What can OSS mailing lists tell us?
A preliminary psychometric text analysis of the Apache developer mailing list

Peter C. Rigby *
Software Engineering Group
University of Victoria, B.C., Canada
pcr@uvic.ca

Ahmed E. Hassan
Dept. of Electrical and Computer Engineering
University of Victoria, B.C., Canada
ahmed@ece.uvic.ca

Abstract

Developer mailing lists are a rich source of information about Open Source Software (OSS) development. The unstructured nature of email makes extracting information difficult. We use a psychometrically-based linguistic analysis tool, the LIWC, to examine the Apache httpd server developer mailing list. We conduct three preliminary experiments to assess the appropriateness of this tool for information extraction from mailing lists. First, using LIWC dimensions that are correlated with the big five personality traits, we assess the personality of four top developers against a baseline for the entire mailing list. The two developers that were responsible for the major Apache releases had similar personalities. Their personalities were different from the baseline and the other developers. Second, the first and last 50 emails for two top developers who have left the project are examined. The analysis shows promise in understanding why developers join and leave a project. Third, we examine word usage on the mailing list for two major Apache releases. The differences may reflect the relative success of each release.

1 Introduction

Compared to most development artifacts, such as source code or bug reports, mailing lists are less structured allowing discussion of a wider range of topics. These lists embed information about the Open Source Software (OSS) development process, design decisions, and developer characteristics. While flexibility is important during development, it complicates the mining of useful information from mailing lists.

Text analysis tools have been used on a variety of artifacts to understand and predict aspects of the development processes. For example, Mockus and Votta [10] used text analysis on CVS logs to characterize changes to the system, such as corrective and perfective changes. More recently Li et al. [6] combined manual classification, text analysis, and machine learning to classify bug databases. Although these techniques could be applied to the analysis of messages on mailing lists, the range of topics and ambiguity of the discussion on a mailing list is larger than in a bug database or CVS commit log. This ambiguity makes the classification step more difficult. Instead of creating our own dictionary or extracting one from the large corpus of emails, we use a context independent, text analysis tool from psychology to examine a mailing list. This limits our research to the psychological and social aspects of the mailing list, but affords us greater external validity.

Our primary goal is to assess the usefulness of the Linguistic Inquiry and Word Count (LIWC) tool as a predictor and classifier as well as a tool to understand the intricacies of OSS development. We conduct three distinct, preliminary experiments on the Apache httpd server's developer mailing list. In the next subsection, we discuss the motivation and rationale for each experiment.

1.1 Overview of our Experiments

What is the personality type of OSS developers? Raymond [16] describes how modesty and fully acknowledging contributions from others are essential traits of the founders of Perl and Linux. We are unaware of any research that empirically examines if there is a particular personality type that is successful as an OSS developer. We build on the efforts of others who correlated word counts with the big five personality traits, a standard measure of personality [12], to assess the personality of core OSS developers. Conscientiousness is one of the big five personality traits. Are core Apache developers more diligent than the general mailing list population?

Does the language and attitude of a developer change as he or she moves from being a new, to a current, to a departing developer?

*Rigby acknowledges the support of a NSERC CGSD scholarship

29th International Conference on Software Engineering Workshops(ICSEW'07)
0-7695-2830-9/07 $20.00 © 2007 IEEE

There has been a great deal of research on why and how developers join OSS projects (e.g., [3, 4]). We are interested in assessing the word usage of not only new developers, but also more experienced developers. Since word usage can measure sociability, we use it to identify reasons why a developer left a project. Are new developers more tentative, are current developers more certain, and are departing developers frustrated or angry?

Does word usage change around the time of a release? Are there differences in word usage before and after a release? Can changes in word usage predict an upcoming release? For example, is the language more optimistic with more future tense verbs before a release? After a release, is language less optimistic with more past tense verbs?

1.2 Organization of the Paper

This paper is organized as follows. First, we introduce the LIWC tool and its associated dictionary. Second, we discuss our methodology and data. Third, we present the results of our experiments. For each experiment, we discuss relevant background and literature, present and discuss our results, and conclude with suggestions for future work and a discussion of the potential applicability of our technique. Finally, we provide a general conclusion and limitations of our study.

