Categorising attributes of requirements to lower probability of cost and schedule overruns

Xuan-Linh Tran, Joseph Kasser

Systems Engineering and Evaluation Centre (SEEC)

University of South Australia

Mawson Lakes Campus

South Australia, SA 5095

Abstract. Poor requirements engineering and management have been cited as a major cause of cost and schedule overruns for at least the last 15 years. However, attempts to address this problem by requirements engineering and management professionals using traditional methods have not been greatly successful.

This paper proposes a research programme focusing on developing information and ways of displaying that information to provide decision makers with the opportunity to make decisions that cannot be performed in the current paradigm to minimise cost and schedule overruns. It does this by introducing an object-oriented model that integrates the requirements engineering and requirements management processes in the system development life cycle (SDLC), where the requirement defines the product, and risk, cost and priority are attributes of the requirement.

The paper first describes the concept, lists the candidate requirement attributes and discusses the integrated model. Then it discusses the early implementation of the model in an educational and research software tool. Finally, the anticipated drawbacks, benefits and plans for future research based on the findings are presented.

INTRODUCTION

Poor requirements engineering and management has been cited as a major cause of cost and schedule overruns in projects failure in the last 15 years(Capers_Jones 1996), (CHAOS 1995) and (OASIG 1996). Over the years, requirements engineering and management have improved through the use of modelling and simulation, and the development of text handling requirements tools. However all these efforts could not reduce cost and schedule overruns by more than about 20% (CHAOS 2004).

In the traditional modelling and simulation process, the focus of the model is on the functional and performance requirements for the product. This paper presents a new method of looking at both the functional and performance requirements and associated attributes (risk, cost and priority) in the modelling and simulation process, proposing a model that integrates the requirements engineering and the requirements management processes in the system development life cycle (SDLC).

The paper first describes the concept, lists the candidate requirement attributes and proposes the integrating model. Then it discusses the early implementation of the model in educational and research software tools. Finally, benefits, drawbacks and plans for future research based on the findings are presented.

THE TRADITIONAL SYSTEMS ENGINEERING APPROACH

The SDLC has evolved several methodologies since the early days of the Waterfall model. One of them, the Spiral model (Boehm 1988) placed explicit emphasis on Risk Management. However, even with Risk Management and the current emphasis on Process Standards and Capability Maturity Measurement, the developer working within the current production paradigm, cannot answer two simple questions posed by the customer during the SDLC, namely:

- *"What Do You Mean You Can't Tell Me How Much of My Project Has Been Completed?"* (Kasser 1997).
- *"What Do You Mean You Can't Tell Me if My Project is in Trouble?"* (Kasser & Williams 1998)

There has been a lot of research into building the right system and doing requirements better (Glass 1992). Much of that research has focused on (1) how to state the requirements in the form of a specification once they have been obtained, (2) using a requirements traceability matrix (RTM), and the tools that incorporate a RTM. However, the research into improving requirements at SEEC has a different focus. It is based on the recognition that requirements engineering and management is evolving as evidenced by the following changes in the definition of the term.

- "*the science and discipline concerned with analysing and documenting requirements*" (Dorfman & Thayer 1990).
- *the systematic process of eliciting, understanding, analysing, documenting (and managing) requirements*" (Kotonya & Summerville 2000). This definition includes the elicitation process as well as the management process

The research is also based on the recognition that a requirement is more than just the imperative statement (Kasser 1995). For example, both (Alexander & Stevens 2002) and (Hull, Jackson & Dick 2002) discuss additional attributes of the text-based requirement (e.g. priority and traceability) in conjunction with improving the writing of requirements. However, in practice, there is difficulty in adding these additional attributes to the traditional requirements document or database and then managing them. This is because the current systems and software development paradigm generally divides the work in a project into three independent streams – Management, Development, and Test (Quality) (Kasser 1995). Thus requirements engineering tools contain information related to the Development and Test streams (the requirements for the product) while the additional properties tend to be separated in several different tools, (e.g. Requirements Management, Project Management, Work Breakdown Structures (WBS), Configuration Control, and Cost Estimation, etc.). Moreover, activities that have become increasingly identified as being critical to success (e.g. risk planning and management) in many cases are not performed in the current paradigm. In those cases where they are performed, they tend to be treated as add-on's, and implemented in a complicated process that could have been designed and pictured by W. Heath Robinson (UK) or Rube Goldberg (USA)[1].

