

Scouting Requirements Quality Using Visual Representations

Orlena C.Z. Gotel, Francis T. Marchese

Department of Computer Science, Pace University, New York, USA
{ogotel@pace.edu, fmarchese@pace.edu}

Abstract

Examining the quality of a set of requirements is a sensible project health check given their role in the engineering of quality software systems. However, not all project stakeholders may recognize the value of requirements audits or inspections, and scrutinizing the details of a requirements document can be perceived to be too time-consuming, distracting and costly an activity to undertake early on in a project. We suggest that a major benefit of any such review activity is the discussion that is triggered amongst stakeholders about the artifact under consideration, in this case the requirements document, and that more cursory approaches may yield some of this value and be more appealing so as to encourage this actual practice. In particular, we propose that a visualization of an emerging requirements document could be generated as a vehicle for preliminary review, in advance of more concerted efforts directed towards finding defects in the predominantly text-based artifact itself. We call this initial review activity ‘scouting’, provide heuristics to support it and evaluate the potential of Wordles as a candidate visual representation. This work hence proposes applying a pre-existing visualization technique to an important problem area in software engineering.

Keywords--- Audit, Inspection, Quality, Requirements Document, Requirements Visualization, Review, Scouting, Software Engineering Visualization, Wordle.

1. Introduction

Obtaining a status report on the quality of a set of requirements is a sensible project health check. However, not all project stakeholders may recognize those factors that characterize a ‘quality’ set of requirements, and they may not always appreciate the role of requirements audits or inspections in determining this. Whether the requirements are atomic, traceable, consistent, testable, prioritized, etc. are amongst a number of properties that should be checked to establish the ‘likely quality’ of the written requirements [1, 4, 14], and an even more

important pre-requisite is that those requirements are also confirmed to reflect valid needs [20].

While document reviews are widely regarded as an important quality assurance technique in software development and an industry best practice [3, 8, 17, 24], they are mostly undertaken for code. Referred to as inspections, walkthroughs, desk checks, pass-arounds, pair programming, audits or peer review, they are still somewhat associated with the process as originally outlined by Fagan for inspecting software [5]. Consequently, many practitioners can overlook how critical this practice can also be to improving the quality of software development artifacts other than code. Given that requirements documents are one of the earliest project artifacts in which defects are introduced in a project, these are obviously one of the most cost-effective artifacts to focus review upon [4, 22]. However, since requirements are often volatile early on in a project as an understanding is evolving, there is an obvious cost/benefit issue to consider.

Less formal alternatives to conventional inspection practices have recently been highlighted in the literature [9] and more agile approaches are now emerging [7]. For example, extreme inspection revolves around thirty-to-sixty minute inspections, undertaken continuously by sampling artifacts and applying a few simple initial checks, so as to estimate the defect density in specifications [7].

In this paper, we suggest that project stakeholders would benefit from a way to assess the quality of an evolving requirements document quickly, at a very high-level and with little effort, so that this can be undertaken on a regular and ongoing basis so as to help assure quality. More critically, what is needed is a way to bring a wider range of project stakeholder into this process to promote requirements discussion, especially when reading and reviewing requirements documents is unlikely to be their top priority.

The paper is organized as follows. In Section 2, we discuss some typical requirements quality checks and the barriers to undertaking reviews of requirements documents in practice. In Section 3, we examine those requirements characteristics that could be rendered visible to form an alternative vehicle around which to focus these quality checks. We highlight the use of

Wordles for this purpose given their representation of text, their prevalence and ease of creation. In Section 4, we describe a study that was undertaken to compare the role of Wordles with that of traditional requirements documents for performing quick and effective quality checks, and we summarize our findings in Section 5. We draw some initial conclusions and make recommendations for future work in Section 6.

2. Checking for Requirements Quality

Less has been written about requirements inspections than about software inspections, and auditing guidelines tend to emphasize requirements engineering processes over the requirements products developed. The very words ‘inspect’ and ‘audit’ imply conducting a careful and critical *examination* of material, and so there are many barriers to the use of requirements document inspections and audits in practice. (Though there are subtle differences in intent and conduct, for the purpose of this paper we will use the terms interchangeably.)