2 Linguistic Inquiry and Word Count

Do word counts have any value in understanding the psychology of the individual who wrote the text?

The LIWC is based on counting function words or particles. The number of particles in English is small, and it is very difficult for an individual to change the function words that he or she uses [13]. By simply counting words, the meaning of the text is lost. Word count programs cannot catch sarcasm, irony, or a given contextual meaning for words with many meanings. The power of a word count tool lies in the dictionary that divides the different counts into meaningful dimensions. The LIWC uses a psychometrically-based dictionary. The dictionary has been validated by independent judges and used in a number of experiments by Pennebaker and others [1]. The LIWC dictionary contains 70 dimensions and over 2300 words and word stems [2]. The following quotes illustrates how the LIWC works.

1. `I sent these patches in a couple weeks ago. They may be of interest, maybe not.`

2. `This patch fixes the important half of the bug.`

In the first quotation, it appears that the author is uncertain as to the value of his contribution. In the second quotation, the author seems confident. Words like ‘maybe’ are associated with tentativeness and words like ‘important’ are associated with certainty. The LIWC tool simply counts words that are contained in its dictionary. The dictionary is grouped into basic linguistic and psychometric dimensions with each word belonging to one or more dimension. Although some dimensions are context dependent (e.g., sports and sex), the dimensions used in this paper are context independent. For example, the two previous quotations were about patches. Since the words ‘patch’ and ‘bug’ are not in the dictionary, however, the result would have been the same if we were talking about a proposal to fix a building, instead of a software patch.

There are three top level categories used in our study: language composition, psychological processes, and relativity. Table 1 highlights the LIWC dimensions used in our study. We discuss the categories and their associated dimensions below.

1. *Language composition* consists of the standard word count dimensions such as pronouns, total words, and articles. These simple measures have been useful in determining, for example, if an individual is lying. Compared to non-deceptive texts, texts that contain deception have lower instances of first person pronouns as the author tries to distance himself or herself from the text [11].

2. *Psychological processes* is subdivided into emotional, cognitive and social processes. Unlike the standard linguistic dimensions which have an unambiguous definition, these dimensions have been created through examination of hundreds of texts from distinct areas as well as voting on inclusion of words by independent judges [13]. These dimensions can help determine if the author is optimistic or anxious, insightful or inhibited, and social or anti-social.

3. *Relativity* is divided into temporal references and past tense verbs. The former indicates that time was discussed (e.g., two weeks ago), while the latter indicates whether the past, present, or future is under discussion.

3 Methodology and Data Source

In order to answer our research questions we need mailing list data. We choose to study the Apache httpd server

[1] See `http://homepage.psy.utexas.edu/homepage/faculty/pennebaker/reprints/` for the dimensions contained in the LIWC dictionary

[2] The complete table describing all 70 dimensions is available at `http://www.liwc.net/descriptiontable1.php` December 2006.

Dimension	Example Words	Explanation
Pronouns	'our', 'I'	Subject or object in a sentence.
Negation of other words	'no', 'not'	Words that negate other words.
Insight	'think', 'know'	Words that indicate understanding about a topic.
Inhibition	'block', 'constrain'	Words that show the individual is restrained.
Tentative	'maybe', 'guess'	Words that indicate that the speaker is uncertain.
Causation	'because', 'effect'	Words that indicate why something happened
Social Processes	'talk', 'us'	Words that are related to social interaction.
Positive and Negative Emotion	'good', 'hate'	Words related to the emotion of the speaker.
Optimism and energy	'pride', 'win'	Subdimension of positive emotion.
Past, Present, and Future	'walked', 'walk', 'will'	The tense of the verb.
Reference to Time	'hour', 'day'	Words related to time.
Inclusive and Exclusive	'with', 'except'	Words that include or exclude others and their ideas.

