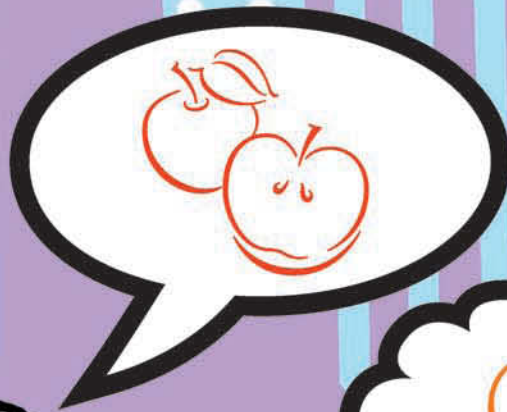


# What's It Mean?

*Reducing Imprecision to Improve Verification*

by David Gelperin





“... most people are not even aware of how ambiguous words can be. People are so skilled at resolving potential ambiguities that they don't realize that they are doing it.” — George A. Miller [1]

Language, both spoken and written, is easy to misunderstand. Imprecise language makes understanding and, therefore, verification more difficult. This article describes techniques for detecting and repairing vague and ambiguous software requirements.

According to San Jose State's Institute for Teaching and Learning [2], the definitions of vague and ambiguous are:

- A word or phrase is said to be vague if its meaning is not clear in context.
- A word or phrase is said to be ambiguous if it has at least two distinct meanings that make sense in context.

Successful software projects require 1) accurate specification (i.e., knowing what's needed), 2) successful communication among project stakeholders, and 3) cost-effective tactics for detecting and mitigating the inevitable defects in stakeholder understanding and communication. Mitigation tactics include early test design, prototyping, and development in small increments.

Imprecise language obscures defects in understanding and inhibits successful communication. Unfortunately, human language is inherently vague and ambiguous; it is inherently imprecise. Because project stakeholders are immersed in imprecise language, they are often unaware of the problem, which makes detection of imprecise language especially difficult.

### Vagueness

Note that “accurate specification,” “successful communication,” “cost-effective tactics,” “early design,” and “small increments,” while familiar phrases, are all vague. The boundaries between accurate and inaccurate, successful and unsuccessful, etc., are unclear. Vague words or phrases in a specification make it impossible to determine with confidence whether a system has a specified characteristic, although extreme failures can demonstrate that the characteristic is absent. For example, the exact meaning of “early test design” is unclear, but de-

signing tests after an entire product has been coded is clearly not early.

### Ambiguity

Natural language is filled with ambiguity as well as vagueness. There are several types of ambiguity. The most familiar are 1) words with multiple meanings (e.g., according to the *Oxford English Dictionary*, “set” has 192 definitions) and 2) ambiguous grammatical structures (e.g., John saw the girl with a telescope). Ambiguity such as the Yogi Berra quote “When you come to a fork in the road, pick it up,” or a headline such as “Enraged Cow Injures Farmer with Ax” may make us smile, but undetected ambiguity in a requirements specification is no joke.

### Imprecision

A single word or sentence can be both ambiguous and vague, such as the following signage: **Seniors get special rates.**

High school seniors? College seniors? Senior citizens? Exactly when does one become a senior? What is a special rate? How does one get this rate?

Ambiguity can be difficult to find but easy to fix. The existence of multiple interpretations may not be clear to a reader or listener, or he may incorrectly assume that the intended interpretation is obvious. Ambiguity can be detected in technical reviews by having a reader orally interpret the specifications in a group setting [3]. Richard Bender [4] provides details on a technique called ambiguity reviews in which documents are checked against a list of ambiguous words and phrases. Ambiguity is generally fixed by simply adding a few words to identify context or by restructuring the expression.

Vagueness is easier to find but harder to fix. A list of vague words could be created and used when reviewing specifications. Examples of such words are: most, few, early, late, sometimes, and rarely. Without discussion, a customer and a developer are unlikely to have the same understanding of *early* delivery or *most* accounts.

