The Use of Jargon in Software Requirements

YouKyung (Kim) TeStrake
Department of Computer Science
University of Northern Iowa
testr000@cns.uni.edu

Abstract

The use of jargon in the computer industry has justifiably been accused of being a leading cause of miscommunication. Jargon is perceived to be a technical group's tool to shift or avoid responsibility. It has also been portrayed as a tool to hide facts from the customers and for covering up faults of the computer industry.

This paper examined how analysts and customers perceive jargon differently and suggests ways to minimize the miscommunication that too often occurs in the software industry. This paper includes a study conducted at the University of Northern Iowa. The results of the study indicate that a significant level of knowledge and perception gaps exist between future software engineers and customers, and that male and female software engineers hold different views when it comes to the use of jargon.

Introduction

Numerous studies have reported the negative impacts of computer system failure on human life. The causes leading to these failures vary from one project to another. The most publicly criticized causes are computer hardware system failure and human operators' errors. However, a number of researchers [1] have shown that the errors committed in the software programs are the primary causes of these system failures. In addition, errors committed in requirement management are key contributors to these software failures.

In their intensive review of 8380 software projects, the Standish Group [2] denotes incomplete requirements and specifications to be one of the two top reasons why software failed. Leffingwell [3] points out that 40 to 60% of software project defects are due to errors made during the requirement stages. In his massive collection of software runaway cases, Glass [4] concludes that 51% of software failures are due to an incompletely stated requirement.

The goal of the requirement phase is to achieve agreement on the views of users and developers concerning software products. Therefore, communication between the customer group and requirement analysis group seems to be the key component in writing a precise software requirement. However, studies indicate that a significant communication gap exists between these two groups. In his software requirement specification review, Rogers argues that a majority of the customer requirements are poorly articulated and misguide the developers. As a consequence, what developers think they are supposed to build is different from what customers think they are going to get. Minasi [5] blames the computer industry for this miscommunication and argues that "the computer industry has traditionally used jargon and technical details to separate itself from its customer." Weiss [6] also argues that analysts and experts are responsible for this miscommunication. According to Weiss, analysts know too many technical details concerning how the product works, and assumes too much about the customers' knowledge resulting in confusion on behalf of the less knowledgeable customers. Thus while intending to aid the customers in understanding product, requirement analysts have increased the likely hood of miscommunication. It is important for analysts to carefully consider how technical details are expressed.

In his enumeration of a "Requirement Bill of Rights" for the software customers, Wiegers [7] states that the first right for customers is the right to "expect analysts to speak the customer's language." Wiegers also suggests that requirement analysts should avoid jargon when dealing with customers. This implies that a significant level of miscommunication originates with the technical jargon used by analysts. The purpose of this paper is to examine how analysts and customers perceive jargon differently and thereby suggests ways to minimize the miscommunication that too often occurs in the software industry.

Jargon

Jargon, slang, and cant are the most commonly used terms to describe a special vocabulary used by particular groups. Although these terms share the same meaning in a broad sense, a specific group is associated with each of these terms. For instance, cant is a term referring to an informal vocabulary used by gangsters in the criminal world. Slang refers to an overall informal vocabulary while jargon is reserved to refer to technical or professional vocabularies of various occupations [8].

Functions of jargon

Every profession has its own jargon, and this has raised a question regarding its purpose. Lutz [9] and Wallraff [10] enumerate several positive functions of jargon. First of all, they both state that the use of jargon symbolizes membership. Using jargon within a group provides members with a sense of security and a sense of being included. In this sense it fulfills the psychological need of differentiating one group from another. Also, jargon provides efficiency and clarity within the members' communication with each other [11]. Since a few words of jargon can convey a lot of knowledge [12], the use of jargon is perceived as a convenient shorthand [13].

Third, the use of jargon adds technical accuracy and quality to the language [14]. Therefore, jargon functions as a tool for improving communication within a group. Finally, jargon produces an "air of profundity, authority, and prestige for speakers and their subject matter" [15]. As a result, the use of jargon helps members build "a culture of respect, trust, growth, learning, and honor" [16]. These positive functions of jargon are also applied to the field of computer science. The use of jargon allows software developers to maintain a "closed club" in which they control membership through the use of their own particular jargon.

However, when jargon is used in the communication between a technical group and ordinary people, the positive functions of jargon disappear. One of the most widely discussed trials in years to come will be the Microsoft antitrust case. During this trial, reporters criticized Microsoft for its use of jargon. Prosecutors claim that the use of jargon by Microsoft was part of an attempt to confound both the judge and the lawyers during the trial [17]. Prosecutors often interrupted testimony by Microsoft officials in order to clarify the jargon being used. If the intention of using jargon is to achieve efficiency and technical clarity, the outcome of these efforts is quite the opposite.

