

Software tools for improving engineering education in the postgraduate seminar environment

Joseph Kasser DSC, CEng, CM.
Systems Engineering and Evaluation Centre
University of South Australia, Adelaide Australia
Joseph.Kasser@unisa.edu.au

Abstract. Postgraduate engineering education tends to be delivered to part time students who are full time employees of government and industry. This paper first describes a variety of tools which the author has developed and used to enhance the learning experience in classes in systems and software engineering first at University of Maryland University College (UMUC) and later in the University of South Australia (UniSA). The paper then summarizes some of the experience of using the tools.

INTRODUCTION

Postgraduate engineering education tends to be delivered to part time students who are full time employees of government and industry. This paper presents an overview of five tools developed to improve engineering education and summarizes the experiences of using the tools in traditional face to face and distance mode classes in systems and software engineering at UMUC and UniSA. The tools described in this paper are:

- **TIGER Pro** - developed as a research tool at UniSA but has also been found useful in teaching students how to write good requirements.
- **Categorized Requirements in Process (CRIP) charts** - developed at UMUC to provide information about the state of projects based on summaries of types of work completed as a project progresses.
- **Program-Activity-Milestone (PAM) charts** - developed prior to entering academia to assist anyone in the initial planning stages of a project.
- **Audio enhancement to PowerPoint presentations** - developed to provide the full presentation experience in the Asynchronous Learning Network

(ALN). Experiences are related from both student and instructor perspectives regarding several different ways of encoding the audio.

- **The WebForum** - developed by students at UMUC as an asynchronous web-based online conference site suitable for conferences as well as classrooms.

Having watched students in several courses struggle to learn how to use a commercial tool and do a practical exercise at the same time, in developing these tools the prime directive was that the user's (student) learning time should be minimal, and the tools be as easy to use as was a slide rule¹.

TIGER PRO

Tiger Pro addresses the problem of teaching students about the consequences of writing poor requirements and the difficulty of writing good requirements.

This tool is used in classes on systems and software engineering. Poor

¹ Learning to use the C and D scales on a sliderule to multiply and divide was quick; the other aspects to a little longer. Thus learning to use the main feature of these tools should also be quick, while taking additional time to learn to use the other features of a tool could also take longer.

requirements have been cited as a major cause of cost and schedule overruns in the acquisition of Defence and information systems for at least 15 years (Hooks 1993, Kasser and Schermerhorn 1994, Jacobs 1999, Carson 2001, Lindquist 2005). Yet in all this time, academia has not been able to alleviate the problem through traditional teaching techniques. In an attempt to alleviate the problem through education a postgraduate Requirements Workshop for the writing of good requirements was first developed at UMUC in 1998 and later incorporated into classes at UniSA.

The workshop entailed students reviewing and commenting on a sample requirements document. The workshop was limited by time so that only a few pages of the 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 tallied their findings of good and bad requirements, there was little if any consensus on the quality of the requirements, sometimes there were widely differing opinions on the quality of the same requirement. Realizing that the parsing process which was taking up much of the student's time in the workshop could be automated to allow more time to discuss the content of the requirement, a prototype software tool was conceptualised and then developed to perform the function (Kasser 2002). The tool parsed the text of the requirement from the testing perspective looking for the presence of 'poor words' that complicated testing that the system's performance conformed to the requirement. These poor words thus indicated that a defect or a potential for a