THE OBJECT-ORIENTED PARADIGM

Recognising that requirements drive the work done on a project as shown in Figure 1 (Kasser 1995) an object-oriented research project was initiated at SEEC to develop tools to model and display information about the combined product and process properties of requirements. These tools were known as Prototype Educational Tools for Systems and Software Engineering (PETS) (Kasser & Cook 2003; Kasser, Tran & Matisons 2003). In particular the PETS:

- Identified candidate properties of a requirement object based on information in the process (development, management and test and development streams of work in the SDLC) as well as information about the product needed.
- Described some of the functionality that could be added to the requirements object.
- Indicated that object-oriented requirements engineering and management can effect a significant reduction of the effect of the problems currently encountered in the SDLC which cause the cost and schedule overruns.

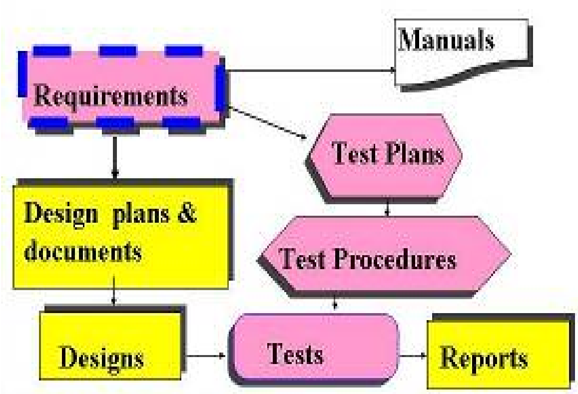


Figure 1:Requirements Drive the Work (Kasser 1995)

A set of candidate process and product attributes of requirements being necessary for effective system and software development was summarised in the form of a set of Quality Systems Elements (QSE) (Kasser 2000). The QSE are not new. They are known and have been used individually in project management and systems engineering for many years. For example, the United States Military Standard 2167A prescribed a set of software development folders (MIL-STD-2167A 1998) that shall include (directly or by reference) the following information:

- Design considerations and constraints.
- Design documentation and data.
- Schedule and status information.
- Test requirements and responsibilities.
- Test cases, procedures, and results.

Some of the QSE have also been incorporated as fields in requirements management tools from time to time. However, these instances seem to be the exception rather than the rule, moreover, even when used, the QSE do not seem to have been used together in an integrated manner. SEEC's object-oriented approach to research integrates them. The PETS project considers the QSE as an initial set of candidate properties of requirements and thus at least improve on the current paradigm by providing a place to store those additional properties in an integrated database containing information about the process and product.

Early research has shown that elements of the QSE can be considered individually and interdependently to model information about systems and software that cannot be provided in the current paradigm, and provide decision makers with information that should reduce cost and schedule overruns. The following candidate attributes are discussed below:

- Acceptance criteria.
- Priority.
- Risk.
- Cost estimates.

Acceptance criteria. Acceptance criteria answer the question "How will we know if the requirement has been met?" Acceptance Criteria are important properties that not only drive the testing stream of work, they also facilitate building the right system in the first place. If the requirements were clearly stated at the beginning, then work on building the wrong system would have been prevented.

Priority. Knowing the priority allows high priority items to be assigned to early Builds, and simplifies the analysis of the effect of budget cuts (Denzler & Vallone 1995) and (Kasser

[1] Cartoonists in the USA and UK who drew cartoons of complicated systems designed to perform simple functions.

1999). Priorities of the requirements can be summarised in the form of a Bar chart as shown in Figure 2. If the majority of requirements for a project have a low priority profile, the project is a candidate for cancellation during periods of tight budgetary constraints. On the other hand, if most of the requirements for another project have a high priority as shown in Figure 2 such a profile may need to be re-prioritised depending on the budget situation.

