A Proposed Research Programme for Determining a Metric for a Good Requirement

Joseph Kasser, William Scott, Xuan-Linh Tran, Sergey Nesterov
Systems Engineering and Evaluation Centre
University of South Australia
Mawson Lakes, SA 5095
Joseph.Kasser@unisa.edu.au

Abstract

Poor requirements have been cited as a major cause of project failures and cost and schedule overruns in the acquisition of Defence and information systems for at least 15 years. Yet in all this time, academia has not been able to alleviate the problem through traditional teaching techniques. Moreover the commercially available requirements tools manage sets of requirements but do not provide the capability to determine if a requirement is good or bad. This paper summarizes the concept behind a research programme to determine a metric for the goodness of a requirement that could be used in industry and the classroom as well as being built into a software tool.

Background

Poor requirements have been cited as a major cause of project failures and cost and schedule overruns in the acquisition of Defence and information systems for at least 15 years (Carson, 2001; Hooks, 1993; Jacobs, 1999; Kasser and Schermerhorn, 1994a; Lindquist, 2005) etc. Yet in all this time, academia has not been able to alleviate the problem through traditional teaching techniques, nor have commercial requirements engineering and management tool producers built the capability to distinguish between good and bad requirements into their tools. There is thus a need for a metric for the goodness of requirements that can be used in industry, the classroom, and be integrated into requirements engineering tools. This proposed research builds on published prior research into alleviating the problem of poor requirements and will tackle the problem from the following perspectives:

- **The test and evaluation (T&E).** A perspective based on useability testing which shows the type of problems with the document which allows an analysis of the cause of the problem so as to be able to fix them (Redish, 2000).
- **The readability and understandability of a document.** A perspective based on a number of readability metrics. However, their validity to technical documents such as requirements documents is unknown (Redish, 2000).
- **Modifying a metric developed to measure the complexity of software source code**. A perspective based on the apparent similarly between text-based requirements and software source code.
- **Number of linguistic elements in the requirement statement**. This is a simple approach, but may be all that is needed.
- **The difference between the goodness** of an individual requirement and the goodness of the set of requirements.

The Test and Evaluation perspective

An early attempt to alleviate the problem of poor requirements from the T&E perspective was a postgraduate Requirements Workshop for the writing of good requirements developed at University of Maryland University College (UMUC) in 1998 (Kasser and Williams, 1999) and later incorporated into classes at UniSA. Its roots stemmed from research into applying Total Quality Management (TQM) when (Kasser and Schermerhorn, 1994b) had showed how poorly worded requirements were able to easily add $500,000 to a project's cost and identified an initial set of requirements for writing requirements to prevent that cost escalation which were later updated and published (Kasser, 1995). Thus written requirements that do not meet the "requirements for writing requirements" are poorly written requirements – by definition. The workshop entailed students reviewing and commenting on the contents of a sample System Requirements Document (SRD). The workshop was limited by time so that only a few pages of a document were reviewed by the students. The focus of the discussion during the workshop was on the grammar and structure of requirements. Moreover, when students discussed their findings of good and bad requirements, there was little if any consensus on the quality of the requirements.

Realizing that the parsing process which was taking up much of the student's time in the workshop could be automated, a prototype software tool was developed to perform the function (Kasser, 2002). The tool parsed the text of the requirement looking for the presence of "poor words" that indicated that a defect or a potential for a defect existed. The list of poor words was gathered from experience and the literature (Hooks, 1993; Kar and Bailey, 1996; Kasser and Schermerhorn, 1994a). The tool evolved into FRED (Kasser, 2004), then TIGER and finally into TIGER Pro (Kasser, Tran and Matisons, 2003). (Kasser, 2004) discussed the five categories of poor words shown in Table 1 and their effect on the systems and software engineering process providing the sample subset of "poor words" found in requirements documents shown here in Table 2.

Table 1 Categories of defects in Requirements

Category	Defect
1	Multiple requirements in a requirement
2	Possible multiple requirement
3	Not verifiable
4	Use of wrong word
5	User defined poor word

While some of the words in Table 2 may have meanings which can be carefully defined, in most instances they are not carefully defined, so it is better to avoid them, and use words that have verifiable meanings. (Kasser, 2004) discussed how the types of defects listed in Table 1 affect the process in which the requirements are used as follows.

