# Determining Metrics for Systems Engineering

By:
Joe Kasser
POB 3419
Silver Spring, MD 20918
Phone - (301) 593-6136

and

Robin Schermerhorn
Computer Sciences Corporation
4600 Powder Mill Rd, Beltsville, MD 20705
Work phone - (301) 572-8634

**Abstract** Systems Engineering Metrics are not easy to establish, perhaps that is why ineffective Systems Engineering is so prevalent. This paper discusses suggested metrics to measure the degree of effectiveness of both the Systems Engineering product and process. Metrics are suggested for reviews and documentation as examples of measuring the quality of the products, and for the effectiveness of Systems Engineers as examples of measuring the quality of the process.

## THE SYSTEMS ENGINEERING PROCESS

Systems Engineering has been recognized as the *process* by which the orderly evolution of man-made systems can be achieved (Blanchard et al, 1981). As a process, it can be defined, measured and improved. One definition of Systems Engineering is (Eisner 1988):

*"An iterative process of top down synthesis, development and operation of a real world system that satisfies in a near optimal manner, the full range of requirements for the system."*

This process takes place over a period of time while the project producing the system passes through the concept definition, requirements analysis, design, implementation, test, delivery and operational support phases. Systems Engineers have a role to play in each phase of the Systems Development Life Cycle (SDLC) or project, and that role is different in each phase. While each phase has its own specific priorities and needs, there are several Systems Engineering factors common to each phase.

## COST CONTROL OF SYSTEMS ENGINEERING

*The consequences of ineffective Systems Engineering at any stage of a project are cost escalations and schedule delays throughout the remainder of the project.* Visualize a typical GANTT Chart for a project. The total cost of the project is the sum of the costs of each line item. Each line item is the sum of its elements in a work breakdown structure. As each item is completed, the cost account is closed out accordingly. However, if a task is performed in an ineffective manner, time is wasted in subsequent tasks with corresponding unplanned cost escalations. The difference between estimated and actual costs is due to several parameters one of which is ineffective Systems Engineering. This Systems Engineering induced deviation occurs as a result of either bad estimates, or ineffective work. Eliminating ineffective Systems Engineering is akin to fixing defects in a manufacturing process.

Many companies in the Systems Engineering business seem to have little control over the quality of their Systems Engineering process and products. Poor quality inflates the cost of current projects and estimates for future projects. Developing effective Systems Engineers provides a company with a competitive edge both for fulfilling the needs of current projects and equally if not more important, for pricing the work to be carried out when writing proposals.

## METRICS

Systems Engineering Metrics are not easy to establish, perhaps that is why ineffective Systems Engineering is so prevalent. If you can't define a metric, think about what makes the process effective or what characteristic or factor leads to the desired results and use that as a basis for the first attempt at a metric. According to Dr. Joseph M. Juran, the ideal unit of measure (Juran 1988):

- Provides an agreed basis for decision making
- Is understandable
- Applies broadly
- Is susceptible of uniform interpretation
- Is economic to apply
- Is compatible with existing designs of sensors.

Since Systems Engineering is mainly process related, Metrics for Systems Engineering have to measure the degree of effectiveness of both the product and the process.

The effectiveness of the Systems Engineering product is measured by:

- The quality of the documentation and presentations.
- The degree of meeting specifications

The major by-products produced by Systems Engineers on their way to delivering the system are presentations and documentation. The customer does not usually see the system until after several presentations (reviews) have been made, and a considerable amount of documentation has been received.

The effectiveness of the Systems Engineering process is measured by:

- The degree of the budget and schedule overrun (due to the Systems Engineering component)
- The effectiveness of the Systems Engineers.

Determining the "right" metrics and using them will result in an improved process and a better product. The better product provides improved customer satisfaction. The improved process will ensure lower costs of doing business, which will provide a competitive edge for your organization in this highly competitive field.

Characteristics and parameters can be collected from current and past projects. These measure:

- The length of time specific tasks in the SDLC take
- the effectiveness of the people performing the tasks
- the quality of the product produced by the task.

If these parameters are identified, converted to metrics. Data associated with them can be collected, and analyzed. Once obtained, the metrics are useable on current projects, and for bidding proposals. In bidding for future work, the estimates will tend to be more accurate. This will tend to reduce cost overruns due to ineffective estimates.