Pattern	Descriptive Elements	Domains (Phrases)	Examples
<b>Entity Profile</b>	Attributes, data types, occurrences, relationships	Nouns, unmodified	Order
		Nouns, modified	VAR controller
<b>Derived Value</b>	Equations	Nouns, modified	Average cycle time
<b>Derived Condition</b>	Boolean conditions on entity attributes	Nouns, modified	Active account
		Verbs, pre-modified	Rarely fail
<b>Quality Profile</b>	Measures, measurement processes, targets	Nouns, modified	Satisfied customer
		Verbs, pre-modified	Accurately identify
<b>Action Contract</b>	Post-, pre-, and constant conditions	Verbs, post-modified	Display complaints
		Verbs, pre-modified	Tentatively identify
<b>Derived Action</b>	Component actions or procedures	Verbs, post-modified	Plan vacation
<b>Event Profile</b>	Reporter questions (i.e., what, when, where, why, how)	Events	Accurately identify potential customers for our new system

Table 1: Rich definition patterns

## Rich Definitions

To repair vagueness, we introduce the concept of *richly defined phrases*. Rich definitions of phrases provide sufficient precision to enable verifiers to define required outcomes.

When vagueness is detected, it could be deleted, replaced, supplemented, marked and then supplemented, or defined with a precisising definition. According to Wikipedia.org:

A **precising definition** is a definition that extends the dictionary (lexical) definition of a term for a specific purpose by including additional criteria that further define the set of things meeting the definition.

For example, a dictionary may define the term “student” as 1) anyone attending an educational institution of any type, or 2) anyone who studies something. However, a movie theater may propose a *precising definition* for the word student of “any person

### Begin Derived Condition

**potential customers**

for our new blood analyzer

= bought any blood analyzer

OR bought any medical device within the past ten years

OR bought any products within the past three years

**bought any blood analyzer**

= True, if any purchased product code = BA

[Note: This is an example of a Derived Value [7]]

**bought any medical device within the past ten years**

= True, if any purchased product type = MD

AND purchase date later than (today - 10 yrs)

**bought any products within the past three years**

= True, if latest purchase date later than (today - 3 yrs)

**Sources:** Guy Sales, marketing manager

**Comments:**

**End Derived Condition**

Figure 1: A derived condition

under the age of 18 enrolled in a school” in order to determine who is eligible to receive discounted tickets.

This article advocates an approach to precisising called rich definitions [5, 6] of phrases and words. Rich definitions precisely define a variety of natural lan-



<b>Begin Entity Profile</b>		
identified potential customer [pre-modified noun] for our new blood analyzer		
Attribute Names	Data Types	Occurrences
<b>Customer Id</b>	Identifier	1
<b>Organization</b>	Name	1
<b>Selection Criterion</b>	(True) Boolean Function	1
<b>Customer Contact</b>	Name	1
<b>Contact Snail Mail Address</b>	Address	1
<b>Contact Email Address</b>	Email	1 or 2
<b>Contact Work Phone</b>	Phone Number	1
<b>Contact Fax</b>	Phone Number	1
Relationship Type	Entity	
<b>1:M</b>	Orders	
<b>Sources:</b> Guy Sales, marketing manager		
<b>Comments:</b>		
<b>End Entity Profile</b>		

Figure 2: An entity profile

guage expressions, as shown in table 1. We illustrate the use of the first five patterns—the most frequently used—in defining the meaning of a single sentence.

Dictionary definitions define words; rich definitions define phrases *and* words. Consider the phrase *active account*. The definitions of “active” and “account” do not provide sufficient understanding of the phrase. In software, this phrase will be instantiated to a Boolean selection function. Using a rich definition enables the accurate creation of this Boolean function.

An email from Guy Sales, the marketing manager, asks for a data-mining program that can scan the customer database and “accurately identify potential customers for our new blood analyzer.”

This requirement is neither verifiable nor implementable because the terminology, while clear, is imprecise. The phrases “accurately identify potential customers for our new blood analyzer,” “identify potential customers for our new blood analyzer” and “potential

customers for our new blood analyzer” must be precisely defined. The following shows how rich definitions can make each of these phrases precise.

### Derived Conditions

We begin by precisizing the phrase “*potential customers* for our new blood analyzer.” Since this phrase will be instantiated as a Boolean function in the program, we use Derived Condition from table 1 to specify this function as a cascade of expressions, as shown in figure 1.

Meaningful labels enable the cascade to show the rationale for the elements of the Boolean expression. Providing this rationale permits customers and other subject matter experts to check the soundness and completeness of the logic.