- **Multiple requirements in a requirement.** More than one requirement in a requirement complicates the building of traceability matrices causing difficulties in the T&E processes. By definition, if the word "shall" appears more than once in a requirement, there are multiple requirements in the requirement and hence this situation is a defect in the requirement. Correcting this defect means splitting the multiple-requirement into separate individual requirements.
- **Possible multiple requirement.** Words such as "and" and "or" indicate that there may be multiple requirements in a requirement. For example, consider the use of the word "and" in a

requirement. A requirement such as

```
"DADS shall display the number of requests pending and requests
processed"
```

is two requirements, the first to display the number of requests pending, and the second to display the number of requests processed. This is a defect that must be corrected by writing the requirement as two requirements. On the other hand, a requirement such as

```
"DADS shall display the combined number of requests pending and
requests processed"
```

is a requirement to display a single total, hence the use of the word "and" is appropriate and is not a defect.

Table 2 Partial List of "Poor Words" in Requirements

Poor Word	Occurrence	Category	Category of Defect
Adequate	0	3	Descriptive, not verifiable
And	0	2	Possible multiple requirement paragraph
Appropriate	0	3	Descriptive, not verifiable
Best practice	0	3	Descriptive, not verifiable
But not limited to	0	3	Unspecified superset, not verifiable
Easy	0	3	Descriptive, not verifiable
For example	0	3	Descriptive, not verifiable
Including	0	3	Unspecified superset, not verifiable
Large	0	3	Descriptive, not verifiable
Many	0	3	Descriptive, not verifiable
Maximize	0	3	Descriptive, not verifiable
Minimize	0	3	Descriptive, not verifiable
Must	0	4	Use of wrong word
Or	0	2	Possible multiple requirement paragraph
Quick	0	3	Descriptive, not verifiable
Rapid	0	3	Descriptive, not verifiable
Shall	**1**	**1**	**Multiple requirements in requirement**
Should	0	4	Use of wrong word
Sufficient	0	3	Descriptive, not verifiable
User-friendly	0	3	Descriptive, not verifiable
Will	0	4	Use of wrong word

- **Not verifiable.** Some words are not verifiable or testable. Words that fall into this category include – large, many, few, including, etc. Some words may be understood at the time and in the context that they are used, but have no meaning further down the schedule when the requirement writers have moved on to other projects e.g. "statistically monitor" (STDADS, 1992).
- **Use of wrong word.** (MIL-STD-961D, 1995) requires the use of the word "shall" to identify a requirement. The words "will", "should", and "must", may be valid in descriptions but are not to be used in requirements.
- **User defined poor word**. This category is provided to allow the user to define words that are "poor words" in the user's organisation.

(Kasser, 2004) also described the development of a Figure of Merit (FOM), a simple one-dimensional measurement for the quality of a document based on the presence or absence of "poor words" derived from the work of (Juran, 1988) which allows comparisons to be made of the quality of documents of different sizes. The FOM was calculated using the formula

*FOM = 100 - (number of defects / number of requirements) * 100*

This formula results in a FOM of 100 for a document that contains zero defects, and a negative number for a document containing more defects than requirements since a single requirement may contain more than one defect. The functionality to provide a FOM for a set of requirements was built into FRED and its successors and used in the classroom. After a few instances of use, the focus of the discussion in the workshop changed from the structure and grammar of a requirement to the difficulty of writing requirements (Kasser, Tran and Matisons, 2003). This is a significant shift in perspective.

The readability and understandability of a document perspective

As discussed above, the FOM is based on a set of "requirements for writing requirements". One of those meta-requirements is "*a requirement shall not be vague*". Antonyms of vague are clear and understandable. Hence there should be no surprise to learn that in a later listing of the criteria of a good requirement (Schneider and Buede, 2000), one criterion was understandability. However, there does not seem to be any way of directly measuring "vague" and "understandability".

A number of different metrics for the readability of documents have been developed over the years (McCallum and Peterson, 1982) and other research efforts (Alvarez, Castell and Slavkova, 1996; Bellagamba, 2001; James, 1999; Kenett, 1996; Robertson and Robertson, 1999; Wilson, Rosenberg and Hyatt, 1997) have provided statistical information about the quality of requirements but do not seem to be doing much with the resulting information. This may have been because raw readability metrics are not suitable for providing useful information about the goodness of requirements. For example, (Redish and Selzer, 1985) discussed the following five significant problems with readability formulas and the way they were used in technical writing for adults:

1. Readability formulas have been applied to technical and business writing with no research basis.
2. Studies show that readability formulas are not reliable and valid predictors of how understandable a technical, scientific, or legal document will be for adults.
3. Shortening sentences and words does not necessarily make the sentences and words easier to understand.
4. The underlying assumption of readability formulas—that any text for any reader for any purpose can be [measured by a single] equation does not mesh with our current understanding of how people process information.
5. Readability formulas do not take into account many features that are critical to people's ability to understand and use documents.

