

*CassBeth*

# Automated Specification Analysis

By

Walter Sobkiw

## Contact Information

[walt@cassbeth.com](mailto:walt@cassbeth.com)

[www.cassbeth.com](http://www.cassbeth.com)

**Copyright © 2009** Walter Sobkiw. This publication may be reproduced, photocopied, stored on a retrieval system, or transmitted as long as Walter Sobkiw or Walter Sobkiw and CassBeth are referenced. As a courtesy, please contact us if you intend to use this work.

# Automated Specification Analysis

by  
Walter Sobkiw

## Abstract

In 2005 a Carnegie Mellon Software Engineering Institute presentation was given to the local INCOSE Delaware Valley chapter on the possibilities of a tool for analyzing requirements. The premise was that the workflow includes requirements management and system modeling tools, but there were no tools to help users write and review requirements. In 2006 NASA started a study on the merits of using tools to support automated specification analysis. Both these efforts acknowledge that there is room for automation in the specification writing and review process. In response to this need a tool was created, the Specification Analysis Tool (SAT) to help users write and analyze specification text. This paper provides a brief history of SAT and describes its basic operations in various analysis settings. These analysis settings include, while writing a specification, during specification peer review, as part of specification delivery during quality assurance checks, while preparing for test using the specification, and Ad Hoc specification analysis.

## Introduction

There has been great progress in creating tools to manage requirements and model proposed systems. However many systems are implemented using specifications. The authoring of good specification text is not a trivial exercise and has been documented for many years. In 2005 Carnegie Mellon performed a study, Report CMU/SEI-2005-TR-014 (ref 1), and once again visited the dilemma of poor requirements text.

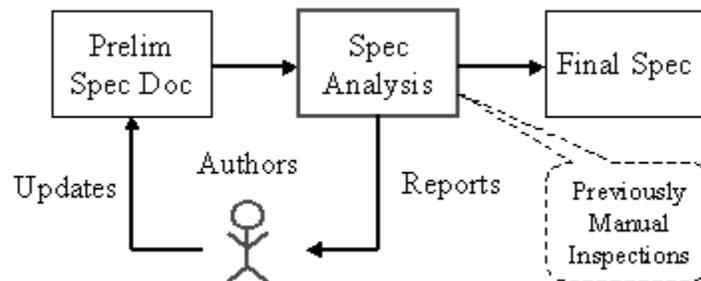
To illustrate the potential magnitude of the issue, over 50% of software defects are requirements problems and over 80% of rework is spent on requirements related defects. This is from a local INCOSE Delaware Valley chapter CMU SEI QuARS Presentation, INCOSE 21 June 05 referencing James Martin and Dean Leffingwell (ref 2).

In 2006 NASA entered the discussion and launched several studies to determine if automated tools could help in their Independent Verification and Validation (IVV) activities. They performed an industry survey and requested presentations from various organizations. Included in the study was the Specification Analysis Tool (SAT). Regardless of the tools and the individual advantages and disadvantages they made several observations and conclusions.

The review time was not shortened. That is because the time tends to be arbitrarily set. People will look for possible issues until they get bored or exhausted. The findings from a tool are more consistent. People tend to miss instances in categories while a tool tends to find all problems of a certain type. People tend to miss a whole category of a problem but are better at finding domain specific problems. There are more findings when a tool is used. The people and machines tend to complement each other.

## The Idea

Typically experts, novices, or some combination produce a specification document. They may or may not use a requirements style guide. At some point the staff decides if they will have a review. If they do not have a review, then they have to deal with the ramifications. If there is a review, then the review is a manual effort needing input from experts. Is there a way to automate the process of reviewing specification text?



The idea is to drop the specification text into a tool and have it produce reports. The findings both at Carnegie Mellon and NASA suggest that the tools complement the output from human reviewers by reducing their workload in areas where computers excel. Even without the formal study findings it is common sense that machines are good at searching, counting, filtering, categorizing, profiling, and visualizing while humans are good at creativity, critical thinking, inspiration, and intuition. For example, the machine allows the human to ignore 50 TBD findings, 25 compound requirements, or mixed units in a document etc.