Lutz and Wallraff describe negative images of jargon when it is used in an inappropriate context. First of all, it functions as a tool for impression rather than expression. Lutz reports that in order to impress their audience, technicians deliberately use jargon in a situation where a simple English expression would have functioned better for communication. Wallraff argues that jargon used by technical groups intimidates ordinary listeners, claiming that a non-technical group will more likely to develop a sense of insecurity and anxiety when the use of jargon exceeds a certain level. This intimidation causes the ordinary audience to avoid any confrontation with the technical group.

In certain circumstances, jargon is perceived to be a technical group's tool to shift or avoid responsibility. The use of jargon plays a role of hiding facts from outsiders and covering up faults of the insiders [18]. Waller [19] reports that "all professionals tend to screen their knowledge, or their lack of it, by using jargon." This tendency is observed more often in a meeting with customers rather than in conversations with their peer group [20].

Jargon in the Computer Industry

The use of jargon in the computer industry has justifiably been accused of being a leading cause of miscommunication [21]. It has also been portrayed as a tool for a computer industry to separate itself from its customers [22].

In his column, Tomlinson shares his experience of being accused of abusing jargon, and provides insight from people in the computer industry regarding the use of jargon. According to Tomlinson, the use of jargon is justified, stating that technicians use jargon mainly to save time.

The popularity of the Internet and software programs have exposed their users to a massive quantity of computer jargon [23]. For example, not many people will go out to assemble a wood stove when they are asked to *log in* [24]. Nor would they catch a mouse hiding behind a wall to *click the mouse*. Language reflects the times as well as the people living in them [25]. The above two expressions are the examples of computer jargon accepted by the general public in the wake of "high-technology revolution" [26].

There are more examples of computer jargon widely used by common computer users. For example, the U.S. Bureau of Census describes their 2000 U.S. Census form as *user-friendly*. User-friendly is computer jargon meaning "easy to use" [27]. Another example of computer jargon commonly used by computer users is *e-mail*. Few people will stop their conversations to clarify the meaning of e-mail.

These and other successful adoptions of computer jargon by the general public have encouraged its use in the computer science field. In light of this, both Weiss' and Minasi's claims are understandable. They imply that people in the computer industry assume too much about their customers' ability to understand their language. These assumptions have justified the use of jargon in the eyes of computer professionals.

Study

Two different groups were studied in this experiment. One group representing software engineers was selected from students majoring in computer science and currently enrolled in a class titled “Software Engineering” at the University of Northern Iowa in the fall semester of 2000. Software Engineering is a mandatory class for those who commit themselves to becoming software engineers. Also, the given class requires junior standing in computer science as a prerequisite for admission. Therefore, these students already have a couple of years of computing and programming experience. This is the reason why this experiment selects this particular group of students and believes that they qualify adequately to represent software engineers.

The other group representing the general public was selected from students who were enrolled in teacher education at the University of Northern Iowa during the fall semester of 2000. In order to qualify the subject’s experience in using software products, this study selected the subjects representing the customers from the students who are currently taking an “Educational Media” class at the University of Northern Iowa in the fall semester of 2000. Educational Media introduces current educational technology to future teachers. Also, students enrolled in this class learn skills to evaluate educational software products. Therefore, this study concludes that the students enrolled in this class adequately represent customers dealing with educational software products. Educational Media is a mandatory class for students who want to earn a teacher’s license issued by the state of Iowa.

Size of Sample

In the fall semester of 2000, 3,414 students were enrolled in the teacher education program and 358 students were attending the Educational Media class at the University of Northern Iowa. The latter number represents 10.5% of total population of students involved with the teacher education program at the University of Northern Iowa.

One hundred sixty-eight students involved with the computer science program were classified as juniors and seniors. Among these, 39 students were enrolled in the Software Engineering class in the fall semester of 2000 at the University of Northern. The latter number represents 23.2% of total population of students majoring in computer science at the University of Northern Iowa. Six of these respondents were graduate students in computer science.

Male and female ratio among the participants

Among the 27 female students classified as a junior or senior in the department of computer science, seven participated in this survey. The latter number represents 25.9% of the total female junior and senior population in the department of computer science. Twenty-six male students classified as a junior or senior in the department of computer science participated in this survey. This latter number represents 18.4% of total male junior and senior population in the department of computer science. The overall ratio between male and female students majoring in computer science of junior and senior standing is 83.9% to 16.1%. The ratio between male and female students who participated in this survey is 78.8% to 21.2%.