### Action Contracts

Now, we use Action Contract [8, 9, 10] from table 1 to precisely define the phrase “*identify* potential customers

for our new blood analyzer.” Action contracts define verb phrases with pre-, post-, and constant conditions. To develop appropriate post-conditions (i.e., conditions true at the end of an action that were not true at the start), you must understand what the action is meant to accomplish.

In this case, you need to know what will be done with the results of “identify” (i.e., whether the marketing department plans to communicate by email, snail mail, phone, or fax). The identified customer information is shown in the entity profile in figure 2.

We used an entity profile to specify the customer information resulting from the action “identify.” Now that we have a name and specification for the results, we can develop the action contract, as shown in figure 3.

We now have a precise enough understanding of “identify” to permit implementation and verification.

### Quality Profiles

Finally, we use a Quality Profile [11, 12, 13] from table 1 to precisely define the phrase “*accurately identify* potential customers for our new blood analyzer.” Quality profiles are used with vague noun phrases and vague verb phrases to provide precise satisfaction measures along with measurement methods and goals (i.e., verification techniques) as shown in figure 4.

### Using Rich Definitions

Diagram 1 shows the semantic structure of the sample sentence.

Rich definitions use natural language with minimal structure so as to be understandable to those familiar with the application domain. It is important that verifiers understand the definitions, but it is equally important that customers and other subject matter experts understand them, as well, so the definitions can be checked for accuracy and completeness.

Our experience in training employees, organized in their natural workgroups, has shown the power of rich definitions to catalyze the discovery of serious misunderstandings and significant implications. When people work to be precise, it catalyzes critical conversations, reduces misunderstandings, and reduces vague-

**Begin Action Contract**

identify

potential customers  
for our new blood analyzer

Post-conditions

A file exists containing 0 or more **identified potential customers**.

[About 15,000 potentials are expected]

A printed summary message contains the total number of identified customers  
as well as totals for each of the three selection criteria.

Constant-conditions [Constant conditions are always true.]

Read-only access to the customer and order history databases is available.

Pre-conditions [Pre-conditions are true at the beginning of an action, but not at the end.]

None

**Sources:** Guy Sales, marketing manager

**Comments:**

**End Action Contract**

Figure 3: An action contract

**Begin Quality Profile**

accurately identify

potential customers for our new blood analyzer

**QUALITY-A**

Accurately identify potential customers—identify with few mistakes

**MEASURE1**

% of identified potentials who buy a blood analyzer

**METHOD1**

Count the number of identified potentials and the number of these potentials  
who buy a blood analyzer up to the end of the first marketing campaign plus  
90 days.

**MIN goal1**

5% of the potentials

**TARGET goal1**

10% of the potentials

**Stretch goal1**

15% of the potentials

**MEASURE2**

% of unselected customers who buy a blood analyzer

**METHOD2**

Send an email notice to 30,000 randomly chosen unselected customers.  
Count the number of unselecteds who buy a blood analyzer up to the end of  
the first marketing campaign plus 90 days.

**MIN goal2**

no more than 5% of the unselecteds

**TARGET goal2**

no more than 3% of the unselecteds

**Stretch goal2**

no more than 1% of the unselecteds

**Sources:** Guy Sales, marketing manager

**Comments:**

**End Quality Profile**

Figure 4: A quality profile

accurately identify potential customers for our new blood analyzer

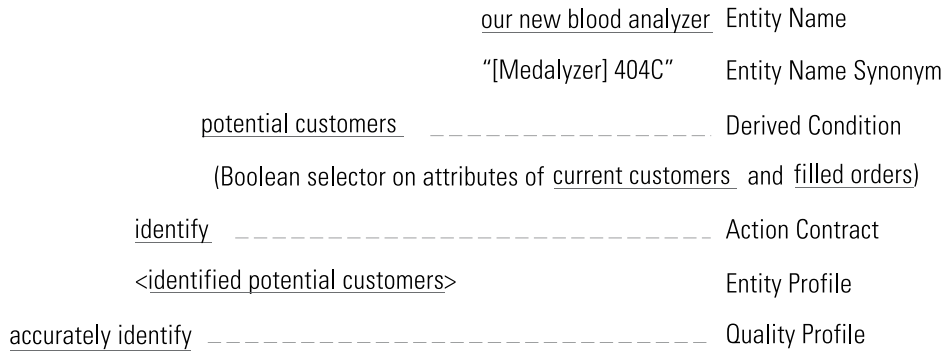


Diagram 1: Phrase semantics using rich definitions

Reduced cycle time	Strategy	Increased \$ cost per unit	Increased development time
3.2	Modify analysis process	0	4 weeks
2.8	Use faster standard processor	150	6 weeks
2.5	User faster custom processor	400	12 weeks

Table 2: Cycle-time options

ness in specifications.