## The Specification Analysis Tool

The Specification Analysis Tool (SAT) was created as a direct result of the original presentation given by Carnegie Mellon Software Engineering Institute at the local INCOSE Delaware Valley chapter in 2005. It was inspired when the question was asked, "where can we buy such a tool?". The answer did not yield a product and company response.

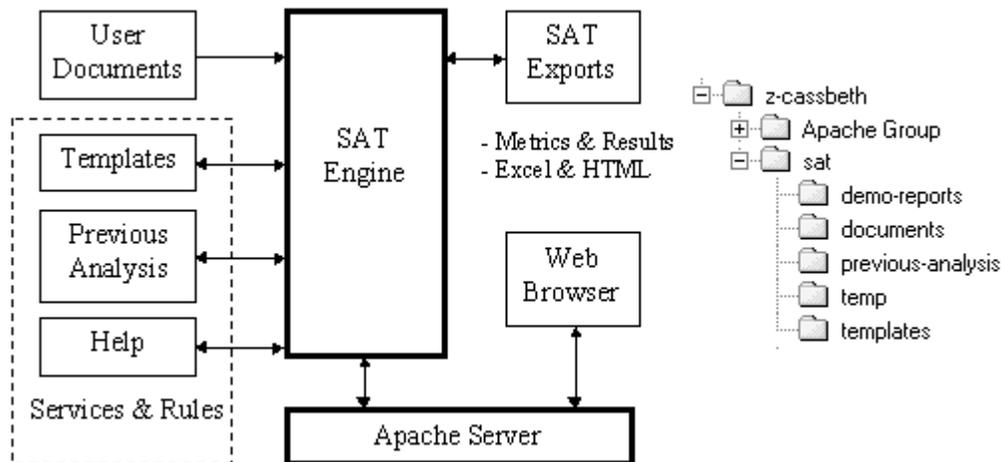
SAT is based on the following key requirements:

1. Its rules for finding bad requirements text should NOT be proprietary but open and visible to everyone. The rules should be easy to change and be coincident with an analysis run. This would allow the users to quickly tune the tool to their domain and adjust for potential false findings.
2. It should be fast. The analysis should be performed on the fly where the user might tweak rules and press the submit button every 30 - 60 seconds. Although large specifications may take more than 60 seconds to return results, the response is still within a reasonable time. It definitely should not be a batch or overnight run.

3. Do not discount the power of a machine to consistently search, count, organize, and duplicate search-based findings. It is better than using a word processor or a requirements database tool to search ad hoc during a specification review.
4. Let the tool reside and run on the user computer. This includes a simple install and remove mechanism and the ability to peer into where the tool is located. Allow the user to view the tool "libraries" using the standard operating system services.
5. Allow the user to load specifications and save results anywhere on the network. Allow anyone to duplicate a previously executed analysis run and modify that run as needed.

The SAT architecture is based on a compiled PERL engine that runs on the user computer. When the application is started the bundled Apache server starts, the default browser is opened, and the default analysis web page template is loaded. The user can use the default template or click into the Templates Library and load another template. It is anticipated that each organization will develop one or more templates and store them in the SAT template library.

Why PERL? - Because it has great regular expression processing. Why APACHE? - Because the Internet interface and web search engine experience has permeated our collective experience. In fact in many ways SAT is similar to a supercharged search engine with the ability to save off and access previous search / analysis experiences.



SAT will process the user document based on the loaded template or previous analysis settings. Previous analysis runs can be stored and re-executed on the same document or a new document using the same settings or different settings. Previous analysis runs also can be converted and treated as future templates. The templates are organized by services and each service is organized by rules. The user can create, modify, or delete services or rules when building a template of an analysis run.

## SAT Details

SAT comes with several templates. They have different services and rules. The default template offers a universal set of services and rules that may not apply in most organizations. The default template services are described as follows.

Requirement Text Analysis looks for word and phrase patterns that typically result in less than optimal requirements. In the past these rules have been encoded in checklists or kept in the heads of senior staff. Most organizations have different takes on these rules, and they tend to change with time. This is a basic service, but critical if the project expects to succeed. This should be the starting point of all the analysis.