Among the 3,414 students who were majoring in teacher's education at the University of Northern Iowa, 2,000 were female and 1,014 students were male. Seventy-six male students and 248 female students participated in the survey. The overall ratio between male and female students involved in teacher's education at the University of Northern Iowa is 29.7% to 70.3%. The ratio between male and female students who participated in this survey was 23% to 77%.

Age differences among the participants

Twenty-two participants from the Software Engineering class were between the ages of 21 and 23. Also, 11 students were between the ages of 24 and 30, and six students were above 31 years of age. Three hundred participants from the Educational Media class were between the ages of 19 and 23. Also, 16 students were between the ages of 24 and 30, and seven students were above 31 years old. One student did not identify her age.

The Objectives and Methods of the Experiment

This study was conducted as a blind survey form. Two different survey forms were prepared for these groups. The first part of each survey form was designed to assess the backgrounds of the respondents. The customer survey form consisted of questions asking the subject's major, age, gender, years in using software products, and class standing (Junior, Senior, etc.). The software engineer survey form consisted of questions asking the subject's age, gender, and class standing.

The second part of the survey form was designed to examine the subjects' awareness of computer jargon commonly used in software industries. For this purpose fifteen computer terms were selected. These included *user-friendly, crunch mode, hidden file, source code, program hangs, system crash, end-user, downtime, IP Address, illegal operation, hot key, Trojan Horse, computer worm, GUI,* and *Gigabyte*. These terms were chosen from <u>Dictionary of Computer and Internet Terms</u> (6th ed.) [28] and <u>Computer Concepts</u> (3rd ed.) [29].

Two different questions were provided for the two groups of students who participated in this survey. The students attending the Educational Media class were asked to circle the number next to the computer term that they were familiar with (in other words, they had heard the term before in daily conversation and recognized them although they could not provide its definition). The same list of computer terms was used in the survey form for the students enrolled in the Software Engineering class. However, they were asked to circle the number next to the computer terms that they would hesitate to use in their communication with potential customers.

Two questions were used to assess the software engineer group's views regarding the use of computer jargon (see Table 1). The first question asked the participants to circle the number next to the reason for not using the computer jargon in their communication with potential customers. Three reasons were provided for this question. The second question asked the participants to circle the number next to the reason for using computer jargon. Six reasons were provided for this question. These reasons were selected from Douglas and Douglas', Lutz's, and Wallraff's descriptions of positive and negative functions of jargon. The results and findings of this survey will be described below.

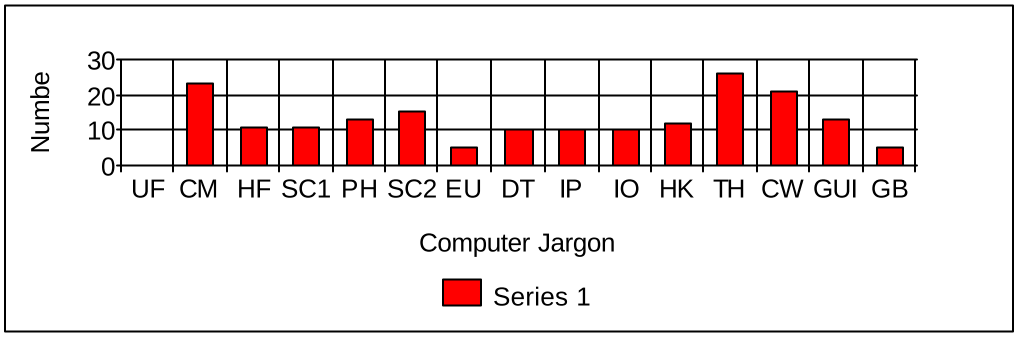
Table 1: Questionnaire from the Survey

Questions and Choices
E. Why would you hesitate to use the terms you just circled in question D?
1. Because I don't know the terms myself
2. Because I assume that customers would not know the terms
3. Because these terms are too technical
F. Why would you use the terms you did not circle in question D?
1. Because there is no alternative way to say it
2. Because I want to save time
3. Because I want to provide technical accuracy
4. Because I assume that customers would know the term
5. Because I want to give an impression to my potential customer that I am an expert in this domain
6. Because the customer used the term first

Summary of the Survey

The Software Engineer Group

The first question in the survey asked the participants representing the software engineering group to circle the number next to the computer jargon that they would hesitate to use in communication with their potential customers. The result of the survey question 1 is summarized in Figure 1.