The Flesch-Kincaid Grade Level. The most commonly used readability formula is the Flesch-Kincaid Grade Level (Kincaid, 1975). It is a measure of the years of US schooling needed to be able to read a document. It produces a numerical value that ranges from 0 to 12 which represents a readability level from Kindergarten to Grade 12. The formula for the Flesch-Kincaid Grade Level score is (.39 x ASL) + (11.8 x ASW) – 15.59 where:

- ASL = average sentence length (the number of words divided by the number of sentences)
- ASW = average number of syllables per word.

The Flesch-Kincaid Grade Level score has already been tried on requirements with mixed results (Tran and Kasser, 2005). When TIGER Pro was used in a Requirements Workshop in mid May 2005 in which there were 85 participants who were professionals in the Defence Industry involved in writing or using requirements, pre- and post- workshop questionnaires were handed out to participants[1]. The questionnaires comprised a set of 37 good and defective requirements obtained from a 20-year old NASA Goddard Space Flight Center (GSFC) system requirements document. For the pre-workshop assessment, participants were required to read each requirement and decide if it was good or bad. In the post-workshop assessment, the same set of requirements was provided but the order was different. In addition, not only were the participants required to read each requirement and decide if it was good or bad, they were also required to state the type of defect (as discussed above) they found in the requirements.

There was an improvement of 71% is the determination of good and bad requirements post workshop. However, it was also noticed that (1) the determination of the goodness of some requirements proved to be difficult for most participants and (2) there was some confusion about the type of defect in a defective requirement. To assess whether the level of reading difficulty in the requirements given in the workshop had any affect on the participants' responses, both the pre- and post-workshop correct answers for each requirement were plotted against the Flesch-Kincaid Grade Level readability score provided in Microsoft Word for that requirement as shown in Figure 1 and Figure 2 (Tran and Kasser, 2005).

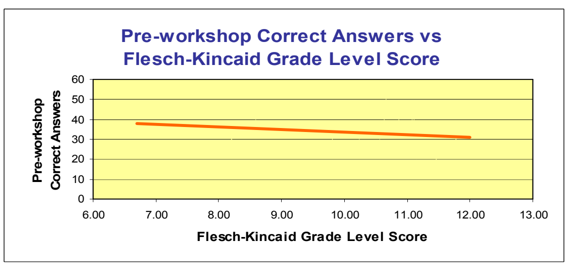

Figure 1 Pre-workshop correct answers vs. Flesch-Kincaid Grade Level Score

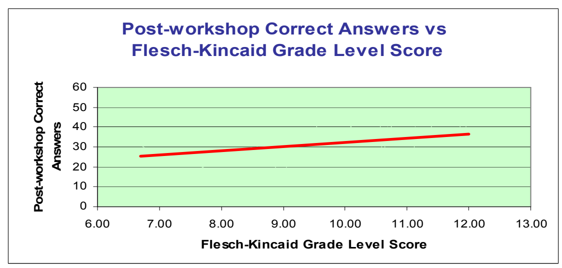

Figure 2 Post-workshop correct answers vs. Flesch-Kincaid Grade Level Score

The following deductions can be made from Figure 1 and Figure 2.

- In the pre-workshop survey (Figure 1), as the reading grade level of the requirements increases, the number of correct answers decreases.
- In the post-workshop survey (Figure 2), as the reading grade level of the requirements increase, the number of correct answers also increases.

The first finding seems to be intuitive once articulated. However, the only reference to the readability a requirement that has been found in the systems engineering literature is (Wilson, Rosenberg and Hyatt, 1997) which discusses the development of an Automated Requirements Measurement (ARM) tool by the Software Assurance Technology Center at the GSFC.

The second finding needs to be researched. The obvious conclusion is that the workshop focussed on the more complicated statements and improved their comprehension. However, this is more likely support for the finding that readability metrics may not be suitable for providing useful information about the goodness of requirements (Redish and Selzer, 1985).