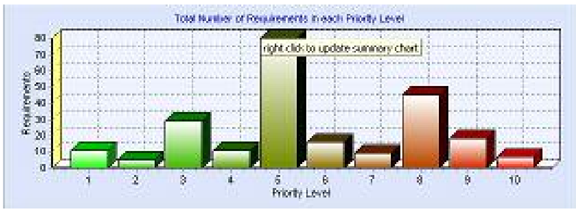


Figure 2 Priority Profile of a Project

Risk. Understanding the nature of project risks is important in determining where and when to apply resources in order to help mitigate those risks. With appropriate modelling, the risk property could determine an appropriate Risk Profile for the type of system to be realised. The risks could be assigned values between 1 and 10, and the resulting profile presented as a Bar chart to provide a visible risk profile as shown in Figure 3. By incorporating risk atributes of requirments in the object-oriented requirement, a Risk Mitigation plan would become an abstracted view of the QSE database.

Figure 3 Risk Profile of a Project

Cost estimates. Tasks are generally defined in a WBS from which schedules and milestones are identified. Each task is allocated a task cost account number. This construction of the WBS provides both the organisation and the customer with information for planning and improving future task estimates for the costs of products. The object-oriented requirement contains attributes for the estimated cost together with a variance percentage. The resulting cost profile presented in spreadsheet format is as shown in Figure 4.

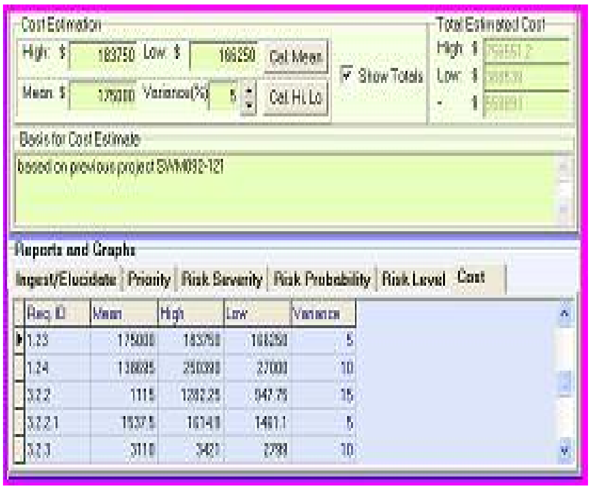


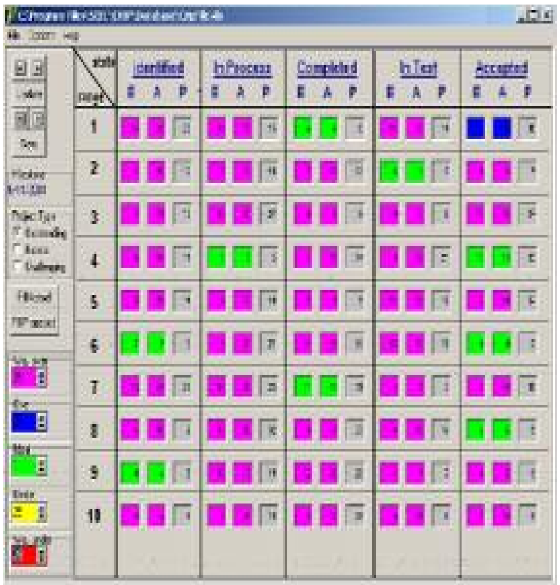
Figure 4 Cost Estimate Profile of a Project

CATEGORISED REQUIREMENTS IN PROCESS (CRIP) CHARTS

Categorised Requirements in Process (CRIP) Charts are a tool for measuring percentage completion of projects. Measuring progress during the creation of a system or the development of software is difficult. This is why project managers cannot answer the simple question posed by a customer such as *"What do you mean you can't tell me how much of my project has been competed?"* (Kasser 1997) with any accuracy until the project has been completed. Hence Categorized Requirements in Process Charts (CRIP) were developed to provide better answers to the question than is provided by current measurements of work done, funds spent, etc. (Kasser 1999). CRIP charts use a technique similar to Feature Driven Development (FDD) (Palmer & Felsing 2002) to monitor the state of a feature or requirement during the SDLC. FDD Charts however, show the state of every requirement or feature, namely suitable for detailed discussion by developers, but 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 steps of process using CRIP charts are:

1. Categorise the requirements. Categories are identified for the requirements. Typical categories are priority, and risk of the requirement, estimated cost to implement the requirement, and complexity level (the estimated difficulty of implementing the requirement).