Note.	UF: User-friendly	CM: Crunch mode	HF: Hidden file
	SC1: Source code	PH: Program hangs	SC2: System crash
	EU: End-user	DT: Downtime	IP: IP Address
	IO: Illegal operation	HK: Hot key	TH: Trojan Horse
	CW: Computer worm	GUI: GUI	GB: Gigabyte

Figure 1: Number of times rejected

Twenty-six out of 39 participants rejected *Trojan Horse (TH)* as being unsuitable for use in communication with their customers. Also, twenty-three participants selected *crunch mode (CM)*. Twenty-one participants replied that they would hesitate to use *computer worm (CW)*. Fifteen respondents selected *system crash (SC).*

As Figure 1 shows, participants answered that *user-friendly (UF)* was the term that they would not hesitate to use in communication with their potential customers. Also, most participants replied that they would not hesitate to use the terms such as *end-user (EU)* and *gigabyte (GB)* in communication with their potential customers.

On average, 4.74 terms were rejected by the respondents as being unsuitable for use with their customers. However, an interesting pattern appeared between male and female respondents. While female respondents replied that an average of 6 items were unsuitable, male respondents chose 4 items. Figure 2 represents the difference between the amount of jargon rejected by male and female respondents. Seven male respondents replied that they would hesitate to use one computer jargon in communication with their customers. On the other hand, two female respondents chose 11 terms as being unsuitable for use with their customers. Overall, 5 out of 9 female respondents replied that they would hesitate to use seven or more items of computer jargon listed in the survey. This number represents 55.5% of female respondents involved in this survey. Figure 5 also indicates that 4 out of 30 male respondents replied that they would hesitate to use seven or more items of computer jargon listed in the survey. This number represents only 13% of male respondents. This suggests that female respondents are more likely hesitate using computer jargon in communication with their potential customers than their male counterparts.

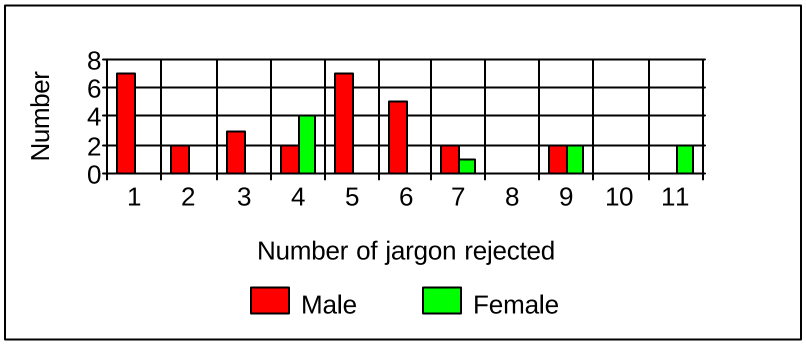


Figure 2: Comparison between male and female: Number of terms rejected

The second question in the survey asked the respondents why they would hesitate to use computer jargon in communication with their potential customers. Although the respondents could choose more than one answer for this question, this paper summarized only the respondents' primary reason for not using computer jargon.

Eighteen out of 39 respondents replied that they would hesitate to use computer jargon because their customers would not know the term. This number represents 50% of respondents who provided an answer for this question. Eleven out of 39 respondents replied that they would hesitate to use computer jargon because the terms were too technical. This number represents 30.6% of respondents. Finally, seven respondents replied that they would hesitate to use computer jargon because students enrolled in the software engineering class did not know the terms themselves. This number represents 19.4% of respondents. Three respondents disagreed with the choices this survey offered and did not provide an answer (see Figure 3).

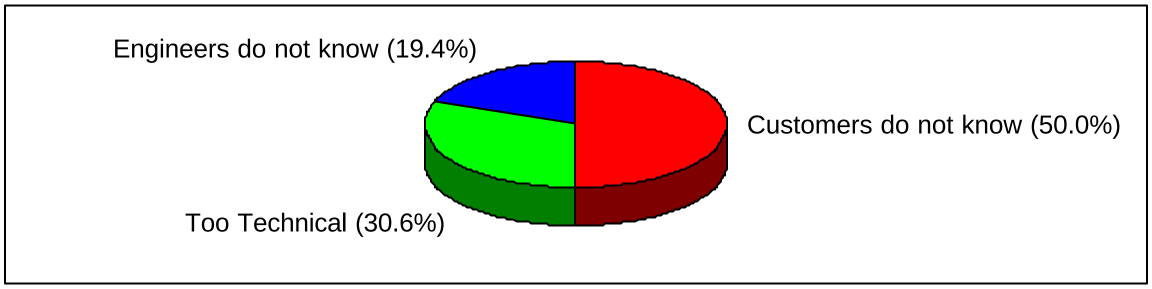


Figure 3: Reasons for not using computer jargon.