Table 1. Selected LIWC dimensions

project because it is a large, successful, mature OSS project that has been extensively studied by other researchers (e.g., [1, 9]). A measure of success is market share. Apache has over 60% of the server market with the large majority of Apache servers still running version 1.3, which was first released in 1998 [17]. Apache developers are usually volunteers with other time commitments who rarely meet in person, so almost all project communication is recorded. The most important forum for developer communication is the developers' mailing list. We extracted the body of all email messages on this mailing list between 1995 and 2005 – approximately 104,000 emails.

In the first two experiments we study the characteristics of individual developers. Since we are interested in "successful" or "leading" developers, we select four developers who each had the most commits for a given two year period. For example, developer B was the top committer for 1999 and 2000, while developer C was the top committer for 2001 and 2002. For comparison purposes, we also group emails from multiple developers based on a particular characteristic. For example, we group all individuals who have sent less than 30 messages during the 10 year period we studied, thus creating a group that represents infrequent contributors. Although we could have used questionnaires to answer these questions, questionnaires are often biased by perceptions of past events, by the questions that are asked, and by the individuals that respond.

We are only interested in English text that was written by the individual who is sending the message. We removed all diffs, attachments, and quoted replies (i.e., lines beginning '$>$'). We did not remove email signatures and code and HTML that was not part of a diff. In future, we would like to remove email signatures by diffing emails from the same individual (which involves resolving an individual's potentially many email addresses) and removing text that crossed a threshold number of emails. To reduce the effect of improper parsing, we calculate each dimension as a *percentage* of the number of words found in the LIWC dictionary per email, instead of the standard percentage of total number of words per email. This technique excludes code and HTML, which is unlikely to be contained in the LIWC dictionary. For example, '$< br >$' is not contained in the LIWC dictionary, but will be counted as a word.

The LIWC tool takes an input text file and returns a file containing the LIWC dimensions related to the text file. The tool does not run from the command line, so instead of re-running it to answer each research question, we ran it once and stored the output in a database. The database was queried for each research question. This query created a file that could be analysed by either Weka, a machine learning tool, or SPSS, a statistical package.

3.1 Analysis Techniques

In this subsection we present the analysis techniques used in our study. **Decision Trees.** The C4.5 Decision Tree developed by Quinlan builds a tree containing decision and leaf nodes [15]. C4.5 inductively builds a tree where each non-leaf node is a decision point for a specific attribute and each leaf node is an outcome prediction. C4.5 builds on several previous decision tree algorithms by Quinlan adding features such as allowing continuous and missing data. Each path through the tree is a predictive rule. C4.5 prunes as it builds and drops uninteresting attributes. We use the Weka implementation of the C4.5 algorithm, called J48. For example, a decision tree might create two branches, pre- and post-release. In the pre-release branch we might have a large number of emails with a high percentage of optimistic words.

Standard Statistical Tests. The t-test compares the means of two independent samples. An ANOVA (analysis of variance) compares the means of three or more inde-

Subject	Neuro	Extro	Open	Agree	Consc
Num. of Clusters	1	3	3	2	3
A	8.10^1	$2.15^{1,2}$	22.47^1	-9.20^2	$-4.07^{2,3}$
B	8.79^1	2.61^2	19.30^1	-10.46^1	-4.24^2
C	7.35^1	5.95^3	32.84^2	-10.58^1	-4.34^2
D	7.86^1	0.53^1	44.18^3	-11.12^1	-5.36^1
≤ 30	7.07^1	6.41^3	32.87^2	-11.15^1	-3.42^3
> 30	7.33^1	7.61^3	32.85^2	-10.87^1	-4.24^2

Table 2. Composite measure of Personality. Values are relative, not absolute. Post Hoc Tukey's HSD Test ($\alpha = 0.05$). Superscript represents Tukey cluster number.

pendent samples. Tukey's Honestly Significant Difference (HSD) post-hoc comparison test is run following a statistically significant ANOVA. It compares each sample to every other sample, indicating which samples differ from each other at statistically significant levels. It also groups the samples into clusters of samples that do not differ significantly within clusters but differ significantly between clusters. For example, in the personality experiment developers are clustered based on their personality characteristics.