2. Quantify each category into ranges. Each of these categories is then split into no more than



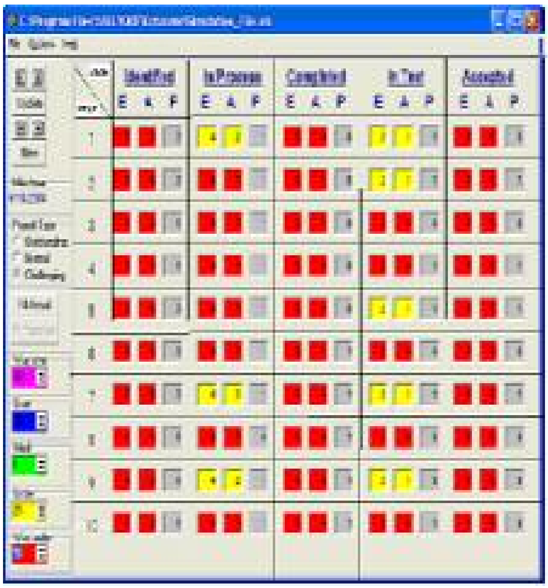


Figure 5 Typical CRIP Charts

10 relative (not absolute) ranges. A requirement may be moved into a different range as more is learned about its effect on the development during the implementation phase. Thus, the priority of a specific requirement or the cost to implement may change between reporting milestones. However, the rules for setting the range limits must not change during the life of the project.

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 having the highest priority; the range limits are re-examined to spread the requirements across the full range.

4. Monitor the states of implementation. The state of implementation of each requirement varies during the project as 'identified', 'in process', 'completed', 'in test', 'accepted'. During the process of building the system, requirements move from the identified stage to accepted stage.

Note that 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.

POPULATING AND USING THE CRIP CHART

Examples of CRIP charts are shown Figure 5. Each cell in the chart contains three numbers:

- expected (E) from last reporting period,
- actual (A) achieved
- 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 Charts shown Figure 5.

Colours are used to draw attention to the state of a cell in the table. For example, green signifies that the number of actual and expected requirements for that cell in the reporting milestone match, red signifies that actual is below expected, and blue signifies that actual is ahead of expected.

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 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.

DISPLAYING COMPLEX INFORMATION IN A SIMPLE AND USEABLE MANNER

Combinations of properties. While the work to date has allowed questions such as “is a high-risk high-cost requirement really needed?” or “is a high-cost low-priority” to be posed at the Systems Requirements Review (SRR), it has not been able to provide a way of displaying those properties to identify the requirements for which those questions should be posed. The next development stage in the PETS project will explore these features to develop ways of presenting the information to draw the attention of the users to these situations. The goal of the research is to develop a display at least as efective as the classic drawing by Charles Joseph Minard showing the fate of Napoleon’s army in its 1812 campaign in Russia. Minard’s drawing display plots six variables in a two dimentional framework (Tufte 1983).

A SIMPLE RANKING MODEL FOR COST, RISK AND PRIORITY

This research is concentrates on developing ways to present information about the properties of the requirements. In order to do this we propose a simple but realistic underlying data model to process the information being presented. This section discusses such a model.

Although risk and cost analysis information are widely used by project team members in the System Requirement Review process, it (Ionita et al. 2004; Fussell & Field 2005)can be a lengthy and complex process and still does not provide users an overall picture of the situation pertaining to the entire requirements document. Furthermore, with this approach, these attributes of requirements are split between the Management stream and the Development and Test streams of the project.

To introduce the concept of the model in a simplistic way, only the following three attributes of requirements are considered in the model at the initial stage of the research. They are the cost estimate attribute, the risk attribute and the priority attribute. In addition, the data used in the model will be simplified to single values only. This paper assumes users would be able to identify their own sets of values for the candidate attributes based on the relevance and experience to their industries. Our purpose is therefore not to elaborate further of how to identify attributes, but rather to formulate a mathematical model based on the values of these requirements attributes that users have defined. These value are then used as a basis for the development of ways to display views of the data in the model in a manner so aas to facilitate decision making and maximising the proabability of making the best decisions..

The model tabulates these candidate attributes in a 2-Dimentional matrix, converts attribute values from text to numeric format, then applies mathematical rules on these values to arrive at a number of values. In the prototype display, these values are then coloured according to their ranking schemes to provide users an instant overview of the situation of the requirements, aiding project managers and technical staff in the evaluation of projects.