If rich definitions are developed through group consensus, their value is directly proportional to the effort invested. Rich definitions that are difficult to develop are valuable—especially if developed early in the requirements process. Developing “difficult” definitions is a powerful risk-reduction tactic.

Many requirements texts recommend that a specification include a glossary. According to a Web poll [14] of current practice, 55 percent of respondents include a glossary 80 to 100 percent of the time, 20 percent of respondents include a glossary 40 to 79 percent of the time, and 25 percent of respondents include a glossary 0 to 39 percent of the time. Current specification glossaries primarily define single words. The following quotes suggest that such glossaries leave out most of the semantics of their application domain:

We define multi-word expressions (MWEs) very roughly as “idiosyncratic interpretations that cross

word boundaries (or spaces)”. [telephone booth, but not cell-phone booth or telephone closet.]

As Jackendoff [15] ... estimates ... the number of MWEs in a speaker’s lexicon is of the same order of magnitude as the number of single words.

In fact, it seems likely that this is an underestimate, ... Specialized domain vocabulary, ... overwhelmingly consists of MWEs, ... [16]

If you want to capture more meaning, include rich definitions in your glossary.

### Detecting Vagueness

Vagueness can be detected during technical reviews, but it is better for authors to detect vagueness during document creation. Automated tools are available that detect words that may indicate vagueness [17, 18], but the au-

thor must decide if there is actually a problem. Often, however, the author will not be able to repair vagueness without consultation. Research and consensus will be required.

### Using Intentional Vagueness

Sometimes vagueness indicates a conscious or unconscious decision to omit details. Sometimes it indicates ignorance of those details.

When entering a store to buy a new TV or computer, you hope to find a knowledgeable salesperson to explain options, advantages, and costs associated with an intended purchase. Telling the salesperson what you would like to know helps him help you. Likewise, when entering a new restaurant, you hope a knowledgeable server will provide recommendations. Depending on your food preferences, how hungry you are, and the prices, you will make a selection. It is natural for customers to seek information from those who know the details.

*“Intentional vagueness is appropriate when developers know as many or more details than the customer or when neither developers nor customer knows details. Politicians and diplomats as well as creators of advertising and propaganda are intentionally vague for different reasons. Note that they do not mark their vagueness.”*

This applies to customers of new or modified software systems, as well. Some requirements will be vague due to a lack of knowledge. Customers hope that developers will provide details as to options and costs. To explicitly express this hope, mark vague requirements such that developers will understand that they are to provide alternative interpretations.

Back to our marketing story: The product manager for a new blood analyzer specifies that the analysis cycle time must be *reduced* (vague). Let's assume the current cycle time is 3.5 minutes and the competitor's best is 3.1 minutes. The developers research the situation and respond that the average cycle time can be reduced as shown in table 2.

When statements are intentionally vague, they should be marked. Any marker signifying intentional vagueness may be used. The marker (and any associated comments) invokes a social contract between the customer providing the vague requirement and the developers designing the product. The developers are expected to identify alternative interpretations and the cost of each. The alternatives then are provided to the customer, who makes a selection as to the precise meaning of the requirement.

Intentional vagueness is appropriate when developers know as many or more details than the customer or when neither developers nor customer knows details.

Politicians and diplomats as well as creators of advertising and propaganda are intentionally vague for different reasons. Note that they do not mark their vagueness.

## Conclusion

To promote greater awareness, a national safety organization provides bumper stickers saying: Start Seeing Motorcyclists. Similarly, this article encourages you to: Start Seeing Imprecision. Many believe that a fish is unaware of the water in which it swims. Because we are immersed in imprecise language, we need to work hard (perhaps aided by tools) to become aware.