We claim that: inspections take time to both plan and undertake, time perceived to deter from the ‘real’ engineering; inspections demand knowledge and expertise, and are best undertaken in conjunction with independent parties, so are costly to exercise; inspections have negative connotations associated with faultfinding and blame allocation that may go counter to an organizational culture; following any change to an inspected artifact, the previous results may be rendered null and void, so artifacts may need regular inspection to maintain relevancy; and people find careful line-by-line and page-by-page textual reviewing work tiresome, so open to error where not conducted with zeal.

When the inspections are of early project artifacts rather than code, the consequences of defaults may not be so apparent and pressing. Consequently, and coupled with the more general barriers, it can be difficult to try to encourage project stakeholders to take the time to undertake any form of requirements inspection whatsoever; success seems inevitable early on in a project and there is often the need to be seen to be ‘moving forward’ doing ‘more interesting’ things. Unfortunately, this is exactly the point in the software development lifecycle where there is the most value to be had in getting diverse stakeholders together to discuss the requirements.

Although requirements checklists form an important component of any quality assurance practice within those organizations focused on software process improvement (e.g., to “*Objectively Evaluate Work Products and Services*” is one of the specific practices of the Process and Product Quality Assurance Process Area of the Capability Maturity Model Integration [3]), their use by other organizations is not always so common. A number of dedicated approaches to help detect defects in requirements exist, such as perspective-based reading and N-fold inspections [15, 19, 20, 21], but these are all still subject to the issues listed above.

The assumption of this paper is that while there are a number of properties a requirements document of reasonable quality would be expected to exhibit, checking for these properties can be costly, consuming too much time and effort, and so a team or an organization may not consider there to be perceivable benefit. Also, the traditional format of the requirements document may not make it so readily accessible to a wide range of stakeholder. A project team needs to select the cheapest approach that will reduce the risk associated with defects remaining in a given deliverable to an acceptable level [24]. We therefore seek to find alternative, less-costly and more engaging approaches.

There is no ‘definitive’ checklist as such for assessing the quality of a requirements document. Nevertheless, and based upon recommended best practices [1, 4, 14], one would expect to see some statement of the problem being tackled, the key business goals and the needs of the major stakeholders. One would expect the requirements to be written in the language of the domain and to be free of design constraints. One would also expect the individual requirements to possess desirable properties, such as identifiers, rationale and priorities. Table 1 consolidates typical high-level questions that could be used to begin to assess the quality of a requirements document.

Table 1. Requirements Quality Questions

- | |
|---|
| <ol style="list-style-type: none"> 1. If you could name the intended software system, what would you call it? 2. Who are the main stakeholders for the system? 3. What are the main functional requirements of the system? 4. What categories of non-functional requirement are important to the system (e.g., usability, maintainability, security, reliability, dependability, performance, safety, availability, capacity, portability, etc.)? 5. What appears to be the ONE most important non-functional requirement? 6. What level are the requirements written at (e.g., vision, business, user, system, software, etc.)? 7. What techniques are used to describe the requirements (e.g., text description, diagrams, scenarios, use cases, prototypes, screen shots, database schema / data model, UML, etc.)? 8. The <i>general contents</i> of the requirements document: Constraints identified? Assumptions identified? Risks identified? Scope, system boundary and wider environment described? Change and version control established? 9. The <i>requirements</i> specified in the requirements document: Unique identifiers given to requirements? Priorities allocated to requirements? Rationale, justification or source provided for requirements? Test cases specified for requirements? Traceability of requirements established? 10. The <i>language</i> of the requirements document: Terms you do not know the meaning of? Terms that stand out as not fitting in with the others? Acronyms or abbreviations present? Design and implementation-specific language present (e.g., programming languages, platforms, etc.)? Failure modes, exception conditions and error behaviors specified? |
|---|

3. Visualizing and Scouting Requirements

We propose the concept of ‘scouting’ as a preliminary activity to highlight both when and where it may be worth the effort of conducting a more careful inspection of requirements documents, also pinpointing upon what to focus this effort. The intention is for scouting to be an interactive and collaborative activity centered on a single visual representation of the requirements. All the prevailing approaches to requirements inspection concentrate on the document itself and so, accordingly, mostly revolve around written text. We suggest that a visualization of the requirements document, accompanied by the ten quality-related questions listed in Table 1, would better facilitate this more cursory scouting task. In this way, we do not intend scouting to be a replacement for inspection, but to act as an indicator and to provide for a gentle introduction to a frequently neglected topic in requirements engineering. This forms part of a wider research agenda in requirements engineering visualization [11, 12].