The model is built based on the traffic light diagram concept where each cell contains values combined from the three candidate attributes. Values for each attribute in each cell are assigned based on the following three factors:

1. The distribution of ranges/levels and values,
2. The distribution of colour codes,
3. The determination of the attribute weighting ratio.

Factor 1 - The distribution of ranges/levels and values

The cost attribute: is the estimated costs to perform a task or to acquire an item. Cost is mainly estimated based on labour categories, hours, rates, materials, opportunities, etc. One may argue that allocation of cost estimates to every single requirement is not a practical task. It may be difficult in the current activity-based paradigm, but in an alternative future object-oriented product based paradim should be quite possible. With given user-defined low and high cost estimates for each requirement, the cost estimate can be categorised into several ranges. However, for the purpose of demonstration in this model, the cost attribute is represented by three ranges: low, average and high, and each of them is assigned a numerical value as follows:

- low cost estimate = 1
- average cost estimates = 2
- high cost estimate = 3

The risk attribute: the opposite of opportunity, it is generally referred as the degree of exposure to negative events and their probable consequences. Project risk level is characterized by three risk factors namely: risk event, risk probability and the amount at stake (Conrow 2003). For simple demonstration purpose, the model will initially work on three risk levels: low, medium, and high by assigning a numerical value to each of them as follows:

- low = 1,
- medium = 2,
- high = 3.

The priority attribute: is heavily dependent on the stakeholders. Different organisations categorise priority differently (Firesmith 2004). For instance, requirements are generally broken down into three broad categories: essential, desirable and optional. Since requirements with essential priorities are 'must-have' requirements (particularly in defence applications) their inclusion is a given. Hence, this model will focus only on requirements with desired or optional priorities. Again, for the research purpose and simple prototyping, the model converts these desired priority values into numeric values ranging from 1 to 3 where:

- high = 1,
- medium = 2,
- low = 3.

<u>Note</u> that the numerical values given to priorities are of opposite order from those given to cost and risk attributes. For example, the most desired requirement, or *best scenario*, would contain low cost (value = 1), low risk (value = 1) and high priority (value =1) and the least desired requirement, or *worst scenario* occurs when a requirement contains high cost (value = 3), high risk (value = 3) and low priority (value 3).

The numerical values representing these attributes are then calculated using a mathematical formula, which will be discussed below and are displayed in columns in a matrix

Factor 2 - The distribution of colour codes

Traffic light colours are used on each cell of the matrix to flag the likelihood of a requirement being met. It is up to the user to define the value for each colour based on their experience and industry protocol. If X is the call value, we can define the traffic lights as follows:

	Green – Accept. Likely to meet, where: $X \le a$
	Yellow – Watch. Likely problem area, where: $a < X \le b$
	Red – Mitigate. Problem area, where: $X > b$

Factor 3 - The determination of the attribute weighting ratio

This is the ratio distributed between the three attributes. This ratio is determined by users based on the project characteristic or industry protocol and preference. For example, some users would weight priority higher than cost and risk, while others weight cost more than the other two attributes. Recognising that the process of deriving the weighting ratio for these attribute can be a complicated. Therefore since the purpose of this research is focusing into displaying information and not on developing the information, the initial model will use the same formula to determine the combination to display based on the following two weighting ratios:

- **Ratio 1:** all attributes are identical:

  cost : risk : priority = 1:1:1

- **Ratio 2:** cost and risk are identical but priority is double:

  cost : risk : priority = 1:1:2

THE MATHEMATICAL FORMULA

The values of the requirement, cost, risk and priority attributes can be combined in several ways including simple summation, multiplication, or combination thereof. If all factors were to be multiplied, the results would be overly sensitive where small changes to the weighing factors would have dramatic outcomes. For instance, let us assume for a particular requirement, we have:

- requirement factor 0,
- cost factor 2,
- risk factor 1,
- priority factor 3

Multiplying these factors together would yield 6 (2 x 1 x 3). However, if we were to double each of these factors, then the result would be 48 (0 x 4 x 2 x 6). Such assessment is considered unrealistic in the real world. Even though the formula for the model is not that important because the focus of the research is on ways of displaying the combination of attributes, several formulae have been tried, tested and evaluated to determine a reasonable formula for this model. The best one seemed to result when adding instead of multiplying the values of the attributes. Therefore, the following formula was used for the intial display concept:

$$X = ReqW + (C \times Cw) + (R \times Rw) + (P \times Pw) \qquad (f1)$$

Where:

- X = cell value;
- ReqW = requirement weighting factor
- C = cost;
- R = risk;
- P = priority
- Cw = cost weighting factor
- Rw = risk weighting factor
- Pw = priority weighting factor

Nornally, the weighting factors of the requirement, cost, risk and priority are calculated based on the requirement state during the system development process and other factors of the requirement. This is a complex process and individually tailored to each industry and the nature of the specific project. For this model we assume the ReqW is minimal, so it can be rounded down to zero

APPLICATION

Stage 1. The model is in the form of a matrix; the first stage begins with placing attribute values into the first four columns where:

- Column 1: contains the unique identification numbers for the requirements concerned.
- Column 2: contains the cost estimate values of each requirement. For demonstration purpose, these values are sorted in numerical order from low (1) to high (3).
- Column 3: contains the risk level values of each requirement. For demonstration purpose, these values are sorted in numerical order from low (1) to high (3).
- Column 4: contains the priority values of each requirement. For demonstration purpose, these values are sorted in numerical order from high (1) to low (3).

Stage 2. Substituting the predetermined values of candidate attributes from the four columns described in stage 1 into variables in formula f1, the derived results for the two cases identified below are placed in column 5 (case 1), and column 6 (case 2) as shown in Figure 6.

Note that the ranges, weighting factor and colour coding distribution values are all determined by user. This allows users the capability to customise these values to fit the nature of individual projects and industry. Below are two sample cases this model depicts:

Case 1: contains the following factors:

- weighting scale ratio is 1:1:1
- X = cost + risk + priority
- colour coding distribution:

  $3 \leq X \leq 4$: (green)

  $4 < X < 7$: (yellow)

  $7 \leq X \leq 9$: (red)

Case 2: contains the following factors:

- weighting scale ratio is 1:1:2
- cell = (cost/2) + (risk/2) + priority
- colour coding distribution:
- $X \leq 3$: (green)
- $3 < X < 5$: (yellow)
- $X >= 5$: (red)

With these simple displays, the requirements engineer can discuss the requirements in displayed red and yellow with the customer before the SRR.. The revision to the requirement can be by mean of changing the requirement, documenting the risk mitigation or revising the acceptance criteria.

STAGE 1				STAGE 2	
REQ ID	COST	RISK	PRIORITY	Case 1	Case 2
				1:1:1	1:1:2
3.2.1	1	1	1	3	2.00
3.25.9	1	1	2	4	3.00
3.26.3	1	1	3	5	4.00
3.54	1	2	1	4	2.50
3.8.25	1	2	2	5	3.50
3.12.5	1	2	3	6	4.50
3.12.9	1	3	1	5	3.00
3.5.1	1	3	2	6	4.00
3.8.14	1	3	3	7	5.00
3.7.22	2	1	1	4	2.50
3.13.4	2	1	2	5	3.50
3.9.2	2	1	3	6	4.50
3.11.26	2	2	1	5	3.00

3.12.5	2	2	2	6	4.00
3.22	2	2	3	7	5.00
3.21.45	2	3	1	6	3.50
3.11.56	2	3	2	7	4.50
3.2.65	2	3	3	8	5.50
3.8.56	3	1	1	5	3.00
3.8.25	3	1	2	6	4.00
3.9.69	3	1	3	7	5.00
3.4.34	3	2	1	6	3.50
3.7.28	3	2	2	7	4.50
3.1.51	3	2	3	8	5.50
3.2.33	3	3	1	7	4.00
3.45.2	3	3	2	8	5.00
3.22.8	3	3	3	9	6.00

Figure 6 Cost-Risk-Priority Possible Outcome Matrix

A graphical display of these possible outcomes for a set of requirements with a particular priority level can be produced as shown in Figure 7. This provides users a general overview of the requirements situation in a similar manner to FDD displays.