Requirement Text Analysis  Show Search  Show Simple Rules  Show Complex Rules  
 Untestable  Multiple Imperatives  Options  Unsure  Unbounded  
...  Undefined  Compound Req  Internal Reference  Not Standalone  Fragment  
 Directive  Incomplete  Weak Phrases  Weak Words  Buzz Words

Most think of specification analysis within the framework of this service. There are several SAT templates available, which represent different views from different organizations on this topic. These templates are as interpreted for the SAT.

This analysis is based on a particular community and their view of what makes a good requirement. For example, tables in this community are considered a source of problems. Other communities feel that tables add to the understanding of a specification and encourage tables as a form of capturing the requirements. As a user of SAT you have the ability to easily change the rules to match your unique needs. The result is as follows:

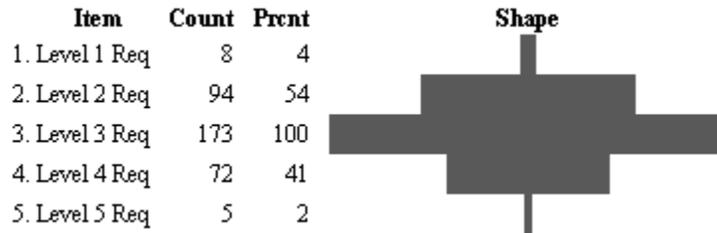
- 13. SAT-70** SAT may be packaged with the APACHE web server.  
...  **Vague I:** Requirement Text Analysis Risk: Item: **may** Instance: 1 Comment: There are different levels of vague. This is the worst type of vague. You must commit to your reqs and move anything that is vague to studies.
- 14. SAT-72** The directory structure of SAT is shown in **Figure TBD**.  
...  **Directive:** Requirement Text Analysis Risk: Item: **Figure** Instance: 1 Comment: Req's that are in a table or figure are not standalone. Some communities believe that tables and figures add understandability to a spec. However, in proof like settings of certification, they lead to missed tests and misinterpretation of contents. Figures and tables should be moved to design documents.  
...  **Incomplete:** Requirement Text Analysis Risk: Item: **TBD** Instance: 1 Comment: Incomplete specifications vary in severity. This is the worst case of incompleteness. Once the TBDs are gone more sophisticated techniques are needed to address this area. The other SAT services can help in this area along with SRDB parent child reports.

Not all of the above objects contain an imperative (e.g. shall). That is because a database attribute (IsReq) was used to identify an object as a requirement. The user can set their imperatives to anything and either process all the objects or only the objects with imperatives (shall, must, etc).

Find Duplicate Objects looks for duplicate objects. If a mask pattern is entered, similar objects will be shown as duplicates, provided the mask pattern is good. This situation tends to surface when people refuse to sue consistent names within a specification.

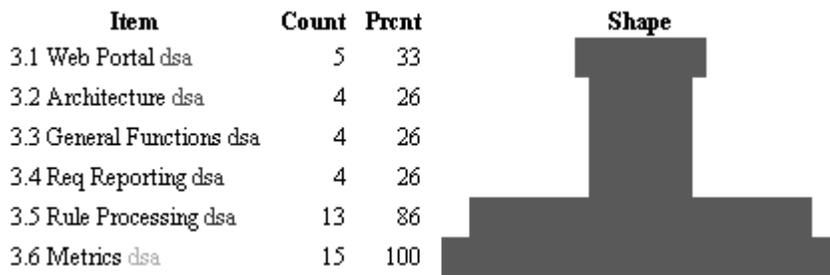
- 2. REQ-53, SAT shall allow a user to run different rules across the same document input.,IsReq
- ... Duplicate Imperative: REQ-68, SAT shall users to run different rules across the same

Generic Structure Analysis looks for and counts requirements at particular document paragraph levels. It works by triggering a count of requirements following an accessed pattern such as paragraph numbers. These are considered “children”. This is a higher level service that becomes useful when looking at decomposition levels for possible problems. The result is as follows:



The number of children at a particular level translates to a document shape. For example, if there are more children at lower document paragraph levels than higher levels, the document shape will be a pyramid, suggesting reasonable decomposition and leveling of requirements. There are different document shapes and each has implications. The document shapes are random, rectangle, pyramid, inverted pyramid, trapezoid and diamond.