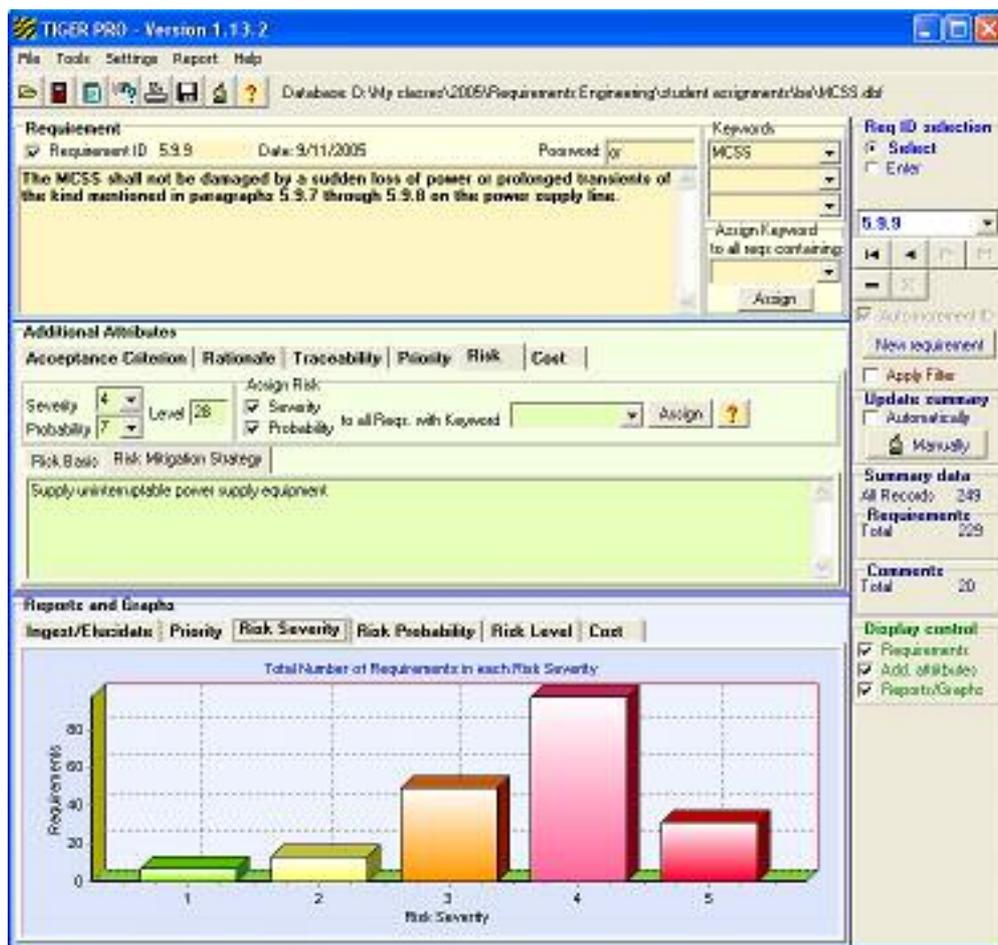


Figure 1 TIGER Pro – Typical Main Screen

defect existed in the requirement. The initial list of poor words was gathered from experience and the literature including (Hooks 1993, Kar and Bailey 1996, Kasser and Schermerhorn 1994).

A typical main screen view of TIGER Pro (Kasser 2004) is shown in Figure 1. Before using TIGER in a class tutorial on requirements engineering in three postgraduate courses (subjects) the discussions in the tutorials had focussed on the structure and format of requirements. After TIGER was introduced and used to elucidate sample requirements, the focus of the in-class discussions changed to cover the difficulties of writing good requirements (Kasser, Tran and Matison 2003). As TIGER had evolved into TIGER Pro to contain acceptance criteria to help clarify the meaning of the requirements by containing the answer to the question 'how will I know when the requirement is met?', at least one student in two consecutive courses asked to be taught how to write good acceptance criteria. Thus once again a significant change in student perspective was achieved.

CRIP CHARTS

This tool is used in classes on systems and software engineering to address the problem of measuring project progress. Measuring progress in the creation of a system or the development of software is difficult. Thus project managers cannot answer a simple question posed by a customer – i.e. 'what do you mean you can't tell me how much of my project has been completed?' (Kasser 1997) with any accuracy until the project has been completed. CRIP Charts (Kasser 1999) were developed as part of a research program into the properties of object-oriented requirements to provide some answers to these types of questions. CRIP charts use a technique similar to Feature Driven Development (FDD) (Palmer and Felsing 2002) to monitor the state of a feature or requirement during the System Development Life Cycle (SDLC). FDD

Charts however, show the state of every requirement or feature, namely they are suitable for detailed discussion by developers, but provide information overload for managers and do not cover the entire SDLC. CRIP Charts on the other hand cover the entire SDLC and provide summaries suitable for management but have to be integrated into the process. The four step CRIP approach is:

1. Categorize the requirements.
2. Quantify each category into ranges.
3. Place each requirement into a range.
4. Monitor the differences in the state of each of the requirements at the SDLC reporting milestones

The first part of the approach avoids the problem of comparing requirements of different complexities. The last step is the key element in the CRIP approach.