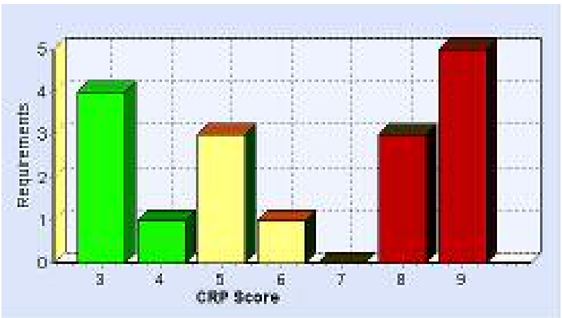


Figure 7 Cost-Risk-Priority Profile for requirements with high priority

BENEFITS

The benefits of considering the cost, risk and priority attributes of requirements as well as the product specification attributes are:

- Requirements attributes become more associated with the requirements engineering and management process.
- Requirements and their associated attributes are more than just textual statements and can also be visualised in other formats such as graphics or formulae.
- Can be used in a tool such as Quality Function Deployement (Hauser & Clausing 1988) for sensitivity analysis.
- provides the ability to customise ranges, individual attribute weighting factors, weighting ratios, colour-coding distribution, and choices of display. For example, users can customise the model to display red and/or yellow cells only.
- Ease of implementation. For example, users can customise the number of attributes or the type of attributes to be considered in the model.
- Ease of design and programming.
- Ability to links all work back to the original requirements.
- Provide early identification of problems in the implementation process via CRIP Charts.
- Questions such as “What would a display of risk vs. cost vs. priority look like to be useable?” are answered.
- Provides new perspectives on the system. For example, various properties can be examined at the SRR (before the system is built and the cost to effect changes is relatively low) and questions posed that are not easily identified or answered in the current paradigm. Typical examples are:
  - Is a high (estimated) cost to implement, low priority requirement really needed?
  - Have the high priority requirements been assigned to early Builds? This is desirable, so that future budget cuts would tend to eliminate the lower priority performance characteristics.

DRAWBACKS

While the above model provides some simplistic pictorial ranking of requirements, interpretation issues still remain.

- The traffic light diagram model such as the one described above does not provide a solution to the problem of displaying information about these combined properties in the case of a large requirement document where there would be hundreds of red and yellow cells appearing in the matrix such as are seen in FDD. This is impractical since what is really needed from the management perspective are summaries such as those provided by CRIP Charts.
- Some users may use the model blindly, treating it as ‘holy grail’ indication of the state of the requirement, forgetting that

human decision still has the final say in any decision-making situation.

The resulting values can be manipulated. Hence, a strict ranking scheme must be established to clearly provide a standard for correct scaling. This is not always easy in complex systems. Therefore, further research is needed to find a better solution to the problem.

CONCLUSION

This paper proposed a research programme focusing on developing information and ways of displaying that information of combined attributes of requirements to provide decision makers with the opportunity to revise problematic requirements before the SRR. If properly implemented, the concept can help to reduce cost and schedule overruns on projects. However, the traffic light approach studied in this paper suffers from major drawbacks in this situation, hence requires further research. One of the immediate suggested improvements on the model is the more accurate ranking of the weighting factors (Buede 2000) for the requirement and each of its attributes.

Another issue that should be looked into is a better way to display these information rather than the traffic light approach, a display that is at least as efective as the classic drawing by Charles Joseph Minard of Napoleon's army in its 1812 campaign in Russia (Tufte 1983).

SUMMARY

This paper proposed a research programme focusing on presenting decision makers with the opportunity to make decisions based on simple visual display. It did this by introducing a database model that graphically and mathematically integrates the requirements engineering and requirements management processes in the SDLC, where the requirement defines the product, and its risk, cost and priority are attributes of the requirement.

The model demonstrates a simple traffic light concept for displaying the combined attributes of requirements. However further study is recommended.

ACKNOWLEDGMENTS

The authors would like to thank Mr William Scott for his invaluable contribution and support.

BIOGRAPHIES

Xuan-Linh Tran holds a Bachelor of Mathematical Science, majoring in Applied Mathematics and Computer Science from the University of Adelaide, and a Master of Information Technology from the University of South Australia. After starting a career as a software engineer, at Metro Meat Ltd in Adelaide and also at TP Data AB in Malmo, Sweden, Xuan-Linh joined Systems Engineering and Evaluation Centre (SEEC) at the University of South Australia in 2003 as Associate Researcher.