[1] This was the first use of such a questionnaire in one of the workshops

Other readability indexes are also computed by formulae designed for computer software (McCallum and Peterson, 1982). In general we tend to measure things that are easy to measure, so a small amount of effort will be expended in modifying TIGER to incorporate other readability indices designed for English such as the Gunning-Fog index (Gunning, 1952), the Simple Measure of Gobbledegook (SMOG) index (McLaughlin, 1969), the Coleman-Liau Index (Coleman and Liau, 1975) the Automated Readability Index (Kincaid, 1975) and the Homan-Hewitt Readability Formula which estimates the readability level of single-sentence text items (Homan, Hewitt and Linder, 1994), as well as the Fernandez-Huerta index for Spanish (Fernández-Huerta, 1959), and the Douma index (Douma, 1959) for Dutch. The structure of French and Dutch differs with that of English so the relative scores might provide some useful information. These measurements can serve as a point of reference. (Si and Callan, 2001) developed an alternate statistical approach to measuring the readability of technical pages which considers the content as well as the language that looks promising and needs to be researched to determine its applicability in measuring the readability of requirements.

Modifying a metric developed to measure the complexity of software source code.

While there are formal languages for requirements, their use is resisted even though requirements are written in a semi-formal language or subset of natural English. Thus from the perspective of this research, the text of requirements can be treated and hence measured in a similar manner to computer software source code which is written in a semi-formal "descriptive" language. Metrics such as the Halstead Metrics (Halstead, 1977) that consider content as well as the language are used to measure the complexity of software source code. However we need to determine if the measurement provides anything useful. The research question here is "can we modify the Halstead Metrics or a similar metric to produce a metric for the understandability of requirements?" A conceptual approach for developing a one-dimension formula is shown in Table 3. However, it is likely that we would need different word classification (grouping) and the understandability index would be based on a weighting formula or a multi-dimension metric, where different groups contribute different weights to the index in the manner of the Dale-Chall index (Dale and Chall, 1948) which counts sentence length and whether a word appears on a list of 3000 acceptable words (SRI, 2000). The research approach will be to create (predefined or automatically generated) dictionaries in the manner of FRED, grouped as "poor (vague) words", "acronyms", and "key words". These dictionaries will create the context for the measurement of the goodness of single requirements. The research will investigate:

- How many groups of words are required to create understandability index? For example, "poor words" group should be identified and considered as reducing …. factor for understandability.
- How to create (define or build) each dictionary?

Number of linguistic elements in the statement

The length of a statement is a function of the number of elements, subject, verb, object, qualifiers, etc. it may be possible to develop a simple usable metric based on the number of elements in the statement. This avenue will also be explored, incorporated into TIGER and the resulting scores compared with those generated by the readability indices.

Table 3 Suggestions for modifications to the Halstead Metrics for the measurement of the understandability of the text of requirements

Halstead Notation	Programming language equivalent	Text of requirement
n_1 – unique operators	Unique variables	Nouns or just keywords from the domain?
n_2 – unique operations	Unique operations (operators) , like +, - and keywords, like if, case …	All verbs or just specific ones, such as “shall”?
n = n1 + n2 Vocabulary	Sum of unique operands and operations	To be determined.
N1 – total number of operands	All variables in the code	All keywords or all words?
N2 – total number of operations	All operations in the code	All verbs or specific ones?
N=N1+N2	Program length	Length of the requirement text
V= N*log (n)	Program volume (or information in the code)	V=N * log (n) – Does this correlate with understandability?

The difference between the goodness of an individual requirement and the goodness of the set of requirements

From the readability perspective we are looking for good individual requirements as well as a good set of requirements. From the useability perspective we are looking at good individual requirements. Consider each in turn:

- **The requirement statement**. The statement is written in semi-formal English based upon a stylised version of English grammar. This stylisation must conform to standards for writing requirements and rules for writing technical documents. This compliance has the effect of removing some of the less predictable styles used in writing prose. The content is the particular words used in the sentence. Standards also play an important role in the selection of words and the subsequent meaning they convey. An example of this is the use of the verb “shall” over alternatives, such as “will” or “must” (MIL-STD-490A, 1985; MIL-STD-961D, 1995) Section 3.2.3.6).
- **The requirements set**. This level focuses on the combined meaning of the requirements and how well the set reflects the real needs of the customer because if the true need is not identified, the system may not meet the need, and if the system does not meet the need, the effort spent developing it was wasted (Kasser, 1995). Thus the quality of the need identification determines to a large degree the suitability of the system for the intended use (Ansorge, 2000).