4 Personality

What is the personality type of OSS developers?

Background and Related Work. Mairesse and Walker [7] use the LIWC dictionary, the MRC Psycholinguistic Database, and Weka to create a personality recognizer. The tool uses two corpora to create Weka models that can then be applied to predict the personalities of unseen subjects. These models are based on essays written by and spoken text of psychology students [8, 14]. We applied the Weka models to the Apache project's developer mailing list. The results illustrate the inflexibility of the models. For example, all developers appeared to be extremely extroverted because the size of the mailing list text was substantially larger than the text in the psychology student sample. Further examination of decision trees created by the tool revealed that attributes that are unlikely to be discussed on a developers' mailing list, such as sports, eating, and sex, were used as predictors. It is possible to create new models for Mairesse's tool; however, to create these models we would need subjects to complete a standardized big five personality questionnaire. It is unlikely that a representative sample of OSS developers would complete such an involved personality test.

Pennebaker and King [14] found that certain LIWC dimensions were correlated to the big five personality measures [12]. Although the effect size was small, the results were statistically significant. A short description of each of the big five personality factors as well as the LIWC variables with which they are correlated follows.

- *Neuroticism* the tendency to express negative emotions such as anger. Neuroticism was correlated with Articles (.13), Positive Emotion (-.13), and Negative Emotion (.16).
- *Extroversion* the tendency to seek the company of others. Extroversion was correlated with Tentativeness (-.14), Negations (-.12), Social (.12), and Positive Emotion (.15).
- *Openness* the ability to understand and respect unusual ideas. Openness was correlated with First-Person Singular words (-.13), Articles (.13), Words of more than 6 letters (.16), and Present tense verbs (-.15).
- *Agreeableness* the tendency to be cooperative rather than antagonistic. Agreeableness was correlated only with Articles (-.15).
- *Conscientiousness* the tendency to be self-disciplined and diligent. Conscientiousness was correlated with Negations (-.15) and Negative emotions (-.15).

Experiment. We hypothesize that top committers will have a similar personality type to each other, and that the personality of top committers will differ from that of the entire mailing list in the following ways: top committers will be less neurotic, more extroverted, more open, more agreeable, and more conscientious than the general mailing list. For example, we feel that successful leaders will be more open to others' ideas than the general population.

For each personality trait, we created a composite measure by adding or subtracting the independent, correlated LIWC dimensions. For example, neuroticism is articles $-$ positive emotion $+$ negative emotion. The correlations between the word types and the personality characteristic are of similar sizes (.13 to .16), so we did not multiply the word counts by the values of the correlations. We first create a baseline by assessing the LIWC dimensions for individuals who make few posts to the mailing list (i.e., between 1 and 30 messages) as well as for the remainder of the mailing list (i.e., 31 or more messages), excluding the top four committers. We then calculate the LIWC dimensions for the top four committers and compare them to the baselines. These baselines allow us to assess whether top committers have a different personality type from the general trend on the entire mailing list. We further compare the top committers to each other. We refer to the two baselines and the developers as subjects.

Results and Discussion. Initial results were disappointing, as there were large discrepancies in samples sizes (i.e.,

29th International Conference on Software Engineering Workshops(ICSEW'07)
0-7695-2830-9/07 $20.00 © 2007 IEEE

number of emails) among the subjects. These discrepancies caused Weka decision trees to ignore the subjects with few emails and meant that the level of statistical significance for Tukey's HSD post-hoc comparison test was not guaranteed. Using a standard data mining technique, we created similar sample sizes by randomly selecting 500 messages from each subject [18]. To cluster the subjects we ran an ANOVA with a Tukey's HSD post-hoc comparison test. The composite personality measure is presented in table 2. Since it is a composite measure, the absolute values are meaningless, but comparing the relative levels among the subjects allows us to determine personality differences. The superscripts in table 2 indicates for each personality trait which cluster the subject belongs to. Each cluster varies at statistically significant levels from the other clusters. If there is only one cluster, there is no statistically significant difference among the subjects. Subjects with higher cluster numbers exhibit high levels of a trait. For example, developer C is very extroverted, while developer D is not. We now discuss the results in terms of the personality traits and subjects.

