFT Software Engineering Notes, 14: 22-26 (January 1989).

Ostrand, Thomas J. and Elaine J. Weyuker. "Collecting and Categorizing Software Error Data in an Industrial Environment," The Journal of Systems and Software, 4: 289-300 (1984).

Parnas, David L., A. John van Schouwen, and Shu Po Kwan. "Evaluation of Safety-Critical Software," Communications of the ACM, 33: 636-648 (June 1990).

Piechota, Charles L. "The Twilight Zone," Professional Safety, 37: 32-35 (January 1992).

Russo, Leonard L. Software System Safety Guide. CECOM-TR-92-2. Fort Monmouth NJ: US Army Communications-Electronics Command, May 1992 (AD-A250321).

Thayer, T. A. Software Reliability Study. Final Technical Report. Griffis NY: TRW Defense and Space Systems 1976 (RADC-TR-76-238).

Wiener, Lauren R. Digital Woes. Reading MA: Addision-Wesley Publishing Co., 1993.

# Vita - Captain Richard Escobedo

Captain Escobedo was born in San Antonio, Texas, on 15 September 1964. He attended Holy Cross High School in Texas where he excelled in academic as well as extra-curricular activities, graduating second in his class in 1983. He attended The University of Texas at San Antonio, San Antonio, Texas, graduating with a Bachelor of Science in Electrical Engineering in May 1988.

Captain Escobedo's commission from Reserve Officer Training Corps was on 20 May 1988. For his first assignment, Captain Escobedo moved to Edwards AFB, CA to work with the 412th Test Wing as a flight test engineer. His duties included mission support flying, program management of the aerial refueling test capability, and the upgrade of the support fighter and pacer fleet. He was hand-picked to receive training as a mission support flyer with the Advanced Cruise Missile Chase Program (ACM), where he earned his non-rated mission support flyer wings. He performed test operations support with the ACM program in the F-4C and KC-135 aircraft. As manager for the Air Force's aerial refueling instrumentation test capability, Captain Escobedo contributed to the successful completion of developmental test and evaluation of major aircraft programs. These programs include the B-2 Stealth Bomber, the C-17 Airlift Transport, and the F-22 Advanced Tactical Fighter programs. Aerial refueling testing was conducted from modified KC-135 and KC-10 aircraft, which were maintained by his instrumentation team. He modernized the Air Force Flight Test Center's (AFFTC) general support and pacer fleets. Aging A-7 and F-4 test support aircraft were replaced with modified F-16 and F-15 aircraft that enabled the AFFTC to continue its test mission well into the next century. He was selected to attend the Air Force Institute of Technology, in May 1992.

Permanent Address:
Richard Escobedo
174 Honey Jay
San Antonio, TX 78228

# Vita - Captain Jim Thomas

Captain Thomas was born in Colorado Springs, Colorado, on 29 July 1966. He attended Junius H. Rose High School in Greenville, North Carolina. In August 1988, he obtained an ROTC commission along with a Bachelor of Science degree in Mathematics from North Carolina State University.

Captain Thomas's first assignment was to Columbus AFB, Mississippi for Undergraduate Pilot Training in March 1989. After six months of flight training, he was assigned to Los Angeles AFB, California in the Satellite Communication Program Office. His first job, as a production acquisition manager, entailed coordinating technical, contractual, budgetary, and program management aspects of military acquisition with other program office members. This involved leading several government teams through the negotiation of new contracts as well as changes to existing contracts. He negotiated numerous contractor proposals involving satellite launch operations and on-orbit orbital support. After two years, Captain Thomas became the DSCS III orbital support manager. He was responsible for coordinating the activities of many organizations in order to maintain an operational constellation of communication satellites. He also participated in the successful launch of three satellites that will provide military communications well into the next century. In May 1993, Captain Thomas was assigned to the Air Force Institute of Technology.

While in Los Angeles, Captain Thomas married his wife, Deborah. He learned what true happiness was after their marriage in December 1991. He is looking forward to many happy years with Deborah at his side.

Permanent Address:

Jim Thomas
2000 Brook Road
Greenville, NC 27858

# REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188



| 1. AGENCY USE ONLY (Leave blank) | 2. REPORT DATE | 3. REPORT TYPE AND DATES COVERED |
|---|---|---|
| | December 1994 | Master's Thesis |

**4. TITLE AND SUBTITLE**
A COMPARISON OF ERROR CATEGORIZATION SCHEMES FOR USE IN SOFTWARE SYSTEM SAFETY PROGRAMS

**5. FUNDING NUMBERS**

**6. AUTHOR(S)**
Richard Escobedo, Captain USAF
Jim Thomas, Captain USAF

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**
Air Force Institute of Technology,
WPAFB OH 45433-6583

**8. PERFORMING ORGANIZATION REPORT NUMBER**
AFIT/GSS/LAR/94D-1

**9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)**
ASC/EMSS
Wright Patterson AFB, OH 45433

**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**

**11. SUPPLEMENTARY NOTES**

**12a. DISTRIBUTION / AVAILABILITY STATEMENT**
Approved for public release; distribution unlimited

**12b. DISTRIBUTION CODE**

**13. ABSTRACT** *(Maximum 200 words)*

Software safety is becoming increasingly important in the development of DOD advanced weapon systems. To make software safer, hazard conditions must be avoided along with the errors that accompany them. The first step in identifying errors is classifying error data. The area of software error classification is not as advanced as other software development areas. The technical literature lacks examples of comprehensive taxonomies that can be applied to various computer software domains and applications. The predominant approach is to organize errors into categories particular to the program currently in work. The typical error scheme is made of narrow categories that are not interrelated. Errors have been classified by symptom, by cause, by life cycle phase, by severity, and by software control. The focus of this research was to determine the best way to classify errors in order to aid system safety in software development. The research identified common areas used in industry that aid in error classification. A telephone survey of experts in safety and software was used to obtain input on the most effective classification schemes. The research also proposed a taxonomy that will be ideal for DOD software development. Since software is becoming a larger part of advanced weapon systems, development of error-free and safe software to operate and support these weapon systems is increasingly important.

| 14. SUBJECT TERMS | 15. NUMBER OF PAGES |
|---|---|
| Software Safety, Software Errors, Error Categorization | 91 |
| | 16. PRICE CODE |

| 17. SECURITY CLASSIFICATION OF REPORT | 18. SECURITY CLASSIFICATION OF THIS PAGE | 19. SECURITY CLASSIFICATION OF ABSTRACT | 20. LIMITATION OF ABSTRACT |
|---|---|---|---|
| Unclassified | Unclassified | Unclassified | UL |