In eXtreme Programming (XP), user stories are short sentences written or drawn on a physical index card to act as “...reminders to have a conversation with your project stakeholders regarding their requirements” [2]. They instigate the ‘real’ discussion that is necessary to come to an understanding on a requirement. Likewise, we anticipate that a visualization of a requirements document could capture the essence of the system concept [10] and act as a trigger for stakeholder discussion about the overall requirements and their quality. When stakeholders are ‘too close’ to a document and its contents, a visual rendering may provide an alternative communication vehicle to scrutinize its content and form more objectively and selectively.

3.1 Wordles

A Wordle is a way to generate a visual representation of a text document [6]. It is a visualization whereby a key word that occurs in the text only appears once in the image but the size of the word reflects its frequency of occurrence in the original text. It therefore highlights the prominence of concepts and captures the gist of the text in a document.

Jonathan Feinberg created Wordles as variations on tag clouds (sometimes more generally referred to as text clouds). A tag cloud is a visual representation of the word content of a website. Tags are usually single words, typically listed alphabetically, with a tag's importance designated by font size or color [25]. Tag clouds have become ubiquitous, appearing on websites and blogs. Tag clouds are primarily used for navigation and visualization on Web 2.0 sites (e.g., Flickr).

Wordles go beyond the original use of tag clouds because they may be used to visualize any text. Speeches, songs, RSS feeds and historical documents are just a few examples. Wordles provide for advanced formatting such as changing font styles, color palettes and rearranging word orientations to make for more

engaging images. Two Wordles are illustrated in Figures 1 and 2. These images display Wordles that have been created with half the words oriented horizontally and half the words oriented vertically in a serif font. All horizontal, vertical, or completely random positions and orientations are possible.



Figure 1. Wordle of an Actual Requirements Document (generated by: <http://www.wordle.net/>)



Figure 2. Wordle of a Requirements Document Template (generated by: <http://www.wordle.net/>)

The main website for creating a Wordle is maintained by Jonathan Feinberg [6]; another more stylistically limited version may be found at IBM’s Many Eyes website as part of a large collection of data visualization tools [13]. Other web-based tools exist, such as TagCrowd [23] and TagCloud (IBM ManyEyes) [13]. The process of using TagCrowd is the same as for Wordle in that in each a visualization is constructed by inserting text into a textbox embedded within the respective webpage. A TagCrowd and a Wordle differ in that a Wordle provides for more creative use of text and layout. While the TagCloud software surpasses the Wordle software in that it allows for greater text analysis, such as the comparison of two documents, Wordle was sufficient for our experimental needs.

Overhead is a major issue with many visualization methods and software systems. Users must download software, install it, learn how to use it and integrate it into their workflows. Hence, a visualization system with nearly a zero transitional barrier for use enjoys a significant advantage over similar systems for adoption. Both Wordle and TagCrowd fit this bill. All a user needs to do to create a visualization is to execute a copy-and-paste operation between his or her document and a textbox on either webpage, then select create. An image follows immediately. Based on ease of use, either the Wordle or the TagCrowd software could have been used for this research. Wordle was selected because of accessibility and the additional features discussed above.

3.2 Hypothesis

Our hypothesis is that a Wordle of a requirements document can provide for an effective visualization to help ascertain the quality of a requirements document at a cursory level. Highlighting prominent terms should emphasize the problem that is being tackled, and for whom, along with the gist of the system concept. It should also be immediately clear whether the document is written in the language of the domain or populated with design constraints. Coupled with questions to guide ‘reading’ the visual representation, it should yield a first impression on quality that is comparable with scouting the text of the requirements document itself.

3.3 Scope

We selected Wordles because they employ a simple translation between two textual representation schemes and only introduce a limited number of discriminators (i.e., size of text). We do not claim that Wordles are a panacea for the problem at hand and we do not suggest that they will help to uncover all the important quality issues associated with a requirements document. In particular, the following quality criteria are considered out of scope and better served by alternative visual representations: Are the requirements written clearly, consistently and unambiguously, and at an appropriate level of detail? Are the requirements atomic? Are there any conflicting requirements? Are there any duplicated requirements? Are the cross-references and dependencies between the requirements correct? Are the requirements complete and correct? Are the requirements feasible? Have the requirements been validated and signed-off? Is there a glossary of terms and definitions to accompany the requirements? Determining suitable visualizations for these quality criteria is an open research question.