Personality traits. For *neuroticism*, there is no statistically significant differences among the subjects. Since the developers do not vary significantly from the general mailing list population, it is likely that the top committers are not emotionally unstable individuals. The least *extroverted* developer is developer D. Although Developer D's score appears relatively low (0.53) when compared to Developer A (2.15), there is no statistical difference. Developer A and B have intermediate levels of extroversion. Developer C and the general mailing list population are the most extroverted. It appears that the top committers are less extroverted than the general mailing list population. Developer D has the highest *openness* score. While Developer A and B have the lowest scores. Developer C matches the general mailing list population. No obvious pattern emerges from the openness measure. Developer A is the most *agreeable* developer. There is no difference among the other developers and the general population. *Conscientiousness* is the only measure in which the two general mailing list subjects differ. Individuals who have sent less than 30 messages appear to be the most conscientious. This result is counter to our hypothesis. One possible explanation is that individuals who send few messages are intimidated by the regular project members and are very conscientious about what they write. Developer A belongs to both the highest and intermediate clusters. While Developers B, C, and the general mailing list population belong to the intermediate level. Developer D appears to be the least conscientious.

Subjects. With the exception of agreeableness developer A and B belonged to the same clusters for all the measures. These developers were the top developers during the development of Apache 1.3 and 2.0 respectively. Developer C and the general mailing list population (> 30 messages) were the same for all measures. Developer D was not consistently associated with any other subject. Comparing developers A and B to developer D and the general mailing list, we see that developers A and B were less extroverted, less open, but equally conscientious.

Future Work. Pennebaker and King [14] report only small correlations between LIWC dimensions and the big five personality traits. The correlations also come from sample essays in which the individual does not receive a response. These two limitations may limit the validity of our results. However, this experiment is a first attempt at understanding the personality type of OSS developers. Unfortunately, there is no existing standard against which we can compare our findings; Mairesse's benchmark is context dependent. The benchmark used in this study, the general mailing list population, appears to be useful. In the future, however, it will be necessary to compare our benchmark with other benchmarks, in order to determine similarity. For example, the top committers did not differ significantly from the general population on neuroticism – perhaps all OSS developers are high on neuroticism. In order to validate our benchmark, we will need to test it against a sample with known scores on these personality traits.

Developers A and B, who were responsible for the two significant Apache releases, are similar to each other but differ from the general mailing list population in that they are less extroverted and less open. Given that the Apache project is recognized as being very conservative in the contributions that it accepts, the lack of openness and extroversion may not be unexpected [9]. In future, we will test the hypothesis that less extroverted and less open developers are more effective at producing major releases.

The lack of consistency among developers and the similarity of developer C and the general mailing list requires us to reject our original hypothesis. The two major Apache releases were times of significant change, the top committers during this time had different personalities from the rest general mailing list. We hypothesize that significant changes to the project occur when the leading developers differ from the rest of the population.

We feel that some notion of role and project state must be incorporated into future experiments. For example, a project may be mostly performing maintenance or altering the API. We hypothesis that different project states and roles attract developers with different personalities, but within each state or role the personalities should be similar.

5 New, Current, Departing Developers

Does the language and attitude of a developer change as he or she moves from being a new, to a current, to a departing developer?

Background and Related work. There has been a great deal of research on why and how developers join OSS projects. For example, Ducheneaut [3] uses a modified version of ethnography to follow the socialization process of new developers in OSS projects. Herraiz et al. [4] follow the integration of new professional and volunteer developers. Using the LIWC we hope to examine, in an automated fashion, how developers change throughout their time with the project.

Experiment. Ideally, for this experiment, one would have individuals and not email addresses, as one individual may have multiple email addresses [1]. However, for simplicity's sake, in this initial experiment we examine the top committers for which we have resolved email addresses. For the two top committers who have left the project we compare the word usage of their first and last 50 emails. We run a t-test for each developer comparing their starting emails to their departing emails. In our second approach, we partition the four top committers' emails into equal sized groups. We use decision trees as well as an ANOVA to determine which groups have similar word usage. We want to see if developers' word usage shifts over time.