1. Categorise the requirements.

Categories are identified for the requirements. Typical categories are priority of the requirement, the estimated difficulty of implementing the requirement, estimated cost to implement the requirement, etc.

2. Quantify each category into ranges.

Each category is then split into no more than ten ranges. The ranges are relative, not absolute. A requirement may be moved into a different range as more is learned about its effect on the development during the implementation phase. However, the rules for setting the range limits must not change during the life of the project. Thus, the priority of a specific requirement or the cost to implement may change between reporting milestones.

3. Place each requirement into a range.

Each requirement is then placed into one range for each category. If all the requirements end up in the same range, such as all of them have the highest priority, the range limits are re-examined to spread the requirements across the full range.

4. Monitoring the states of implementation. Each requirement can be in one of five states at any time during the SDLC. These states of implementation of each requirement during the project are “identified”, “in process”, “completed”, “in test”, and “accepted”. During the process of building the system, requirements move through these states from “identified” to “accepted”. The summaries of the number of requirements in each state are reported at project milestones.

USING THE CRIP CHART

Each cell in the CRIP Chart contains three numbers,

- expected (E) from last reporting period,
- actual (A) achieved, and
- planned (P) for next reporting period.

For the first milestone reporting period, the values for ‘expected’ are derived from the project plan for the time period. The ‘actual’ value is the number

measured during the reporting period, and the ‘planned for next reporting period’ is a number derived from the project plan and the work done during the current reporting period. From then on the planned numbers are based on the state of the project. Numbers move horizontally across the CRIP Chart over time. As work progresses, the numbers flow across the columns from ‘identified’ to ‘accepted’ At each reporting milestone, the *changes* in the *state* of each of the requirements *between* the reporting milestones are presented as can be seen in the typical CRIP Chart shown in Figure 2.

Colours can be used to draw attention to the state of a cell in the table. For example the colours in this figure are allocated such that:

- **Violet** – shows requirements implementation is well ahead of estimate.
- **Blue** – shows requirements implementation is ahead of estimate.
- **Green** – shows requirements

	<u>Identified</u>			<u>In Process</u>			<u>Completed</u>			<u>In Test</u>			<u>Accepted</u>		
	E	A	P	E	A	P	E	A	P	E	A	P	E	A	P
1	5	5	1	4	10	0	3	3	1	2	2	1	1	1	0
2	5	9	1	4	10	0	3	3	0	2	2	1	1	1	1
3	5	13	2	4	9	1	3	7	1	2	4	1	1	1	0
4	4	7	0	4	9	1	3	7	0	2	2	0	1	1	1
5	5	5	1	4	8	1	3	5	0	2	2	1	1	1	1
6	5	11	0	4	9	1	3	5	1	2	4	1	1	1	0
7	5	5	1	4	9	0	3	3	1	2	4	1	1	1	0
8	5	11	1	4	8	0	3	5	1	2	4	0	1	1	0
9	4	7	1	4	10	0	3	7	0	2	2	1	1	1	1
10	5	5	1	4	9	1	3	5	0	2	2	1	1	1	0

Figure 2 CRIP Chart for Category X

implementation is close to estimated values.