The female respondents' primary reasons for not using computer jargon were different from those of male respondents. Female respondents mostly hesitated to use computer jargon because the terms were too technical. Five out of nine female respondents replied that they would hesitate to use jargon for this reason. This number represents 55.6% of female respondents involved in this survey.

Male respondents, on the other hand, replied that the primary reason for not using computer jargon in communication with their potential customers was because their customers would not know the terms. Seventeen out of 30 male respondents agreed with this choice. This number represents 63.0% of male respondents who provided an answer for this question.

The third question asked the respondents to provide reasons for preferring computer jargon to regular English expressions. Two out of 39 participants in this survey disagreed with the choices this survey offered and did not provide an answer. Among the 37 respondents, 15 replied that a primary reason for using computer jargon was because they assumed that customers would know the terms. This number represents 40.5% of respondents who provided an answer for this question.

Thirteen out of 37 respondents who provided an answer for this question replied that a primary reason for using computer jargon was because they wanted to give an impression to their potential customers that they were experts in this domain. This number represents 35.1% of respondents who provided an answer for this question. Nine students replied that a primary reason for using computer jargon in communication with their potential customers was because they wanted to provide technical accuracy. This number represents 24.3% of respondents (see Figure 4).

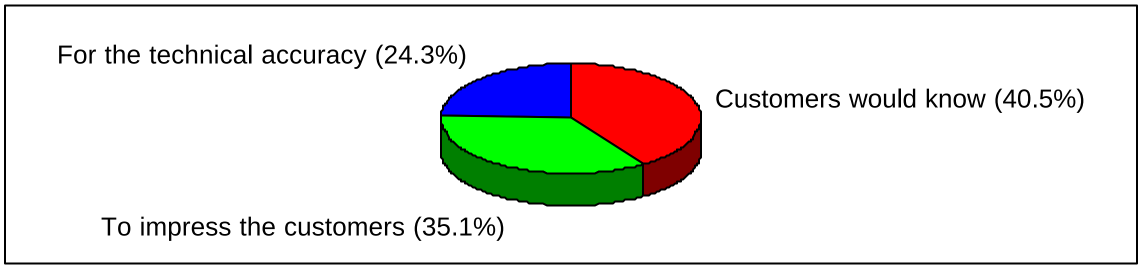


Figure 4: Reasons for using computer jargon.

Two implications can be gleaned from Figure 4. First of all, 40.5% of respondents replied that a primary reason for using computer jargon was because they assumed that their customers would know the terms. The same respondents replied that they would use *user-friendly, end-user*, and *gigabyte* in communication with their potential customers. Therefore, if their assumption is correct, students representing the customer group will identify the terms such as *user-friendly, end-user*, and *gigabyte* without any difficulties. As will be shown later, this assumption is not true.

Also, the respondents seem to have a perception that the use of computer jargon is a way of indicating their domain knowledge to their customers. The result of this survey supports Wallraff's and Lutz's claims that jargon functions as a tool for impression rather than for expression.

Finally, this survey includes an overall performance of the students currently enrolled in the Software Engineering class regarding the awareness of computer jargon. On average the respondents matched 10 to 11 computer terms with their correct definitions. Male respondents performed slightly better than did female respondents.

Customer Group

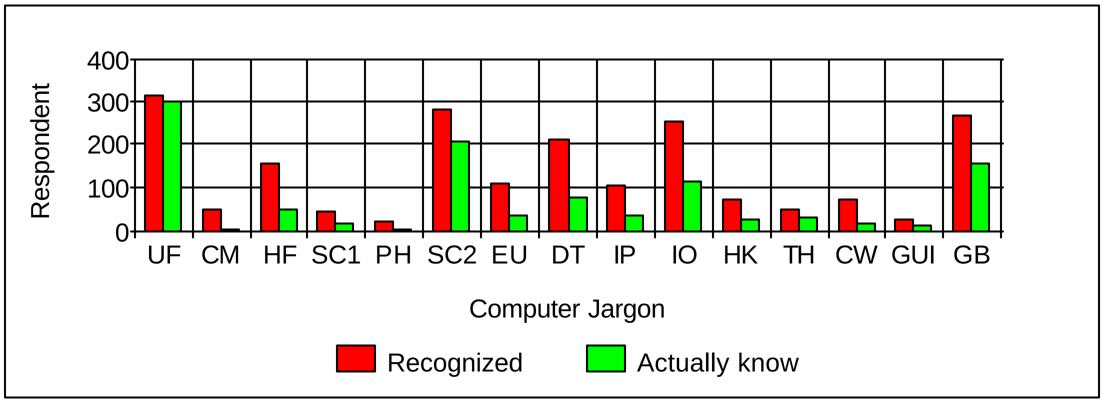
The participants were asked to identify the years of experience they have in using software products such as MS Word, Claris Works, Spreadsheets, etc. Twenty-seven out of 324 participants replied that they had experience of one year or less in using software products. Forty participants replied that they had two to three years of experience in using software products. One hundred nine participants answered that they had four to six years of experience in using software products. Finally, 148 participants replied that they had seven or more than seven years of experience in using software products. In other words 79.3% of the participants had more than four years of experience in using software products.