Results and Discussion. Table 3 presents the results of a t-test comparing the first and last 50 emails for the two top committers who have left the project (Developers A and B from the previous experiment). The interpretations that follow are not the only possible interpretation, but are supported by a preliminary qualitative analysis. Unlike the previous experiment where a baseline can be created to validate the results, validation in this experiment can only be obtained through comparison to existing literature and qualitative analyses.

Despite making few commits in his last three years, developer A continued to post to the mailing list. From the large increase in past tense verbs it is likely that developer A would answer questions about things he had done in the past. Further evidence is the increase of negations (e.g., 'no', 'never'), which could indicate a person telling someone else how something was not done. He was more sociable and used more pronouns when he first began.

Once developer B stopped committing, he also stopped posting to the mailing list. Developer B appears to have shifted from the use of pronouns that included himself, to more instructive 'you' pronouns. Discussion of time in general and the present was higher at first, with discussion of the future increasing before he left. There was a decrease in the number of positive emotions. Words related to insight decreased as he was leaving, while causal words and exclusive words increased. This may be indicative of a change where developer B moved from producing new insights to explaining the cause or arguing why a particular solution will not work with the way the code is structured. The decrease in positive emotions, increase in instruction, potential shift from creator to critic, and increased discussion of the future is indicative of someone whose role has changed, but as we have seen, will be leaving the project shortly.

Dimension	Subdimension	Dev A	Dev B
Standard WC Pronouns	Total Pronouns	-3.4	-1.9
	We	-1.0	-1.7
	Self	*	-2.6
	You	*	1.8
	Other people	-2.3	*
Standard WC	Negation	1.1	2.3
Emotion	Positive	*	-2.2
Cognitive	Cause	*	1.0
	Insight	*	-1.4
	Inhibit	*	-0.4
Social Processes		-3.23	*
Relative	Time	*	-2.4
	Present	*	-2.0
	Past	3.4	*
	Future	*	1.0
	Inclusive	-3.0	*
	Exclusive	*	1.9

Table 3. Mean differences for the first and last 50 email messages. (* $p > 0.05$, otherwise $p \leq 0.05$)

Dividing all the emails for each individual developers into multiple groups was problematic. With few groups, less than ten, the beginning and end merged with the middle. When we used many groups, ten or more, it became difficult to interpret the results. The clustering of groups created by Tukey's HSD tests were not obvious (e.g., the groups were not ordered by time). Since we can see a definite distinction between the beginning and end results, the Tukey cluster result indicates that the groups overlapped the "true" divisions. Similar results were obtained by using decision trees in Weka. The percentage of correctly classified groups decreased to marginally better than chance as the number of groups increased to ten. Although the developers' word usage may change over time, the groupings we used were likely too rough to properly examine this change.

Future work. Event-based groups, such as Cohn et al.'s study of Americans before and after the September 11th, 2001 terrorist attacks, may improve the results [2]. Other event-based divisions may be obtained through manual analysis. However, determining events manually is time-consuming. One potential automated technique is to divide a developer's emails based on the frequency with which he or she is sending messages; each change in frequency above a given threshold would result in a new group.

In future work, we would like to determine the difference between a successful newcomer and a newcomer who does not receive commit privileges. This analysis would involve comparing the first messages for the two types of developers. The period would end when the developer received commit privileges or stopped submitting patches. If a simple set of LIWC dimensions can accurately distinguish between the two types of developers, then, it would be possible to integrate productive members more quickly and avoid time wasted on integrating unproductive developers.

6 Releases

Does word usage change around the time of a release?

Background and Related Work. We are interested in discovering if there is a consistent change in word usage before and after a release. For example, Raymond [16] noticed that developers are often optimistic and shocked that a release, which they see as "perfect," contains many defects. Are developers more optimistic before a release?