- **Yellow** – shows requirements implementation is slightly below estimate.
- **Red** – shows requirements implementation is well under the estimated value

One chart can show that a problem might exist. Any time there is a deviation from expected to actual, the situation needs to be investigated. A comparison of the summaries from different reporting milestones can identify progress and show that problems may exist. On its own however, it cannot identify the actual problem. For example in Figure 2:

- The cell in the “Identified” column for Range 6 shows that the project planned that five requirements would be identified, but eleven requirements were actually identified and the project expects to identify none in the next reporting phase. Something may be wrong here and the situation needs to be investigated.
- In Range 3, it was expected that the part of the system implementing two requirements would go into test in the last reporting period, yet four made it into test and one is are planned for the next reporting period. Testing seems to be ahead of schedule.

The CRIP charts when viewed over several reporting periods can identify other types of “situations”. The CRIP chart may be used on a standalone basis or in accordance with budget and schedule information. For example, if there is a change in the number of:

- **Identified requirements** and there is no change in the budget or schedule, there is going to be a problem. Thus, if the number of requirements goes up and the budget does not, the risk of failure increases because more work will have to be done without a change in the allocation of funds. If the

number of requirements goes down, and the budget does not, there is a financial problem.

- **Requirements being worked on**, and there is no change in the number being tested, there is a potential supplier management or technical problem if this situation is at a major milestone review.
- **Requirements being tested**, and there is no change in the number accepted, there may be a problem with the supplier’s process.
- **Identified requirements at each reporting milestone**, the project is suffering from requirements creep if the number is increasing. This situation may reflect controlled changes due to the change in the customer’s need, or uncontrolled changes.

The CRIP chart may be used on a standalone basis or together with budget and schedule information. The CRIP Chart approach to measuring progress can provide a more accurate answer to the buyer’s question than any other measurement approach in use today. When used in student projects (Kasser, et al. 2002), CRIP charts provided a high degree of visibility of the status of a project and indicated potential problems early in the semester (project development life cycle). A software tool for demonstrating project states via CRIP charts has also been developed to assist in teaching about the concept (Kasser, Tran and Matisons 2003). One UMUC student used the CRIP chart in her workplace in conjunction with traditional reporting methods and management’s response was positive.

PAM CHARTS

In management classes, the traditional approach used by students planning a project is to use a Work Breakdown Structure (WBS) to allocate work by listing the activities in the hierarchical format of the WBS. Even after the students have been told that a WBS is a hierarchical

representation of the work being done, namely of a process they still generally start planning work by drawing a WBS. This approach often allows some work elements (processes) to be overlooked.

The Product-Activity-Milestone (PAM) chart (Kasser 1995) shown in Figure 3 uses a product-based approach to develop the activities in the WBS. This simple paper and pencil based tool has been found to be very useful in developing the relationships between the product, the activities and the milestone. The PAM chart is similar to, a cause-and-effect chart and is a tool used in the initial stages of project planning to facilitate:

- Defining milestones.
- Defining the product(s) or goals to be achieved by the milestone
- Determining the activities to produce the product(s). Defining the resources needed to produce the product(s).

The PAM chart consists of the following four parts:

- **The milestone** - Shown as a circle.

- **The product(s) produced** - Drawn as sloping lines leading towards the milestone.
- **The activities** - Drawn as horizontal lines leading to the product line. They are labelled above the line.
- **The resources associated with each activity** - Shown as labels below the activity lines.

The PAM Chart is implemented using paper and pencil. Starting with a blank page, a milestone is positioned at the end of the paper. Arrows are then drawn on the product and activity lines to show the direction of progress. Note there may be more than one milestone within the chart. This is because the simple PAM chart does not explicitly show any activities and resources needed to integrate the products for the milestone.

Although the labels have used letters in this example, in practice one would use WBS style numerical listings. Thus for example, Product 3 would have WBS elements 3.1, 3.2 etc. Once the PAM chart for the products to be produced for a