4. Preliminary Evaluation

A study was designed to investigate the hypothesis outlined in Section 3.2. The study comprised two parts:

Part I. A task to assess whether it is possible to differentiate those Wordles that have been generated from actual requirements documents

from those that have been generated from requirements document templates. This was to act as a primer to familiarize the subjects with Wordles and to determine whether the actual content of a requirements document gets conveyed over and beyond its structural formatting in a Wordle.

Part II. A task to assess the results from scouting a Wordle representation of a requirements document for quality versus scouting the actual document itself. Comparable performance would support our hypothesis.

Following on from a pilot study, two separate study sessions were conducted. The details and conduct of the study sessions are described below.

4.1 Study Artifacts

For the tasks of Part I, three well-known requirements document templates and two sample requirements documents written by acknowledged experts were selected. All these documents were publically available in an electronic format for use. One of the templates was domain-specific and so contained many key terms. The assumption was that this template would be difficult to categorize. The Wordle for a sample requirements document is shown in Figure 1 and the Wordle for a template is shown in Figure 2.

For the tasks of Part II, three sample requirements documents were selected at random from amongst a number of requirements documents created during a graduate software engineering course. Ideally, these three documents would be at varying levels of quality (i.e., one comprising many of the characteristics of high quality requirements, through to one containing very few characteristics), to investigate whether this differentiation comes out somewhat immediately or not when scouting. However, the quality of the three requirements documents that were selected was not especially high, with a marginal difference in quality between the three documents. The documents were also quite small in size, two being fifteen pages long and one being twenty-one pages long. We were restricted in our choices since we needed to ensure that the artifacts used in this part of the study had never been seen by any of the subjects involved and they also had to be available in an electronic format. We formed a baseline quality assessment for each requirements document using the checklist of Table 1 and ranked the overall quality (A, then B, then C). The Wordles for these three requirements documents are shown in Figure 3.

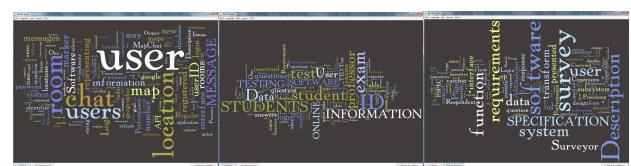


Figure 3. Wordles Used for Part II: A, B and C (generated by: <http://www.wordle.net>)

4.2 Wordle Generation

The process of Wordle creation was kept as simple as possible. Entire requirements documents were first copied and pasted into a text-only word processor to remove graphics and then copied into Wordle. (Later it was found that this step is not required.) Once a Wordle was created, font style, color palette and text orientation were changed until a 'suitable' visualization was reached. Because there are so many possible combinations of these three options it was decided for this initial set of experiments to take a middle-of-the-road approach to Wordle creation by having the software display half the text vertical and half the text horizontal, using a serif font (Powell Antique) for easier readability, and a color space (Blue Chill) that was of medium contrast. It is also possible to select the number of words to display. The default maximum of 150 words was selected. Clearly, experiments could be created to optimize these parameters and assess usability, but the goal of this initial research was to investigate whether Wordles hold promise as a communication medium. For this reason, all the Wordles in our experiments were created to the exact same pattern.

Wordle does not possess a highly sophisticated text processing backend. It neither recognizes plurals nor performs stemming. If such analysis is required, the user must resort to preprocessing. We performed no preprocessing. However, Wordle does allow a user to interactively delete words or select the number of words to display, so that only the most significant words are visualized. In the former case, for example, say the word 'requirements' appears in a requirements document so many times that it obscures the visualization of other words, then the word may be eliminated and Wordle will automatically reorganize the visualization.