This survey asked participants to state whether or not they own personal computers. Among 324 participants of this survey, 234 participants answered that they owned a personal computer. This number represents 72.2% of participants of this survey.

This survey asked the students representing the customer groups to circle the number next to the word with which they were familiar. Among the 324 participants, 316 respondents replied that they could identify the term *user-friendly*. Also, 280 respondents recognized the term *system crash*. Two hundred sixty eight respondents recognized the term *gigabyte.* The terms that a majority of respondents had trouble recognizing were *program hangs, GUI, source code*, and *crunch mode*.

On average male respondents recognized 7.4 out of the15 terms this survey provided. Also, on average female respondents recognized 5.8 out of the15 terms this survey provided. Among the 7.4 terms male respondents recognized, they correctly matched 4.4 terms with their definitions. Thus, male respondents correctly matched 58.8% of the terms that they recognized. Female respondents matched 3.1 of the terms they recognized with their definitions. Thus, female respondents correctly matched 53.3% of the terms that they recognized.

Among 316 respondents who answered that they recognized the term *user-friendly*, 301 students matched the term with its definition. This number implies that 95.3% of respondents who recognized the term actually knew the correct meaning of the term. Two hundred five out of 280 respondents who recognized *system crash* matched the term with its definition. This number implies that 73.2% of respondents who recognized the term actually knew its correct meaning. One hundred fifty seven out of 268 respondents who recognized *gigabyte* actually knew the meaning of the term. Figure 5 represents the disparity between the number of respondents who recognized the term and the number of respondents who matched the term with its correct definition.



Note.	UF: User-friendly	CM: Crunch mode	HF: Hidden file
	SC1: Source code	PH: Program hangs	SC2: System crash
	EU: End-user	DT: Downtime	IP: IP Address
	IO: Illegal operation	HK: Hot key	TH: Trojan Horse
	CW: Computer worm	GUI: GUI	GB: Gigabyte

Figure 5: Disparity between recognition and understanding.

Two implications can be drawn from Figure 5. First of all, it implies that recognizing the terms does not mean that the respondents actually knew them. Second, the software engineer group reported in the previous section misjudged the ability of their customers to understand computer jargon.

As discussed in the report on the software engineer group, *system crash (SC2)* was rejected by the students enrolled in the Software Engineering class as being unsuitable for use in their communication with customers. Surprisingly, this term was one of the most widely understood by the customer group. Also, the software engineer group replied that they would not hesitate to use *end-user (EU)* in communication with their customers. However, as Figure 5 shows, this term was poorly understood by the customer group.

The validity of the survey results

This paper ran two z tests for analyzing the validity of the survey results. The first z test was designed to test the hypothesis that the software engineer group possessed more knowledge in understanding computer jargon than customer group. The second z test was designed to test the claims made by various researchers that the software engineer group misjudged their customers' ability to understand computer jargon. In other words, this test is designed to evaluate the claim that software engineers often overestimate their customers' ability to understand their language.

In the first test, P_1 represents the proportion of the software engineer group who provided a correct answer when matching a term with its definition. P_2 represents the proportion of the customer group who correctly matched a term with its definition. The hypothesis for the first z test was $P_1 > P_2$. In other words, the hypothesis states that in every term this survey provided, the software engineer group would perform better than the customer group when matching terms with their definitions. The null hypothesis of this test, therefore, would be $P_1 \leq P_2$.

Since this test ran in the form of a simultaneous comparison (i.e., comparisons as a whole), this paper followed the *Bonferroni Adjustment* to choose the significance level. The calculation of the significance level was 0.0333 (as a result of 0.5/15). This significance level was compared to each jargon's P-value. If the P-value was less than this significance level, the null hypothesis (H_0) was considered to be insignificant and rejected. Otherwise, the null hypothesis (H_0) was considered to be significant and could not be overlooked.