Experiment. We examine the Apache 1.3 and 2.0 releases (two major releases) by categorizing emails as occurring either a month before or after a release. We run t-tests to determine if there are any statistically significant differences in word usage before and after a release.

Results and Discussion. From table 4 we can see that discussion before the Apache 1.3 release was more optimistic, more tentative, used more future tense verbs, and was less about actual times than after the release. Some of these results make intuitive sense: developers are often optimistic before a release and then less optimistic when the users start reporting bugs. Do these findings hold for other releases?

Examining the Apache 2.0 release, we see that after the release there were less social and less inclusive words than before the release. Although there is no overlap in the LIWC dimensions that are statistically significant between the two releases, it is apparent that the atmosphere surrounding each of the releases was different. For example, with the 2.0 release, sociability seems to have increased after the release, and there is no statistically significant change in optimism as there was for the 1.3 release. The differences between the two releases suggest that it is unlikely that we will be able to predict when a release will occur.

Future Work. Microsoft developers were encouraged to bet on when a product would be released. Communication between project managers and developers was found to be overly optimistic, and the betting scheme provided a more accurate estimate of the actual release date than the project managers did [5].

Although our original goal was to find consistency in the language used around a the time of a release, we find that this was not possible for the Apache 1.3 and 2.0 releases. However, looking for consistency among releases may be misguided as some releases may be successful, while others may fail. Instead, we feel that the presence or lack of a particular LIWC dimension, before or after a release, may be indicative of the success of that release. Just as the Microsoft developers were accurate in predicting the release date, their attitudes before a release may indicate the release's relative quality. Our new research question is as follows: are there LIWC dimensions that can predict group morale and does group morale predict the success of a release?

Dimension	-1.3	-2.0
Optimism	-0.37	*
Tentative	-1.3	*
References to Time	1.1	*
Future tense verbs	-0.7	*
Social Processes	*	0.74
Inclusive	*	-0.64

Table 4. Mean differences for Apache 1.3 and 2.0 releases. (* $p > 0.05$, otherwise $p \leq 0.05$)

To answer this question we need many release dates for the project and a measure of success for each release. The former can be roughly obtained from announce lists in OSS and from more rigorous records in industry. The success of a release could be measured as the time before the next major release, assuming that longer lasting releases are more successful, or as the number of defects for a given release.

7 Limitations

There are four main limitations to our study. First, The LIWC was created as tool to measure various psychological phenomenon. It has been used in a variety of environments, but to our knowledge, has not be used to understand OSS developers. Second, the LIWC may be biased against individuals who's first language is not English. Third, although the Apache httpd server is a large, successful, mature project, it may not be representative of OSS development in general. Fourth, the three experiments that we performed were preliminary. As such, they had a limited number of developers and project releases. Further experimentation and validation must be performed before any of our results can be generalized.

8 Conclusions

This paper represents a first attempt at using a psychometrically-based word count tool to understand OSS mailing lists. Instead of creating our own tool, we used the LIWC dictionary. We conducted three preliminary experiments on the Apache httpd server's developer mailing list and proposed directions for future research. The general conclusions from each experiment are as follows.

Personality. Using the LIWC dimensions that are correlated with the big five personality traits, we created a baseline personality score for the entire mailing list (excluding the top four committers) and compared the top four committers to the baseline and to each other. The two developers that were responsible for two major Apache releases had similar personalities. Their personalities were different from the baseline and other developers on the traits of extroversion and openness. We plan to run further comparisons with other projects as well as examine the effect of a developer's role on his or her personality.

New, Current, and Departing Developers. For the two top committers who have left the Apache project, we extracted the LIWC dimensions for their first and last 50 emails. Although future qualitative examination of these emails is required for confirmation of our findings, we feel that the statistically significant LIWC dimensions are consistent with our initial manual analysis of the developers. Dividing developers' emails into equal groups appears to span multiple events and produce statistically non-significant results. Dividing developers' emails into event based periods would likely produce more interesting results. In the future, we would like to develop a measure based on the LIWC dimensions and the characteristics of past successful and unsuccessful developers that would predict whether a newcomer will become a successful member of the development team.