4.3 Subjects

Two study sessions were conducted with two separate sets of students. Study group one comprised fifteen graduate computer science students taking a second project-based course on software engineering. They had each previously been involved in creating a requirements document for a software development project for the first time. Study group two comprised eighteen graduate software design and engineering students, who were also working full time as software professionals in the finance industry. Fourteen of these subjects were completing the final course of their degree, and all had completed a requirements engineering and a quality assurance course within the previous two years. They therefore had experience in writing requirements documents and in conducting reviews of requirements documents. The profile of the remaining four subjects in this group was similar except that they had not yet taken the specialist requirements and quality courses.

4.4 Study Procedure

Each of the two study groups was split into two groups at random, the experimental group and the control group. We collected data on whether each subject had either seen or used Wordles before prior to the study.

Part I of the study was given to all the subjects regardless of their group and lasted five minutes. Subjects were presented with five Wordles. They were told that the Wordles had been produced either from a template for a requirements document (i.e., a recommended structure for a requirements document, but containing no actual content) or from an actual requirements documents (i.e., containing real content that specified the requirements for an actual system). The task was to examine each Wordle in turn and to select which category it fell into, one minute per Wordle.

Part II of the study differed depending on the experimental group or the control group. The experimental group was presented with three new Wordles generated from actual requirements documents while the control group was presented with the three requirements documents that these Wordles were generated from, with the document title, system name and authors names removed. The subjects were told that they were requirements documents that had been produced by graduate students. The task was to examine each Wordle or document in turn and to answer an accompanying set of questions by way of speed inspection (i.e., scouting). These questions were those listed in Table 1 but provided in a checklist manner for ease of completion, questions 8-10 becoming fifteen yes/no/unsure checkboxes. Time for this task was restricted to one hour. The subjects were instructed to look for evidence that the topics on the quality checklist were covered in the requirements document by scouting either the document or its Wordle. All the subjects were then asked to rank the *probable overall quality* of the three requirements documents (i.e., best, middle, lowest).

5. Findings and Observations

Only one of the thirty-three subjects had previous exposure to Wordles. This subject was placed in a control group. The high-level results are discussed below.

5.1 Part I: Differentiation

The consolidated results for Part I are depicted in Figure 4. Across both study groups, 66% of all the subjects (on average) were able to correctly differentiate between Wordles of requirements documents templates and Wordles of actual populated requirements documents. In general, the subjects from study group two performed better in identifying this distinction, 83% of the subjects correctly categorizing two of the Wordles in under a minute. One exception was observed in attempting to categorize the Wordle generated from the template containing key terms for a particular domain, as

noted in Section 4.1. (This is Wordle 5 in Figure 4). Only 46% of the subjects identified this Wordle as generated from a template, as we anticipated.

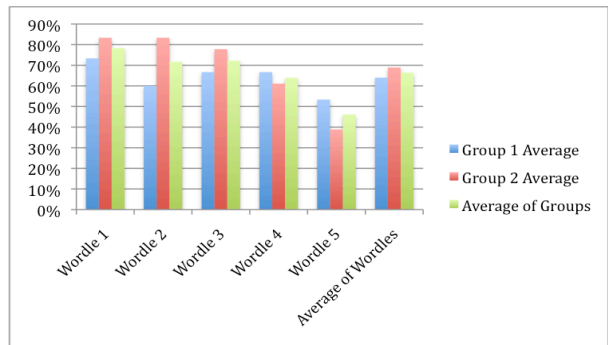


Figure 4. Consolidated Results for Part I

The results appear to suggest that it is possible to differentiate those Wordles with actual requirements content from template ‘buzz words’, in a cursory glance, and when there is no prior exposure to these representations. There is more than 50% chance in all but the trickiest case. It is important that Wordles of empty and vaporware requirements documents are easy to identify. We anticipate that performance would improve as subjects become familiar with the representation and when given more time to examine them, since the study only allocated one minute per Wordle.

5.2 Part II: Scouting Performance

In study group one, the experimental group took 30 minutes (on average) to complete the entire task and the control group took 34 minutes. In study group two, this extended to 40 minutes for the experimental group and 50 minutes for the control group. By removing the four subjects with less prior experience from study group two, the average time became 41 minutes for the experimental group and 58 minutes for the control group. Subjects using the Wordles tended to complete the task more quickly, and the time difference became greater as the subject’s prior experience of requirements writing and review work increased; scouting seemingly becomes closer to a thorough inspection with more experience. The time for the task is also likely to increase with the size of the requirements document, though we did not isolate this factor in our study. This would impact the control group only. If the performance of the experimental and control groups are somewhat comparable, then this has important implications -- it offers a way to be more inclusive with participants and to potentially deal with scale.