Systems Engineers produce reviews and documentation as tangible products. Consider some of the factors that can be used to establishing metrics for both the Systems Engineering products, and the Systems Engineering process.

## REVIEWS

Reviews form an important part of the feedback process in which both the contractor and the customer participate to ensure that the work is proceeding correctly, that is, the contractor is producing the system that the customer wants.

*The purpose of a review is to formalize approval of decisions previously made*. It is not to present surprises or controversial design changes.

**Metrics for Reviews** One metric for reviews is the number of Review Item Discrepancies (RID) generated during or following a review. Since reviews with the same name (i.e. System Requirements Review) have different levels of complexity depending on the size of the system being reviewed, comparing numbers between different projects may not be useful. However, *measuring the number of RIDs does provide a metric for the degree of customer involvement in the process*.

Another metric is the amount of rework to be performed following the review, or the amount of work that has to be scrapped and replaced. In percentage terms, the metric can be used to compare the effectiveness of the Systems Engineering Team across different projects.

## DOCUMENTATION

Documentation is a major component of any system. It serves the purpose of communicating who is to do (or did) what, why, when, where, and how. Document quality is a function of style, format and content. Content is application specific. While there are guidelines as to layout, style and format of documents, and there are various methodologies for specifying the layout of a document, there are no precise guidelines on exactly what constitutes quality in the content of a document. Too often, the technical document preparation process takes the following form. A document is written in the form of a "brain dump". The author documents knowledge. This technique has the following effects:

- the document is written in the author's language, not the that of the users.
- the document contains gaps in the flow of information. These gaps are due to the detailed knowledge of the author which allows the writer to make a transition from one thought to an other, while the reader who does not have that background information is confused.
- the information is not presented in a logical order from a user's perspective.
- the document contains replicated and/or redundant information.
- the document contains the information the author writes, which is not necessarily the same as the information the user needs.

**Requirements for Writing Documents** The result of the above approach is a document that does not present the author's intent in a clear, concise and readable manner. W. Edwards Deming stressed the importance of avoiding errors in transactions in a service industry when he wrote (Deming 1982, 1986):

"*Production of an illegible figure anywhere along the line is as bad as starting off with defective material in manufacturing*."

If an illegible figure is bad, a defective document is much worse. Improve the process! Apply the following technique to perform the author's brain dump in a somewhat more structured manner and produce better documentation. If a document is written once, and meets the following requirements, the result will be a better document at a lower cost, due to the reduced number of changes in the review cycle.

- The information shall be written in the **customer/user's language**.
- The information within the document shall be **pertinent** to the reader.
- The information in a document shall be **complete**.
- All definitions shall be **unambiguous**.
- All information shall be **well organized**.
- All wording shall be **clear and concise.**
- **Redundant or replicated** information shall not be included in the document.
- All **specifications or requirements** shall be stated in a manner that makes them testable.
- The preparer shall use **annotated outlines** and obtain the appropriate concurrence for the contents of the document before starting to write the document.

Consider two categories of documents produced by Systems Engineers:

- Requirements Documents,
- Interface Control Documents.

**Requirements Documents** Requirements documents communicate "what is to be produced". They describe the system in terms of what it is supposed to do, rather than how it does it. They have to contain both extrinsic and intrinsic information. The extrinsic information is background or descriptive information that provides the context for the requirements, goals and priorities. The intrinsic information consists of each individual requirement.

Consider two examples of requirements extracted from a typical Data Archive and Distribution System (DADS) Requirements Document. The DADS ingests data, catalogs the data and stores them. Users access the system to browse, locate specific data and request copies of the data. The first example is:

```
307.1DADS shall guarantee accurate
    reconstruction of the data
    retrieved from the archive.
```

What does *accurate reconstruction* mean? The intent seems to be to make sure that what comes out of the archive is the same as what went in. How can this requirement be tested as written? Another example:

```
508.2When a user logs off with a
    request pending, DADS shall
    automatically notify the user
    that a request was completed the
    next time the user logs one.
```