Domain Structure Analysis is like Generic Structure Analysis except that the user tunes this service to their domain. This service looks for and counts requirements that represent domain specific requirements at each level of interest. It should be viewed as a narrower filter of the Generic Structure Analysis Service.



Again this is like the Generic Structure Analysis service but is applied to a unique domain. The number of children at a particular level translates to a document shape. There are different document shapes and each has implications. The document shapes are random, rectangle, pyramid, inverted pyramid, trapezoid and diamond.

Generic Capabilities Analysis This service looks for and counts all word instances in the accessed object. Selecting a word from the list and submitting the report will provide the related

text. This is a higher level service that relies on the human ability to recognize patterns when a word or group of words is provided.

- |                                     |                                      |
|-------------------------------------|--------------------------------------|
| <input type="radio"/> 30: Intensity | <input type="radio"/> 30: power      |
| <input type="radio"/> 21: unit      | <input type="radio"/> 18: lighting   |
| <input type="radio"/> 16: supply    | <input type="radio"/> 15: Test       |
| <input type="radio"/> 13: assembly  | <input type="radio"/> 13: discharge  |
| <input type="radio"/> 13: flash     | <input type="radio"/> 12: components |
| <input type="radio"/> 11: High      | <input type="radio"/> 11: following  |
| <input type="radio"/> 10: between   | <input type="radio"/> 10: power      |

This service paints a collection of words, first sorted by number, then by name. The hope is that this generic mining based on "shall" -green- then "non shall" -red- will yield patterns on capabilities within an unknown document. The words are then subjected to polysyllabic analysis to provide a reading level. This service allows each word in the report to be searched and returns objects associated with the word. This broadens the view allowing the analyst to further determine the capabilities.

Domain Capabilities Analysis looks for and counts all word or phrase instances set in the access field. Selecting a word from the list and submitting the report will provide the related text. These are user-defined words and the list needs to be built up during the analysis. This is a higher level service that relies on human ability to recognize their capability from a single word or phrase.

#### Number Sort

```
9 rule 7 web 2 browser 2 CGI 1 Web 1 HTML
1 internet 1 Rule
```

#### Accessed Patterns Not Found

```
login log in logoff log off
logon logout admin[^\s]* mine.*
database hmi windows motif
```

This is like the Generic Capabilities Analysis service but is applied to a unique domain. The most interesting part of the report is not what words and phrases are found in a document, but what words and phrases are not found, as shown at the bottom of the report area "Accessed Patterns Not Found"

Key Requirements Analysis lets a user find key requirements in a specification. The key requirements should be methodically sought by asking each stakeholder what they feel are their key requirements. This service looks for and counts all word or phrase instances set in the access field. Selecting a word from the list and submitting the report will provide the related text. This is a very high level service that relies on human ability to recognize their key requirement from a single word or phrase. This is really what it's all about - are the key requirements found in the document?

**Number Sort**

34 power	27 system	16 enclosure	10 System	9 temperature	9 systems
8 Emi	7 environment	6 emi	4 Power	4 Function	4 hardware
4 maintenance	3 Electrical	3 electrical	3 capable	2 Humidity	2 safe
2 salt fog	2 humidity	2 Temperature	2 Safe	2 EMI	2 Systems
2 shock	2 Shock	1 Environment	1 function	1 Salt Fog	1 Safety
1 performance	1 reliability	1 capability	1 mechanical	1 safety	1 data

**Accessed Patterns Not Found**

response time	architecture	distributed	central	hybrid	flexible
growth	flexibility	fails*safe	fails*over	recovery?	software
vibration	facility	facilities	room	training	train
maintainability	support	logistics	ils	maintainability	availability
EMC	TEMPEST	UL	Ergonomics	HMI	MMI
Machine Interface	ATCRBS	Beacon	Primary	Secondary	Mode-S
track	velocity vector	history	mosaic	flight plan	fix
posting	conflict alert	MSAW	altitude warning	plan probe	clearance delivery
security	secure	logs*on	logs*off	logs*in	logs*out
restricted area	track pairing	ADIZ	plot	maps?	range scale