Whether plagued by vagueness or ambiguity, verification will improve when you find and fix imprecision. Repairing vagueness with rich definitions will catalyze critical conversations. In addition, customers should use intentional vagueness to trigger research that identifies cost-effective interpretations. **{end}**

## REFERENCES

- [1] Miller, George A. "Ambiguous Words." KurzweilAI.net, 2001. [www.kurzweilai.net/meme/frame.html?main=/articles/art0186.html](http://www.kurzweilai.net/meme/frame.html?main=/articles/art0186.html)
- [2] "Introduction to Vagueness and Ambiguity." San Jose State University Institute for Teaching and Learning. [www.sjsu.edu/depts/itl/graphics/vague/vague.html](http://www.sjsu.edu/depts/itl/graphics/vague/vague.html)
- [3] Kamsties, Erik, Daniel M. Berry, and Barbara Paech. "Detecting Ambiguities in Requirements Documents Using Inspections." Proceedings of the First Workshop on Inspection in Software Engineering, 2001. [se.uwaterloo.ca/~dberry/FTP\\_SITE/tech.reports/kamsties.berry.paech.pdf](http://se.uwaterloo.ca/~dberry/FTP_SITE/tech.reports/kamsties.berry.paech.pdf)
- [4] Bender, Richard. "Improving Requirements Through Testing." [www.stickyminds.com/s.asp?F=S6854\\_ART\\_2](http://www.stickyminds.com/s.asp?F=S6854_ART_2)
- [5] Gelperin, David. "Let Me Be Perfectly Clear." ClearSpecs Enterprises Technical Papers. [www.ClearSpecs.com](http://www.ClearSpecs.com)
- [6] Gelperin, David. "Just Enough Precision," Webinar Q and A. ClearSpecs Enterprises Technical papers. [www.ClearSpecs.com](http://www.ClearSpecs.com)
- [7] "Defining Business Rules—What Are They Really?" The Business Rules Group, July 2000. [businessrulesgroup.org/first\\_paper/BRG-whatisBR\\_3ed.pdf](http://businessrulesgroup.org/first_paper/BRG-whatisBR_3ed.pdf)
- [8] Gelperin, David. "Specifying Consequences with Action Contracts." ClearSpecs Enterprises Technical Papers. [www.ClearSpecs.com](http://www.ClearSpecs.com)
- [9] Meyer, Bertrand. "Building Bug-free O-O Software: An Introduction to Design by Contract." [archive.eiffel.com/doc/manuals/technology/contract/page.html](http://archive.eiffel.com/doc/manuals/technology/contract/page.html)
- [10] Meyer, Bertrand. "Applying Design by Contract." IEEE Computer, Vol. 25 No. 10, October 1992, pp. 40-51.
- [11] Gilb, Tom. *Competitive Engineering: A Handbook For Systems Engineering, Requirements Engineering, and Software Engineering Using Planguage*. Butterworth-Heinemann, 2005.
- [12] Gilb, Tom. "Advanced Requirements Specification: Quantifying the Qualitative." [citeseer.ist.psu.edu/332850.html](http://citeseer.ist.psu.edu/332850.html)
- [13] Simmons, Eric. "Quantifying Quality Requirements using Planguage." ClearSpecs Enterprises Technical Papers. [www.ClearSpecs.com](http://www.ClearSpecs.com)
- [14] State of the Practice Survey. ClearSpecs Enterprises. [www.ClearSpecs.com](http://www.ClearSpecs.com)
- [15] Jackendoff, Ray. *The Architecture of the Language Faculty*. MIT Press, 1997.
- [16] Sag, Ivan A., Timothy Baldwin, Francis Bond, Ann Copestake, and Dan Flickinger. "Multiword Expressions: A Pain in the Neck for NLP." *Lecture Notes In Computer Science*; Vol. 2276. Proceedings of the Third International Conference on Computational Linguistics and Intelligent Text Processing, Springer-Verlag, 2002.
- [17] Gelperin, David. TEKchecker User Guide. ClearSpecs Enterprises. [www.ClearSpecs.com](http://www.ClearSpecs.com)
- [18] Gelperin, David. StyleWriter & TEKchecker Video. ClearSpecs Enterprises. [www.ClearSpecs.com](http://www.ClearSpecs.com)



Want searchable access to every article ever published in *Better Software* magazine? The StickyMinds.com PowerPass gets you there. Visit [StickyMinds.com/powerpass](http://StickyMinds.com/powerpass) to learn more.