Her work includes research and developing educational tools to support some parts of the teaching and research programs in requirements engineering and management, and software testing and evaluation at SEEC.

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 and SETE conference 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

Alexander, IF & Stevens, S 2002, *Writing Better Requirements*, Addison-Wesley.

Boehm, B, 1988, 'A Spiral Model of Software Development and Enhancement', *IEEE Computer.*

Buede, DM 2000, *The Engineering Design of Systems*, John Wiley & Sons, Inc.

Capers_Jones 1996, *Patterns of Software Systems Failure and Success*, International Thomson Computer Press.

CHAOS 1995, *Chaos*, The Standish Group, viewed March 19, 1998, <http://www.standishgroup.com/chaos.html>.

CHAOS 2004, *Chaos Chronicles*, The Standish Group.

Conrow, EH 2003, *Effective Risk Management: Some Keys to Success* 2nd. edn, American Institute of Aeronautics and Astronautics, Reston VA, USA.

Denzler, DWR & Vallone, A 1995, 'Operations: The Missing Third of Systems Engineering'. *5th Annual International Symposium of the National Council of Systems Engineering (NCOSE)*, St. Louis, MO.

Dorfman, M & Thayer, RH 1990, *System and Software Requirements Engineering*, IEEE Computer Society Press.

Firesmith, D 2004, 'Prioritizing Requirements', *Journal of Object Technology,* vol. vol. 3, no. no. 8, September-October, pp. 35-47.

Fussell, L & Field, S 2005, 'The role of the risk management database in the risk management process', *Systems Engineering, 2005. ICSEng 2005. 18th International Conference on 16-18 Aug. 2005. Pages: 364 – 369.*

Glass, RL 1992, *Building Quality Software*, Prentice Hall, Englewood Cliffs, NJ.

Hauser, JR & Clausing, D 1988, 'The House of Quality', *Harvard Business Review,* vol. May-June, pp. 63-65.

Hull, MEC, Jackson, K & Dick, AJJ 2002, *Requirements Engineering*, Springer,

Ionita, MT, America, P, Hammer, DK, Obbink, H & Trienekens, JJM 2004, 'A scenario-driven approach for value, risk, and cost analysis in system architecting for innovation', *Software Architecture, 2004. WICSA 2004. Proceedings. Fourth Working IEEE/IFIP Conference on*
12-15 June 2004 Page(s):277 - 280 .

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

Kasser, JE 1999, 'Using Organizational Engineering to Build Defect Free Systems, On Schedule and Within Budget', *The PICMET*, Portland OR.

Kasser, JE 2000, 'The WebConference: A Case Study', *The INCOSE - Mid-Atlantic Regional Conference*, Reston, VA.

Kasser, JE 1997, 'What Do You Mean, You Can't Tell Me How Much of My Project Has Been Completed?' *The INCOSE 7th International Symposium*, Los Angeles, CA.

Kasser, JE & Cook, SC 2003, 'Using a Rapid Incremental Solution Construction Approach to Maximise the Completeness and Correctness of a Set of Requirements for a System', *Proceedings of the 13th International Symposium of the International Council on Systems Engineering (INCOSE)*, Washington DC.

Kasser, JE, Cook, SC, Pilgrim, A, Gamaliel, Y & Dat, BTT 2002, 'The CREAP Project: A Case Study of a System Engineering Educational Project', *The 12th International Symposium of the INCOSE*, Las Vegas, NV.

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

Kasser, JE & Williams, VR 1998, 'What Do You Mean You Can't Tell Me If My Project Is in Trouble?', *The First European Conference on Software Metrics (FESMA 98),* Antwerp, Belgium.

Kotonya, G & Summerville, I 2000, *Requirements Engineering processes and techniques*, John Wiley & Sons, Chichester.

MIL-STD-2167A 1998, *Defense System Software.*

OASIG 1996, *The performance of information technology and the role of human and organizational factors. Report to the Economic and Social Research Council, UK*, viewed January 16, 2002, <http://www.shef.uc.uk/~iwp/publications/reports/itperf.html>.

Palmer, SR & Felsing, JM 2002, *A Practical Guide to Feature - Driven Development*, Prentice Hall,

Tufte, ER 1983, *The visual display of quantitative information*, Graphics Press,