The criteria to be assessed when examining requirements can be used to illustrate this distinction. (Schneider and Buede, 2000) examined various methods and texts to create a list of criteria of a good requirement. However, this list can be sorted to distinguish between criteria for individual requirements and criteria for sets of requirements. For example, criteria such as ‘consistency between requirements’ and ‘non contradictory’ apply to the set of requirements while criteria such as ‘verifiable’ and ‘understandable’ apply to individual requirements.

The metric developed from this perspective will combine the attributes of a metric for the readability and understandability of individual requirements as well as the attributes of a metric for the set of requirements.

The research approach

A number of ways of developing metrics have been proposed. We must not lose touch with the goal of this research which is to produce a metric to provide a measure the goodness of requirements. Readability and understandability are just attributes of goodness. The FOM has been used in FRED and its successors TIGER and TIGER Pro to change the perception of students about the difficulty of writing requirements (Kasser, Tran and Matisons, 2003). Thus the FOM which is simple to program may be all that is needed to use as a measure of the understandability of requirements.

From the perspective of the readability metrics discussed above. Firstly they may not be suitable at all. Secondly, we cannot improve the readability of a requirement simply by shortening words and sentences (Redish and Selzer, 1985). In the wording of requirements, short words such as “it” and acronyms, while improving the readability score tend to reduce the understandability of the text. The pronouns have the potential to introduce ambiguity and unfamiliar acronyms increase the difficulty of understanding the text hence lower the understandability of the requirement text. Thus well-written requirements may not have good readability scores.

The proponents of readability metrics caution that text should not be written to achieve a high readability score. However, in the case of the metric for the goodness of requirements the situation is reversed. The requirement is that the text of a requirement shall be written so as to achieve a perfect “goodness” score. Thus the proposed avenue of research mixes readability formulae with usability testing and is as follows:

1. Develop a series of standardized reference requirement documents of various qualities.
2. Incorporate the measurement approaches discussed above into a skeleton version of TIGER.
3. Run tests on the series of standardized reference requirement documents.
4. Determine which of the measurement approaches provide the best metric for the goodness of the requirement text.
5. Determine what mixture of metrics produces the best result.
6. Run the best metric on a series of documents to verify the metric.
7. Incorporate the best metric into TIGER Pro.

Summary

This paper discussed the need for good requirements, briefly looked at the difference between readability and understandability and proposed a research program to develop a metric for the “goodness” of requirements.

References

Alvarez, J., Castell, N. and Slavkova, O., "*Combining Knowledge and Metrics to Control Software Quality Factors"*, Third International conference on Achieving Quality in Software,, 1996, pp. 201-212.

Ansorge, W. R., "*Assembly, integration, verification, and validation in extremely large telescope projects: A Core Systems Engineering Task"*, Proc. of SPIE, 4004, Apr 2000, 2000.

Bellagamba, "*Program to Identify Potential Ambiguities in Requirements Written in English"*, the 11th International Symposium of the International Council on Systems Engineering, 2001.

Carson, R. S., "*Keeping the Focus During Requirements Analysis*", the 11th International Symposium of the International Council on Systems Engineering, 2001.

Coleman, M. and Liau, T. L., "*A computer readability formula designed for machine scoring*", Journal of Applied Psychology, no. 60, 283-284, 1975

Dale, E. and Chall, J. S., "*A Formula for Predicting Readability*", Educational Research Bulletin, **27**, no. January 21 and February 17, 11-20; 37-54, 1948.

Douma, W. H., "De Leesbaarheid van Landbouwliteratuur," Landbouwhogeschool Wageningen (unpublished), 1959.

Fernández-Huerta, J., "*Medidas sencillas de lecturabilidad*", Consigna, no. 214, 29-32, 1959.

Gunning, R., *The Technique of Clear Writing*, McGraw-Hill, 1952.

Halstead, M. H., *Elements of Software Science*, North-Elsevier, Holland, New York, 1977.

Homan, S., Hewitt, M. and Linder, J., "*The Development and Validation of a Formula for Measuring Single-Sentence Test Item Readability*", Journal of Educational Measurement, Volume 31, no. 4, 349-358, 1994.

Hooks, I., "*Writing Good Requirements*", Proceedings of the 3rd NCOSE International Symposium, 1993.