The metrics of the Key Requirements service for an infrastructure system is contrasted with the SAT specification. The infrastructure system is:

Item	Count	Children	% lines	% imperative	% shall	% will	% isreq
1. Functional kra	9		2.13			60	
12. Maintenance kra	4		0.94	57.14	57.14	26.66	
14. RMA kra	1		0.23	14.28	14.28	6.66	
15. EMI kra	16		3.79				
2. Performance kra	1		0.23	14.28	14.28	6.66	
3. Systems kra	48		11.37				
5. Safety kra	4		0.94	57.14	57.14	26.66	
6. Software kra	1		0.23	14.28	14.28	6.66	
7. Hardware kra	10		2.36			66.66	
8. Environmental kra	65		15.4				
9. Mechanical kra	17		4.02				
z Mined Objects	134		31.75				

The SAT specification is:

Item	Count	Children	% lines	% imperative	% shall	% will	% isreq
1. Functional kra	2		3.33	6.45	6.45		
11. Training kra	1		1.66	3.22	3.22	100	
3. Systems kra	1		1.66	3.22	3.22	100	
4. Architecture kra	1		1.66	3.22	3.22	100	
6. Software kra	1		1.66	3.22	3.22	100	
z Mined Objects	6		10	19.35	19.35		

## Tuning

The user selects a service and presses the Submit button for an analysis run. The rules are shown when the service is selected. The user can enable and disable any rule within a service. If a rule needs to be modified, the user can select the appropriate check box, and the rule authoring options appear with the next press of the submit button. Each rule in the service can then be tuned using regular expressions and various option settings. The rules should be tuned to minimize false alarms and match the organizational view of a good specification.

## Display Filtering

In addition to the services and rules being enabled and disabled, there are various report areas that can be enable or disabled. The report areas are Analysis Results, Accessed Words, Accessed Patterns, Metrics, Doc Shape, Reading Level, and Comments.

Analysis Results is the main area that outputs the object text when there is a hit on a rule.

Accessed Words shows all the words in the mined objects that the rules accessed. The words are sorted by count and by alphanumeric sequence.

Accessed Patterns shows only the words in the mined objects that match the words in the rules. The words are sorted by count and by alphanumeric sequence.

Metrics is created when a service and a rule are enabled to support some analysis view.

Doc Shape is created when Count Child Objects is checked.

Reading Level provides reading level information.

Comments provide the service description, rule comments, and summarize the rule settings.

Between the rule setting and display filtering options, final reports can be produced and saved which identify what needs to be communicated to the team.

## Getting Started

To get started the users download and install SAT <sup>1</sup>. If this is a first experience with SAT the users should keep things simple and convert the specification to text format. The users start SAT, select the Requirement Text Service check box, and press the Submit button. The rules become visible and they are pre-checked ready for analysis. The users load the specification by using the Browse button and going to its previously saved location. The users press the Submit button and watch the analysis results.

As a new tool the users need to do a "what if" (ref 3). This is a period where they play with the tool and imagine what if the tool was added to their process. It goes from the simple button pressing to the complexity of the various tool reports. This is a good time to load several specifications into SAT and run several templates to develop a reasonable template for the organization.

## While Writing a Specification

As the users author specification text they can submit their work into SAT. The most effective service at this time is the Requirement Text Analysis service. The users can also offer suggested changes to the template.

At this time the template should move from a generic organizational view to a project specific view. The team leader should take on the role of ensuring that the template is moved to a project specific configuration. This includes populating the services unique to each project or domain:

- Domain Structure Analysis
- Domain Capabilities Analysis
- Key Reqs Analysis

## During a Peer Review

The team leader submits a specification for analysis prior to an internal peer review. The analysis reports should be saved so that everyone can review the results and discuss the findings at the peer review. One or more reports should be saved for each service in the template. For example, using the default template the following reports should be saved:

- Requirement Text Analysis.html
- Find Duplicate Objects.html
- Generic Structure Analysis.html
- Domain Structure Analysis.html
- Generic Capabilities Analysis.html
- Domain Capabilities Analysis.html
- Key Reqs Analysis.html

---

<sup>1</sup> A full version of SAT is available for free. There is also a paid licensed version. Just visit [www.cassbeth.com](http://www.cassbeth.com)

The final reports should maximize the rule tuning and display filtering features to minimize false returns. The object level display filter allows a user to make a go or no-go decision on each returned object in the report.