Here, the intent seems to cover the following situation. A user on the system logs off with a request pending. the request completes while the user is logged off. The intent is that the user receive an automatic notification that the request completed when logging on again. However, look at the wording. The requirement is keyed to "*the next time the user logs one*" not the "*request was completed*". To meet the wording of the requirement, the system must be designed so the user gets a message "*that a request was completed*", the next time he or she logs on, <u>even if the log on takes place before the request completed</u>. The use of the wording "*a request*" does not link the notification to the specific request. Any message will meet the requirement. Lastly there may be a typographical error at the end of the sentence; the word "*one*" is used instead of "*on*". A better way to state the requirement is shown below.

```
508.2When a user logs on, DADS shall
    automatically provide the user
```

```
with the status of any pending
requests generated by that user.
```

Stating the requirement the latter way may also lead to a simpler design because the DADS software may not need to store the state of requests when users log off. When writing requirements, the needs of the design process which allocates the requirements to parts of the system and then implements them must be considered. For example, while the design team uses the requirements to design the system, the test team is also using the requirements to develop a plan for testing the system to verify that it will meet the requirements.

The allocation of requirements may take place in phases (i.e., the system may be implemented in pieces), each part being tested, then integrated with other parts. Requirements documents must be written and organized to facilitate the flowdown process. W. Edwards Deming writes (Deming 1982, 1986b):

"*I could do a much better job (fewer mistakes) if I knew what the program is to be used for. The specifications don't tell me what I need to know (programmer).*"

In keeping with this theme, requirements shall be written so as to be:

- complete,
- testable,
- relevant,
- achievable,
- allocated as a single thought to a single requirement paragraph (section number),
- grouped by function.

In addition, requirements *shall be traceable*, both upwards back to the source, and downwards into lower level documents.

Requirements shall not be written so as to be:

- redundant,
- overlapping,
- vague.

To help meet these requirements, the following words shall never appear within the text of a requirement:

- including,
- i.e.,
- etc.,
- will,
- must,
- should.

**Interface Control Documents** Interface Control Documents (ICD) describe and define interfaces at all levels. ICDs also contain intrinsic and extrinsic information. The purpose of an ICD is to communicate between the people on each side of that interface. Information in the ICD shall be relevant to the interface or to an understanding of the interface. The ICD shall contain:

- a description of where the interface exists within the system.
- a brief description of the system on each side of the interface as seen from the interface.
- a complete description of everything that crosses the interface.

## *METRICS FOR DOCUMENTS*

Metrics for documents relate to how well they communicate information. Metrics must be based on:

- the content of the document (how well it conforms to the requirements listed above)
- the completeness of the document (lack of absence of pertinent information).

One set of metrics is how many of the requirements for producing documents as listed above are violated in the document. Other metrics will be document dependent.

For example, a major set of metrics to try to ensure procedural document completeness is the number of times the Systems Engineer who produced a document is requested by telephone to solve a problem or answer a question for the user of that document. The more questions that are received by the System Engineer, the greater the indication that the published document or deliverable procedure is defective. If the users are unable to accomplish all of the tasks that are required by their jobs without that phone call, then the document is defective.

The phone logs of a Systems Engineer showed that in a typical month there are an average of 16 unplanned phone calls requesting support information. The Engineer takes about 15 minutes to produce the information to the satisfaction of the caller and an average of another 15 minutes to return to the work at hand. Adding up that time results in the loss of one day per month which may not sound like much in itself. Each phone call may interrupt another activity and cause a delay or an error to be made. These metrics should be used with caution, as in some instances it may be more cost effective to budget for the personal support than to produce a perfect document.

## EFFECTIVE SYSTEMS ENGINEERS

Systems Engineers need to be effective. On small projects, the functions performed by a Systems Engineer may be performed by an individual, or by two or more persons in the form of an interdependent group or team. When the functions are performed by a single person, the undocumented background knowledge the person gains is lost to the project when that person leaves the project. If the functions are performed by a team consisting of compatible people with complementary skills, synergism results. If the functions are split between lead and junior engineers, the knowledge will remain if one person leaves. This allows for subsequent re-assignment or promotion of personnel without seriously impacting the project.