Across both the study groups, 39% of the quality rankings (putting the three requirements documents in quality order) aligned with our baseline assessment, the percentage of overall subjects correctly ranking the quality ordering in both groups being depicted in Figure 5. Of these, 51% of the subjects were in a control group and 49% were in an experimental group, so the performance would appear comparable. This is depicted

in Figure 6. What is interesting is that those in study group one, and so those with less prior requirements writing and review experience, performed better with the Wordles when ranking quality accurately than study group two did (with 56% of the correct quality rankings arising from the use of Wordles as opposed to 41%). There is thus some evidence that more benefit is likely to arise from the use of a Wordle when the subjects have had less requirements training; we assume that the subjects are more capable of scouting a requirements document itself for quality when they have had dedicated prior training, as was the case with study group 2.

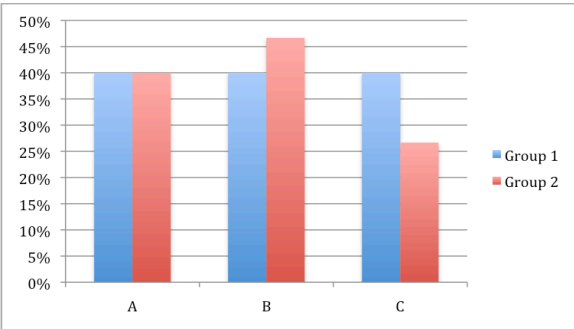


Figure 5. Accuracy of Quality Rankings in Part II

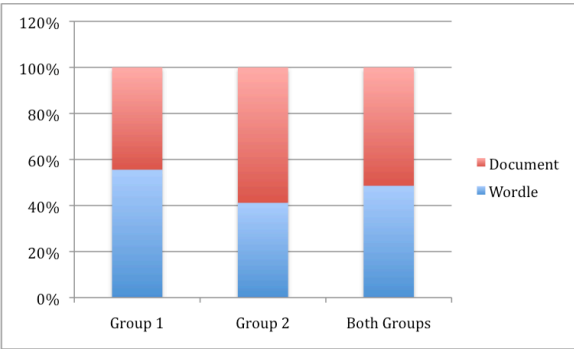


Figure 6. Vehicle Used for Accurate Quality Rankings in Part II

Examining the responses to the quality checklist questions individually, as opposed to just the overall quality ranking, shows an interesting picture. Collating all the responses to all the questions places the requirements documents in the same quality order as our baseline ranking, with 41%, 36% and 23%, as shown in Figure 7. While tabulating all the answers to individual quality questions in the control groups overwhelmingly identifies the highest quality requirements document (A), doing likewise with the experimental group identifies the lowest quality requirements document (C), as shown in Figure 8. What stands out in the experimental groups is the ease of identification of the poorest quality requirements document when the subjects have experience, the other two documents drawing similar results, and this was not so dramatically reflected in the accuracy of the overall ranking they gave in Figure 5. These results show that, despite a careful consideration of quality criteria, the overall impression of quality may not reflect this assessment. This is an observation that

demands further examination: what *exactly* influences perception of requirements document quality?

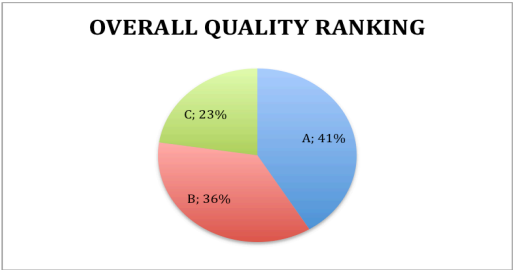


Figure 7. Overall Quality Rankings in Part II (Irrespective of Vehicle)

Two other important observations can be made based on the checklist question data. In study group one, the tendency was for subjects in the experimental group to be twice as uncertain as to whether the underlying requirements document exhibited certain quality properties than the control group. This increased to triple the uncertainty in study group two. Figure 9 tabulates this relative proportion of ‘unsure’ responses to quality questions across groups. Conversely in the control groups, 35-55% of the subjects were able to answer ‘yes’ or ‘no’ to the quality questions, quite categorically, irrespective of whether or not their assessment was correct. In the experimental groups, this ability to assert the presence or absence of quality properties with certainty was reduced to 25-45%. These results would suggest that Wordles are better vehicles for uncovering grey areas and hence for triggering discussion.