Jacobs, S., "*Introducing Measurable Quality Requirements: A Case Study*", the IEEE International Symposium on Requirements Engineering, 1999.

James, L., "*Providing Pragmatic Advice On How Good Your Requirements Are - The Precept "Requirements Councillor" Utility*", The 9th INCOSE International Symposium, 1999.

Juran, J. M., *Juran on Planning for Quality*, The Free Press, 1988.

Kar, P. and Bailey, M., "*Characteristics of good requirements*", 6th International Symposium of the International Council on Systems Engineering (INCOSE), 1996.

Kasser, J. E., *Applying Total Quality Management to Systems Engineering*, Artech House, Boston, 1995.

Kasser, J. E., "*A Prototype Tool for Improving the Wording of Requirements*", Proceedings of the 12th International Symposium of the INCOSE, 2002.

Kasser, J. E., "*The First Requirements Elucidator Demonstration (FRED) Tool*", Systems Engineering: The Journal of the International Council on Systems Engineering, Volume **7**, no. 3, 2004.

Kasser, J. E. and Schermerhorn, R., "*Determining Metrics for Systems Engineering*", The 4th Annual International Symposium of the NCOSE, 1994a.

Kasser, J. E. and Schermerhorn, R., "*Gaining the Competitive Edge through Effective Systems Engineering*", The 4th Annual International Symposium of the NCOSE, 1994b.

Kasser, J. E., Tran, X.-L. and Matisons, S., "*Prototype Educational Tools for Systems and Software (PETS) Engineering*", Proceedings of the AAEE Conference, 2003.

Kasser, J. E. and Williams, V. R., "*The Student Enrolment and Course Tracking System Meta-Project*", Portland International Conference on Management of Engineering and Technology (PICMET), 1999.

Kenett, R. S., "*Software Specification Metrics: A Quantitative Approach to Assess the Quality of Documents*", Nineteenth Convention of Electrical and Electronics Engineers in Israel, pp. 166 -169, 1996.

Kincaid, J. P., "Derivation of New Readability Formulas (Automated Readability Index, Fog Count and Flesch Reading Ease Formula) for Navy Enlisted Personnel," National Technical Information Service, Springfield, 1975.

Lindquist, C., "*Fixing the Requirements Mess*", CIO Magazine, 2005.

McCallum, D. R. and Peterson, J. L., "*Computer-based readability indexes"*, ACM'82 Conference, Association for Computing Machinery, 1982, pp. 44 – 48, 1982.
McLaughlin, G. H., "*SMOG Grading: A New Readability Formula"*, Journal of Reading, Volume 12, no. 8, 639-646, 1969.
MIL-STD-490A, *Specification Practices*, Department of Defense, 1985.
MIL-STD-961D, *Department Of Defense Standard Practice For Defense Specifications*, 1995.
Redish, J., "*Readability Formulas Have Even More Limitations Than Klare Discusses"*, ACM Journal of Computer Documentation, Volume 24, no. 3, 132-137, 2000.
Redish, J. C. and Selzer, J., "*The Place of Readability Formulas in Technical Communication"*, Technical Communication, Fourth Quarter, 1985.
Robertson, S. and Robertson, J., "*Reliable Requirements Through the Quality Gateway"*, 10th International Workshop on Database and Expert Systems Applications, 1999.
Schneider, R. E. and Buede, D. M., "*Properties of a High Quality Informal Requirements Document"*, the 10th International Symposium of the INCOSE, 2000.
Si, L. and Callan, J., "*A statistical model for scientific readability"*, the tenth international conference on Information and knowledge management, ACM Press, 2001.
SRI, "The ATOS™ Readability Formula for Books and How it Compares to Other Formulas," School Renaissance Institute, Madison, WI, 2000.
STDADS, *ST DADS Requirements Analysis Document (FAC STR-22), Rev. C, August 1992, as modified by the following CCR's:- 139, 146, 147C, 150 and 151B*, NASA/Ford AeroSpace, Greenbelt, MD, 1992.
Tran, X.-L. and Kasser, J. E., "*Improving the recognition and correction of poorly written requirements"*, the Systems Engineering Test and Evaluation (SETE) Conference, 2005.
Wilson, W. M., Rosenberg, L. H. and Hyatt, L., "*Automated Analysis of Requirements Specifications"*, the IEEE International Conference on Software Engineering, 1997.