Since it is difficult for one person to be accomplished in all the areas, teams of interdependent people, each with specialized knowledge in different areas would seem to be the optimal approach for Systems Engineering on a large project. When teams use participative management techniques, another benefit accrues. The team members tend to pick up the tasks they most enjoy doing, so those tasks are done well. The manager provides the guidelines and monitors the schedule.

**Metrics for Effective Systems Engineers** The baseline for developing metrics for effective Systems Engineers are the characteristics listed above for individuals. However, the metrics must appraise people not only for what they do as individuals, but also how well they work with the team, and how they grow other people in the project.

Finding objective metrics for people is not easy. Make a start with the characteristics of effective people. Effective people are (Covey 1989):

- **Proactive** - they choose and take responsibility for their actions.
- **Planners** - they develop a system concept, and use it as a framework for making decisions. They set priorities, formulate goals, then do the work.
- **Time managers** - allows people to identify and then concentrate on the tasks that need to be completed in a timely manner. This characteristic also includes the ability to manage time, keep track of action items, and ensure that decisions are made in a timely manner.
- **Seekers of mutual benefit solutions** - they look for win-win solutions, which means they explore options before making decisions that benefit all involved parties. Projects are completed by people working together. Negotiation skills and other such skills for working with people are critical (Lewicki et al. 1985).
- **Communicators** - they try to understand the other parties' point of view, and then communicate. Poor communications are the prime reasons for almost any kind of failure. Without a clear vision of the design, and the implementation path, it is impossible to get the project members all moving in the same direction, and keep them on that road. Verbal and non verbal communications skills are needed to ensure communications are clear, concise, and unambiguous (Sigband 1982).
- **Synergistic** - they work in teams, valuing the perspectives brought by individual differences.
- **Continually learning something new** - they continually renew and expand their skills.

Other characteristics are:

- **The education factor** - provides a measure of the understanding of the processes used during the SDLC and the degree of technical competence.
- **The motivation factor and the inherent ability within the person** - these provide a measure of the effectiveness of the person doing the work.
- **The experience factor** - provides a measure of the degree of sensitization to any potential problems.

For any part of the SDLC, the final metric is a combination of each of the individual characteristics listed above (and any applicable others) weighted by a factor specific to the situation. The experience and education factors have been used in an intuitive manner, since the number of Systems Engineers assigned to a project does vary during the SDLC.

## SUMMARY

This paper has discussed some of the characteristics that can be used to provide metrics for Systems Engineer and the Systems Engineering process.

## REFERENCES

Blanchard, Benjamin S. and Fabrycky, Wolter J., *Systems Engineering and Analysis*, Prentice Hall, 1981.

Eisner, Howard, *Computer Aided Systems Engineering*, Prentice Hall, 1988.

Deming, W. Edwards, *Out of the Crisis*, MIT Center for Advanced Engineering Study, 1982, 1986, p 422.

Deming, W. Edwards, *Out of the Crisis*, MIT Center for Advanced Engineering Study, 1982, 1986, p 87.

Covey, Steven R., *The 7 Habits of Highly Effective People*, Simon & Schuster, 1989.

Lewicki, Roy J. and Litterer, Joseph A., *Negotiation*, IRWIN, 1985.

Sigband, Norman B., *Communication for Management and Business*, Scott, Foresman and Company, 1982.

Juran, J.M., *Juran on Planning for Quality*, The Free Press, 1988, pp 76-78.

## AUTHOR'S BIOGRAPHIES

**Joe Kasser** has spent the last 20 years applying TQM to Systems Engineering resulting in the achievement of the cost effective implementation of international and domestic aerospace, communications and solar power systems. He is a recipient of NASA's Manned Space Flight Awareness (Silver Snoopy) Award for quality and technical excellence. He is also an ICPM Certified Manager and a recipient of ICPM's 1993 Distinguished Service Award. Parts of this paper were developed for his forthcoming book on "Applying TQM to Systems Engineering", and his proposed Doctoral dissertation on the same topic at The George Washington University.

**Robin Schermerhorn** has spent the last 12 years applying Software and Systems Engineering concepts on large communications software systems. She is a recipient of NASA's Manned Space Flight Awareness Award (Launch Honoree) for verification of NASA Communications systems. Parts of this paper were developed for her Master of Science in Computer Systems Management project on Systems Engineering Metrics at the University of Maryland.