The results of the first test showed that the P-values of *user-friendly, hidden file,* and *system crash* were greater than the significance level. In other words, no significant knowledge gap exists between the software engineer group and the customer group in defining these three terms. However, the results for the other 12 terms support the idea that a knowledge gap exists between the software engineer group and the customer group.

In the second test, P_1 represents the proportion of the software engineer group who replied that they would use a particular jargon in communication with their customers. The P_2 represents the proportion of the customer group who provided a correct answer to the definition for each of the jargon.

The hypothesis for the second z test was $P_1 > P_2$. In other words, the hypothesis states that in every term this survey provided, the software engineer group overestimated the customer group's ability to comprehend computer jargon. The null hypothesis of this test, therefore, would be $P_1 \leq P_2$. The significance level for this test was also 0.0333 using the *Bonferroni Adjustment*. Any P-value less than 0.0333 implies that it is safe to reject the null hypothesis. However, if the P-value is greater than this significance level, the null hypothesis should not be overlooked. Table 5 presents the summary of z-value and P-values of each computer jargon.

In this test, the P-values of *user-friendly* and *system crash* were greater than the significance level. This indicates that software engineer group did not overestimate their customers' ability to understand these terms. However, the result for the other 13 terms supports the claim that the software engineer group overestimated their customers' ability to understand computer jargon.

Implications and findings of the study

The results of the experiment present several interesting points. First of all, as anticipated, the software engineer group was more aware of computer jargon than the customer group. The software engineer group correctly defined 10 to 11 out of 15 terms. The customer group, on the other hand, correctly defined 4 to 5 out of 15 terms. The test comparing these two proportions also supports this conclusion. As discussed, the software engineer group performed better than the customer group in 12 terms.

Second, the software engineer group showed a tendency to overestimate their customers' ability to understand computer jargon. The software engineer group overestimated their customers' ability to understand 13 of the terms. Related to this subject, an interesting point was observed. While the software engineer group overestimated their customers' ability to understand computer jargon in 13 out of 15 computer terms, they underestimated their customers' ability to understand the term *system crash*. This is the reason why the z score in this term was a negative number.

Tomlinson stated that his reason for using computer jargon was to save time. Also, various technicians claimed that their reason for using computer jargon was to provide technical accuracy. Interestingly, the software engineer group that participated in this survey did not share these reasons. As discussed, the software engineer group that participated in this survey stated that the main reason they used computer jargon was their assumption that their customers would understand the terms. Also, the second most favorable reason for using computer jargon was they wanted to present an image as an expert in their domain. This reasoning rather supports Wallraff's and Lutz's claims that jargon functions as a tool for impression rather than expression.

As discussed above, software engineers responded that the reason for using computer jargon was because they assumed that their customer would understand the terms they used. If the software engineer group's assumption is correct then the customer group would define the terms such as *user-friendly*, *end-user,* and *gigabyte* without difficulty since these three terms are highly ranked in terms of software engineers' assumption regarding their customers' level of understanding. However, the analysis of the customer group's response revealed that the term *end-user* was poorly understood.

An interesting pattern was found in comparisons between males and females in the software engineer group. As discussed, female software engineers were more likely to hesitate to use computer jargon than were male software engineers. In turning to the field of gender discourse, various researchers argue that women and men discourse differently. For instance, men tend to show their dominance in their talk while women tend to be more considerate of feelings of others [30]. A similar observation is found in this experiment as well. Male respondents' willingness to use computer jargon in communication with their potential customers can be interpreted as their desire to show dominance over the topic. The hesitance among female respondents regarding the use of jargon can be interpreted as their effort to make their customers feel less intimidated.

Comparisons between males and females in the software engineer group reveal another interesting pattern. A majority of male software engineers pinpointed that the customers' lack of knowledge was their primary reason for not using jargon. However, female software engineers chose technicality of the term and their own lack of knowledge to be their reasons for not using the terms. The field of gender discourse classifies this tendency of differences existing in male and female discourse as the attribution theory

[31]. According to the attribution theory, men are tend to find a source of blame from the outer causes while women are tend to focus on their inner causes. A similar observation is found in this experiment as well.

Conclusion

Various researchers have argued that the most single difficult and important part in building software products is getting correct software requirements. Good communication between software engineers and customers (or users) seems to be the key to success for acquiring a good software requirement. Therefore, excessive use of computer jargon by software engineers in communication with customers cannot be justified.