Releases. We calculated the LIWC dimensions one month before and one month after the releases of Apache 1.3 and 2.0. The 1.3 release had higher pre-release levels of optimism. The 2.0 release had higher post-release levels of social words. There were no common statistically significant LIWC dimensions between the two releases. However, we did not take into account the success of each release. Based on the finding at Microsoft that developers can predict more accurately than project managers when a release will occur, we plan to use the LIWC dimensions to assess the "attitude" of developers around successful and failed releases.

We feel that the we have gained some insight into the type of people who participate in and the discussions that occur on the Apache mailing list. We have attempted to determine the personality types of four top developers, to understand why developers join and leave a project, and to examine the general attitude of developers before and after a release. Each experiment presented unique difficulties, but all show some promise for future work.

References

[1] C. Bird, A. Gourley, P. Devanbu, M. Gertz, and A. Swaminathan. Mining email social networks. In *MSR, ICSE '06*, pages 137–143, New York, NY, USA, 2006. ACM Press.

[2] M. Cohn, M. Mehl, and J. Pennebaker. Linguistic markers of psychological change surrounding september 11, 2001. *Psychological Science*, 15(10):687, 2004.

[3] N. Ducheneaut. Socialization in an Open Source Software Community: A Socio-Technical Analysis. *Computer Supported Cooperative Work (CSCW)*, 14(4):323–368, 2005.

[4] I. Herraiz, G. Robles, J. Amor, T. Romera, and J. Barahona. The processes of joining in global distributed software projects. *Proceedings of the 2006 international workshop on Global software development for the practitioner*, pages 27–33, 2006.

[5] P. Kaihla. Best-kept secrets of the world's best companies. *Business 2.0 Magazine*, March 2006.

[6] Z. Li, L. Tan, X. Wang, S. Lu, Y. Zhou, and C. Zhai. Have things changed now? an empirical study of bug characteristics in modern open source software. In *In proceedings of ASID 2006*, pages 1–9. ACM, 2006.

[7] F. Mairesse and M. Walker. Words mark the nerds: Computational models of personality recognition through language. In *Proceedings of the 28th Annual Conference of the Cognitive Science Society (CogSci)*, pages 543–548, Vancouver, 2006.

[8] M. Mehl and J. Pennebaker. The sounds of social life: A psychometric analysis of students daily social environments and natural conversations. *Journal of Personality and Social Psychology*, 84:857–870, 2003.

[9] A. Mockus, R. T. Fielding, and J. Herbsleb. Two case studies of open source software development: Apache and Mozilla. *ACM Transactions on Software Engineering and Methodology*, 11(3):1–38, July 2002.

[10] A. Mockus and L. Votta. Identifying reasons for software changes using historic databases. *International Conference on Software Maintenance*, pages 120–130, 2000.

[11] M. Newman, J. Pennebaker, D. Berry, and J. Richards. Lying Words: Predicting Deception from Linguistic Styles. *Personality and Social Psychology Bulletin*, 29(5):665–675, 2003.

[12] D. Ozer and S. Reise. Personality assessment. *Annual Review of Psychology*, 45, 1994.

[13] J. Pennebaker, M. Francis, and R. Booth. Linguistic inquiry and word count: Liwc 2001. *Mahwah, NJ: Erlbaum*, 2001.

[14] J. Pennebaker and L. King. Linguistic styles: language use as an individual difference. *J Pers Soc Psychol*, 77(6):1296–312, 1999.

[15] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[16] E. S. Raymond. *The Cathedral and the Bazaar*. O'Reilly and Associates, 1999.

[17] J. Slive. Apache 1.3 vs 2.0. `http://www.unixadmintalk.com/f44/apache-1-3-vs-2-0-a-39028/`, 2006.

[18] I. Witten and E. Frank. *Data Mining:: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 1999.

29th International Conference on Software Engineering Workshops(ICSEW'07)
0-7695-2830-9/07 $20.00 © 2007 IEEE