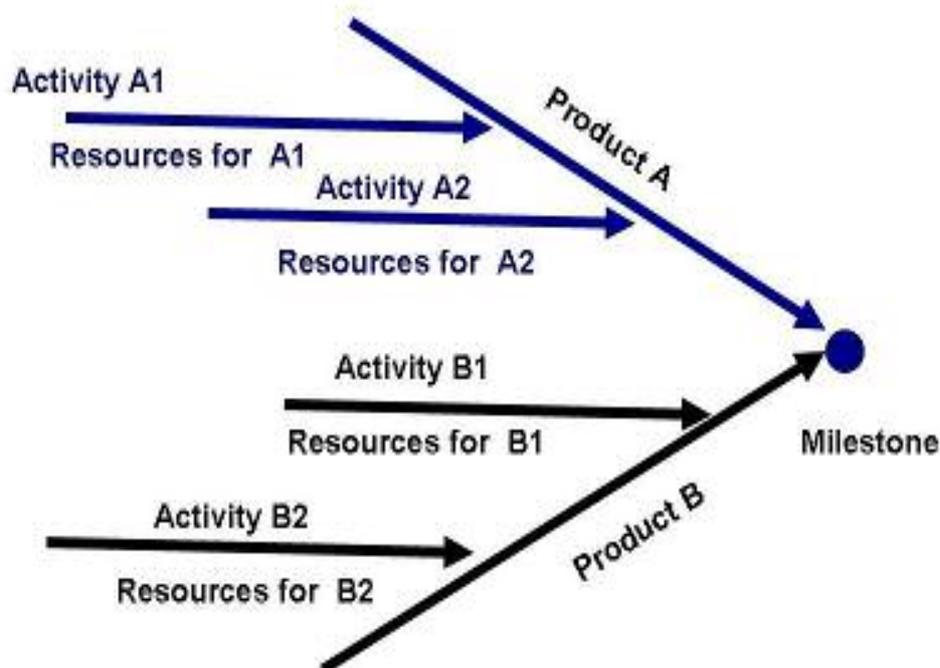


Figure 3 The PAM Chart - (Kasser 1995)

specific milestone has been developed, students use a PAM chart to assign:

- **Work elements** to the task by linking them to complying with requirements. In this way, the cost of the work performed to comply with a task requirement will be recorded and can be used as a baseline to refine future cost estimates.
- **Cost accounts to the WBS** elements on the basis of the products to be produced by the task. This approach will allow the manager to know exactly how much the work on a product produced by a task actually cost.

The data are then entered into a computer project management program for further processing such as resource levelling. This approach ensures that the WBS accurately maps onto all the work in the project providing a 100% mapping of the WBS onto the Project Breakdown Structure (PBS), namely there is never a difference between the WBS and the PBS.

Students generally tend to resist using the PAM chart approach to planning work until the differences in coverage between the PBS and WBS in their work assignments are pointed out. Even when they start using PAM Charts, it takes them a while to completely change from activity-based working to product-based working.

AUDIO ENHANCEMENT TO POWERPOINT PRESENTATIONS

Over the years, the author has experimented with adding audio to PowerPoint presentations in several ways. A low-cost audio editing and enhancing tool available from (<http://www.goldwave.com>) was employed from the start. This provided true word-processing capabilities. The approaches that were tried together with comments on those approaches are:

- **Embedding the audio in the**

PowerPoint file – produces large files which cannot be downloaded over telephone lines in a reasonable time (less than an hour). In addition, the audio cannot be edited, so any change to the presentation meant rerecording the audio. This approach was examined but not used (Kasser 1998).

- **Using separate audio files for each PowerPoint slide.** This approach was used in 1998-2003 and provided the maximum amount of flexibility in creating and editing the audio, but produces a set of large WAV files. This approach was used when producing CD-ROMs but not in the online environment.
- **Creating a Realmedia™ file** from PowerPoint and the separate audio files. This worked very well producing reasonable sized files in 1998-1999 but then the manufacturer discontinued the capability in its upgraded product. However, while it was in use, some students recorded the audio on to cassettes for replay while commuting.
- **Creating an MP3 audio file.** This is the 2004-2006 approach. It requires that the PowerPoint presentation be accompanied by an audio file and verbally synchronized. Synchronisation is achieved verbally by using words such as ‘and on the next slide’ or ‘as slide 27 shows’. These cues are almost unnoticed in the context of the lecture/presentation. The MP3 files were created either by combining the existing separate audio files using an audio editing tool or by dictating the lecture into a Digital Voice Recorder (DVR). This approach is very flexible and lectures have been prepared in various locations including, the home, the office, and airport departure lounges².