Various jargon studies suggest that the use of jargon in inter-group communication should be avoided. The study reported by this paper supports the various researchers' claims that software engineers overestimate their customers' ability to understand jargon. Also, the study supported the premise that a knowledge gap exists regarding the awareness of computer jargon between these two groups.

This study also found that the reasons most often given by the software engineers for using jargon were that their customers would understand the jargon and that they wanted to demonstrate knowledge of their field. Neither reason can be considered acceptable. In the case of the first reason, the assumptions of the respondents in this survey were incorrect. The second reason given simply serves no valid purpose when communicating with customers. Instead of being impressed, customers will likely be turned off by excessive use of jargon.

A suggestion for future research in this field is in order. An analysis of the responses from male and female software engineers reveals that male respondents are more likely to use computer jargon than their female counterparts. This observation needs future attention.

The Greek philosopher Aristotle once wrote "The least initial deviation from the truth is multiplied later a thousandfold" [32]. Sixteen centuries later Saint Thomas Aquinas stated that "little errors in the beginning lead to serious consequence in the end" [33]. Software engineers would be wise to heed the sage advice of these two men. The elimination of jargon in communication with customers will be a solid first step toward eliminating errors in software design.

References

1. Glass, R. L. (1998). Software runaways. Upper Saddle River, NJ: Prentice Hall PTR.
   Leveson, N. (1995). Software system safety and computers. Reading, MA: Addison-Wesley Publishing Company, Inc.
   Minasi, M. (2000). The software conspiracy. New York: McGraw-Hill Companies, Inc.
2. Standish Group, The (1994). Charting the seas of information technology. Dennis, MA: The Standish Group International, Inc.
   Standish Group, The (1995). The CHAOS report. Dennis, MA: The Standish Group International, Inc.
3. Leffingwell, D. (1997). Calculating the return on investment from more effective requirements management. American Programmer 10(4), 13-16.
4. Glass, R. L.
5. Minasi, M.
6. Weiss, E. H. (1991). How to write usable user documentation. Phoenix, AZ: The Oryx Press.
7. Wiegers, K. E. (1999). Software requirements. Redmond, WA: Microsoft Press, p. 28.
8. Hartman. J. (1999). jjj: Leverage Core Competencies [On-line]. Available: http://www.kith.org/logos/words/lower3/jjjargon.html
9. Lutz, W. (1989). Double speak. New York: Blonde Bear, Inc.
10. Wallraff, B. (2000). Word court. New York: Harcourt, Inc.
11. Lutz, W.
12. Wallraff, B.
13. Tomlinson, C. (1996). The Tomlison view, Birmingham Post [On-line]. Available: http://www.webxpress.co.uk/ttv/arts/96.09.17-jargon.htm
14. Douglas, J., & Douglas, C. (1994). The language of quality: Unnecessary jargon or tool for change? Positive Impact Times [On-line]. Available: http://www.positiveimpact.com/articles/unnecessary_jargon.html
15. Lutz, W., p. 4.
16. Douglas, J., & Douglas, C.
17. Sandberg, J. (1998). The perils of geek speak. Newsweek [On-line]. Available: http://www.newsweek.com/nw-srv/issue/25_98b/printed/us/st/mc0125_1.htm
18. Lutz, W.
19. Waller, R. (2000). Jargon abbreviations, acronyms, tab index, and access key [On-line]. Available: http://www.waller.co.uk/acronyms.htm
20. Minasi, M.
21. Minasi, M., Tomlinson, C., Wallraff, B., Wiegers, K. E.
22. Minasi, M.
23. Hartman. J.
24. Coates, J. (1997). New book examines computer jargon. Chicago Tribune [On-line]. Available: http://augustachronicle.com/stories/061297/tech_binary.html
25. Wardhaugh, R. (1996). An introduction to sociolinguistics (2nd ed.). Cambridge, MA:Blackwell Publishers Ltd.

26. Coates, J.
27. Freedman, A. (1989). The computer glossary (4th ed.). Point Pleasant, PA: The Computer Language Company, Inc.
28. Downing, D. A., Covington, M. A., & Covington, M. M. (Eds.). (1998). Dictionary of computer and Internet terms (6th ed.). New York: Barron's Educational Series, Inc.
29. Parsons, J. J. & Oja, D. (1998). Computer concepts (3rd ed.). Cambridge, MA: Course Technology.
30. Holmes, J. (1984). 'Women's language: A functional approach. General Linguistics24(3), 149-178.
31. ibid.
32. Adler, M. J. (1985). Ten philosophical mistakes. New York: Simon & Schuster Inc. p. xiii.
33. ibid.