- ✓ 13. SAT-70 SAT may be packaged with the APACHE web server.
- ☐ 14. SAT-72 The directory structure of SAT is shown in **Figure TBD**.

## As Part of Delivery

Prior to delivery quality assurance (QA) typically gets involved in the release of a specification. QA can submit the specification using the same templates created by the system engineering staff or they can develop their own set of templates. This allows SAT to support the checks and balances inherent in an organization with a separate QA team.

## Preparing for Test

The testers in the organization can submit the specification for analysis from a test prospective. As in the case of QA they can work with the template used during the peer review and their own templates. Again, this allows SAT to support the checks and balances inherent in an organization with a separate test team.

## Ad Hoc Analysis

There are always questions. For example someone may have questions about EMI requirements in a previous system and their applicability to the current system. A user can load the previous specification, enable the Key Requirements service, and mine for EMI requirements.

While writing this paper I stumbled into the INCOSE glossary and it listed common verbs. I loaded the verbs into a new service and mined requirement text that contained imperatives but none of the verbs from the INCOSE glossary. I used the intermediate report to identify new verbs and added them to the rules. In the end this final specification for a delivered system showed that all its imperatives using "shall" or "must" had one or more verbs in the object. That was good. This was an interesting exercise that took 30 minutes to complete. The hardest part was getting the verbs from the INCOSE glossary in .pdf format into text format for entry into SAT.

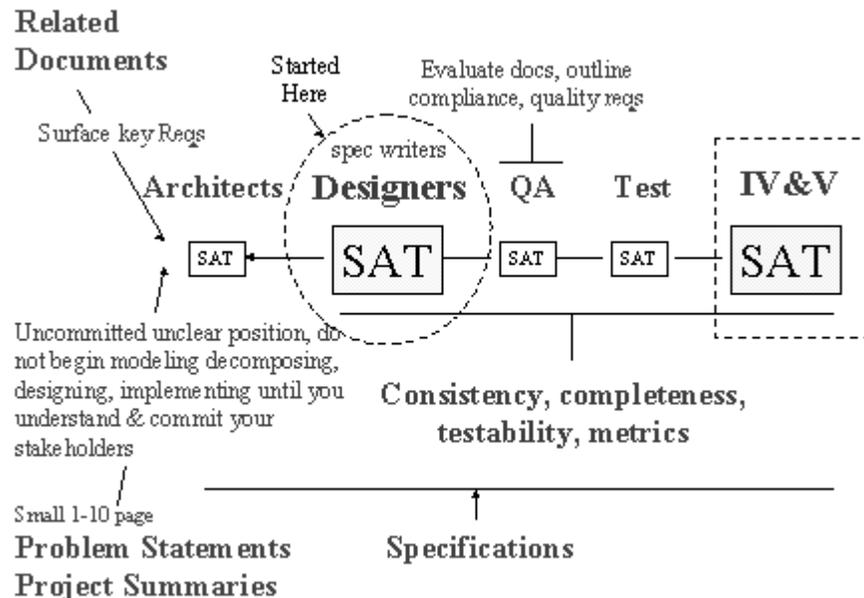
This new service can be added to the default template or offered in a new standalone template. As previously mentioned the goal was for the tool to make the rules non-proprietary and highly visible. There is no reason why these templates can not be exchanged between various communities.<sup>2</sup>

---

<sup>2</sup> A web site can be created where people upload and download templates for various specification analysis