² There was considerable interference from the airport public address system that had to be deleted

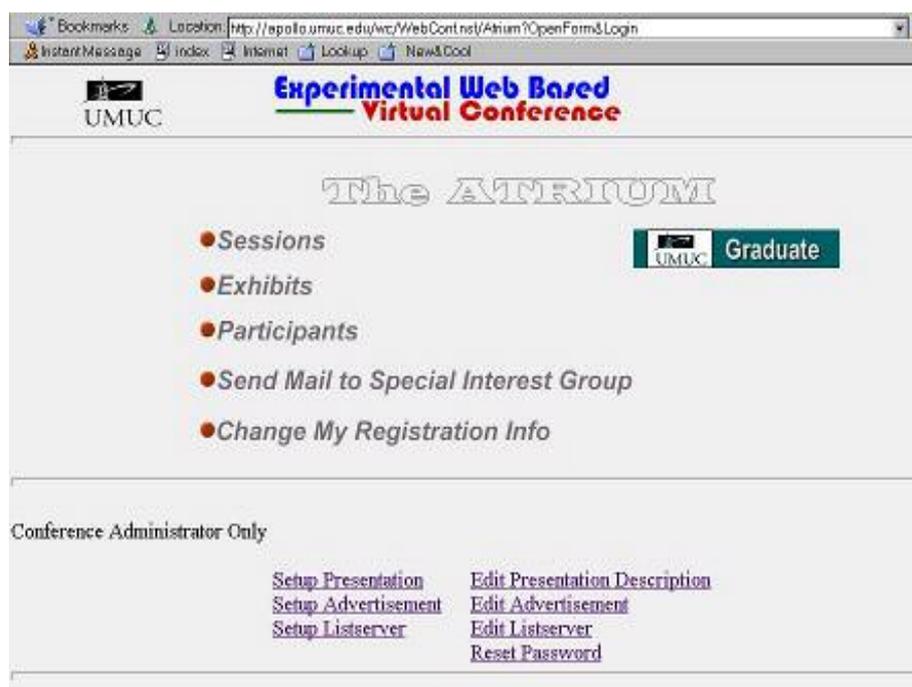


Figure 4 One Version of the WebForum Portal

THE WEBFORUM

While it was easy for the instructor to post PowerPoint lectures and the accompanying audio on the class web site, it was rather more difficult for the students to do the same. Students in two different years in the taught Master of Software Engineering degree at UMUC were posed the problem of developing a web page which provided that functionality as their final project. The project took the form of one pass through the SDLC over the course of the 14 week semester. The Webforum project ran in two different classes in successive years and was implemented with two different designs and architecture, both of which provided the needed functionality. The Webforum (Kasser 2000) is a web tool which can be used in face to face as well as online classes.

The Webforum, portal of which are shown in Figure 4 and Figure 5 was developed both as an Asynchronous Learning Network (ALN) classroom and as an online conference since there is little

from the audio file before posting in the asynchronous classroom.

difference between a conference session and a classroom other than the length of time the presenter talks, and the nature of the audience. While the WebForum provides a place to post presentations and audio, it also allows participants to discuss and rate the presentations.

The Webforum can also be used in the traditional classroom. In several classes (traditional and asynchronous) at UMUC, students were required to make a presentation about their term paper at the end of the semester³ with the following results.

- When students in a face-to-face class were offered the opportunity to create and deliver a distance mode (asynchronous) presentation in the classroom, one student commented positively that it took the stage fright out of the delivery process.

³ This provided them with both an opportunity to make a presentation and get feedback on the content of their paper from the instructor and their peers before submitting the assignment.



Figure 5 An Alternate WebForum Portal (second time around)

- When asked about the presentations, the students commented that the only perceived difference between the live and recorded presentations were that they could not interrupt the recorded presentation.