## What Does It Mean

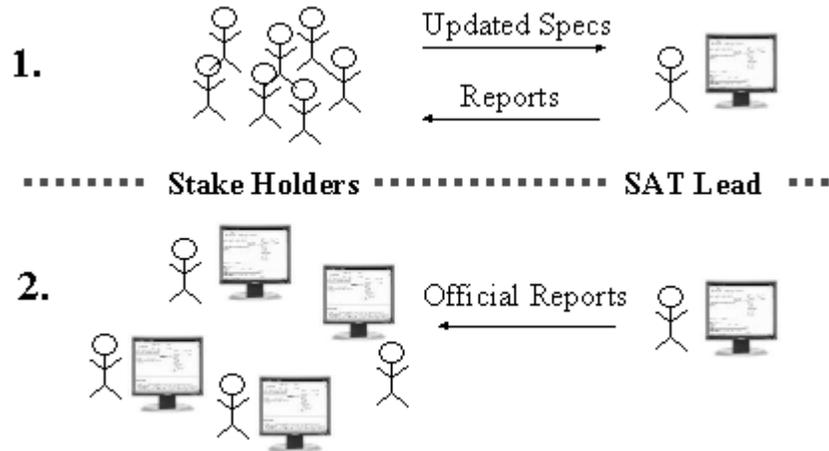
We are all very familiar with tools like word processing, spreadsheets, drawing, email, Internet and some are familiar with specialist tools like system requirements databases, modeling, software development and test tools. SAT is a new tool based on the idea of mining document content. What does it mean to introduce an engine with massive mining flexibility? Can it be configured to help people develop systems? The original purpose of SAT was to help people to review specifications. However what has surfaced is a more complex capability.



Ideally SAT should be introduced into an organization to support internal specification reviews using the basic Requirement Text Analysis service. As the organization gains experience with the concept of mining specification content, SAT then can be applied in other more esoteric areas, the most sophisticated service being the Key Requirements Service. SAT can also start to work with exports from the requirements database, and attributes can start to be added to the analysis sessions. For example if an attribute is used to identify an object as a requirement, that attribute can be used to tell SAT to process only those objects.

Management of SAT is up to the organization. The internal organization and its processes drive it, rather than any tool limitations. For example, there can be a single SAT installation with a single organizational focal point, or everyone in the organization can have an SAT installation. Configuration management of the reports is up to the organization, and how it allocates authority and responsibility for the tool reports. The SAT reports are like all other information products in the organization. Some reside on the network, and others are under formal configuration control.

Staff should not fixate on SAT analysis. Initially the tool should be tuned and reports produced within a few days. As time progresses and the organizational templates mature, this tuning and analysis should shrink to hours. SAT augments the manual review process, but does not replace it.



## Conclusions

The alternatives for dealing with the issue of creating a set of good specification requirements are (1) do nothing, (2) create a requirements style guide, (3) perform manual reviews, (4) use word searches in existing tools, or (5) use a tool like SAT.

One of the major advantages of SAT is that it is a shrink-wrapped product created specifically for this problem. Another advantage is that there is little processing time overhead. Staff does not need to spend hours setting up data and waiting for analysis runs. It has a fast easy install, does not run overnight, and lets anyone perform analysis at their desktop computer. Finally the reports can be tuned to the unique needs of the organization.

## Bibliography

1. Report CMU/SEI-2005-TR-014, Carnegie Mellon SEI Report September 2005
2. Local INCOSE Delaware Valley chapter CMU SEI QuARS Presentation, INCOSE 21 June 05 referencing James Martin and Dean Leffingwell
3. Sustainable Development Possible with Creative System Engineering, Walter Sobkiw, June 2008, Available from Amazon: ISBN 0615216307, [www.amazon.com/dp/ASIN/0615216307](http://www.amazon.com/dp/ASIN/0615216307)

## Biography

Mr. Walt Sobkiw received a BSEE from Drexel University in 1979. He has held various government, defense, and commercial positions at the FAA, Hughes Aircraft, E-Systems, Contraves-USA, Nielsen Media Research, MITRE and L-3 Communications. He has worked in the Air Defense, Air Traffic Control, Simulation Training, and Communications domains for over 30 years on systems with price tags ranging from billions to several hundred thousand dollars. At parties, when asked what he does for a living, Walt says he is a Systems Engineer and asks everyone to visit his web site [www.cassbeth.com](http://www.cassbeth.com) and buy his book.