SUMMARY

This paper being page-limited has briefly discussed a few of the tools developed over the last few years to improve engineering education at UMUC and UniSA. The benefits of using the tools include:

- **TIGER Pro** – improved the teaching of writing good requirements.
- **CRIP charts** - provided information about the state of projects.
- **PAM charts** - helped students plan the initial stages of projects.
- **Audio enhancement to PowerPoint presentations** - provided a classroom experience in the ALN.
- **The WebForum** – provided an ALN conference site suitable for conferences as well as classrooms.

Further information about the tools can be found in the references and on the

author's web site.

CONCLUSION

Using these tools has improved the postgraduate education experience and the results are well worth the investment in building the tools.

RECOMMENDATIONS

Make learning fun, use games and tools in the classroom. If you can imagine a tool but can't develop it, the computer science and software engineering departments are always looking for new projects.

BIOGRAPHY

Joseph Kasser has been a practising systems engineer for 35 years. He is the author of "*Applying Total Quality Management to Systems Engineering*" and many INCOSE symposia papers. He holds a Doctor of Science in Engineering Management from The George Washington University, and is a Certified Manager. He is the DSTO Associate Research Professor at the Systems Engineering and Evaluation Centre at UniSA. He performs research into the nature of systems engineering and the

properties of object-oriented requirements. He is a recipient of NASA's Manned Space Flight Awareness Award (Silver Snoopy) for quality and technical excellence for performing and directing systems engineering.

He is a Fellow of the INCOSE. He also serves as Region VI (Asia-Pacific) representative to the INCOSE Member Board and as President of INCOSE-Australia.

REFERENCES

- Carson, R. S., "Keeping the Focus During Requirements Analysis", the 11th International Symposium of the International Council on Systems Engineering, Melbourne, Australia, 2001.
- 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, Limerick, Ireland, 1999.
- Kar, P. and Bailey, M., "Characteristics of good requirements", 6th International Symposium of the International Council on Systems Engineering (INCOSE), Boston, MA, 1996.
- Kasser, J. E., *Applying Total Quality Management to Systems Engineering*, Artech House, Boston, 1995.
- Kasser, J. E., "What Do You Mean, You Can't Tell Me How Much of My Project Has Been Completed?" INCOSE 7th International Symposium, Los Angeles, CA, 1997.
- Kasser, J. E., "New Initiatives in Tycho via the Web Initiative in Teaching", University System of Maryland Web Initiative in Teaching Summer Design Inquiry Workshop, College Park, MD, 1998.
- Kasser, J. E., "Using Organizational Engineering to Build Defect Free Systems, On Schedule and Within Budget", PICMET, Portland OR., 1999.
- Kasser, J. E., "The WebForum", AusWeb2K-The Sixth Australian World Wide Web Conference, Cairns, 2000.
- Kasser, J. E., "A Prototype Tool for Improving the Wording of Requirements", Proceedings of the 12th International Symposium of the INCOSE, Las Vegas, NV, 2002.
- Kasser, J. E., *The First Requirements Elucidator Demonstration (FRED) Tool*, Systems Engineering: The Journal of the International Council on Systems Engineering, Vol 7, no. 3, 2004.
- Kasser, J. E., Cook, S. C., Pilgrim, A., Gamaliel, Y. and Dat, B. T. T., "The CREAP Project: A Case Study of a System Engineering Educational Project", The 12th International Symposium of the INCOSE, Las Vegas, NV, 2002.
- Kasser, J. E. and Schermerhorn, R., "Determining Metrics for Systems Engineering", The 4th Annual International Symposium of the NCOSE, San Jose, CA, 1994.
- Kasser, J. E., Tran, X.-L. and Matisons, S., "Prototype Educational Tools for Systems and Software (PETS) Engineering", Proceedings of the AAEE Conference, 2003.
- Lindquist, C., *Fixing the Requirements Mess*, CIO Magazine, 2005.
- Palmer, S. R. and Felsing, J. M., *A Practical Guide to Feature - Driven Development*, Prentice Hall, 2002.