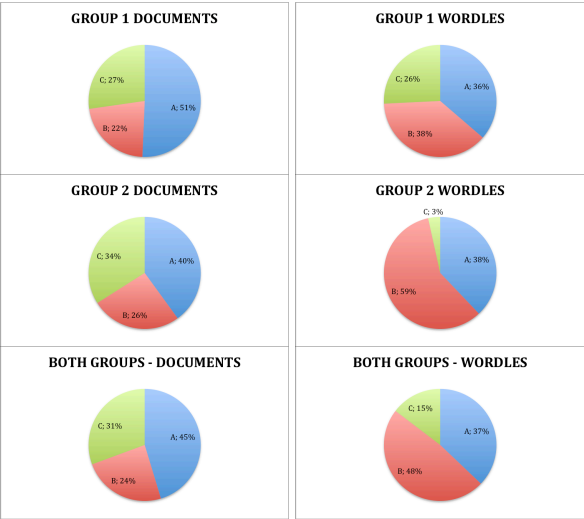


Figure 8. Breakdown of Quality Rankings in Part II

Note that not all the quality checks could be equally well assessed using Wordles as opposed to the requirements documents. All the subjects found it easier to scout the requirements document to make decisions on the presence of change management, traceability and prioritization, and to scout the Wordle to make decisions on language-related issues. The two vehicles would appear to offer complementary strengths.

One other interesting observation on the quality checks was the ability of all the subjects to summarize the system concept, its stakeholders and the key requirements quite reasonably and in a short time period. While all the groups performed similarly here, more elaboration was evident amongst the control group -- the scope of the system tended to creep. One reason for this may have been that more detail raised expectations, associations were made and gaps were filled in. In all cases, subjects stated that more non-functional requirements types had been included in a requirements document than was actually the case, and the control groups tended to find it more difficult to highlight the major non-functional requirement. Where a non-functional requirement type is prominent in the Wordle but not in the requirements document itself, or vice-versa, clarification is obviously necessary. Once again, the tendency for individuals to build a model of the requirements they are scouting that is different from those specified directly in front of them is an important finding and in need of further research.

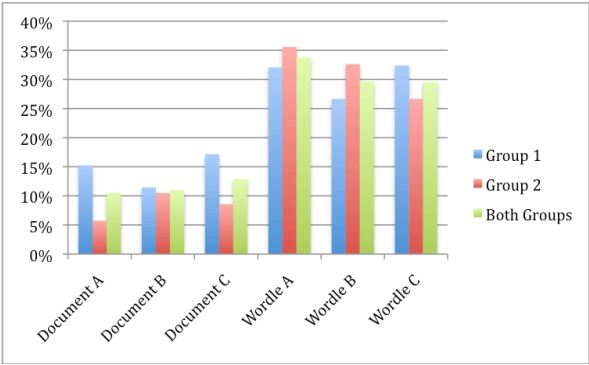


Figure 9. Uncertainty in Part II

5.3 Limitations of the Study

This is a preliminary study designed to evaluate whether there is potential value in using visual representations to support more cursory forms of requirements document inspection. Our work is limited to using Wordles to represent these documents in the first instance and it is possible that there are better representations. Finding the ‘ideal’ visual representation was beyond the scope of our study. Also, our experimental studies were limited in size and by the availability of artifacts. The relatively low number of participants does not make for meaningful detailed statistical analysis. The requirements documents for Part II of the study were not particularly high in quality, varied little in quality and were short. While we ascertained a quality ranking amongst these documents for assessment purposes, this was subtle. The study would be improved with more varied exemplars and a comparison of visualization types and their effectiveness.

More specifically, a number of the subjects struggled with the color scheme chosen to represent the Wordles and with small print on paper. Dark blue text on a black background for some of the words rendered these

words less prominent to the scout. It may also be preferable to view Wordles online, offering zooming capabilities, or via a large projection. The subjects were not asked to explain their assessment and overall ranking due to time constraints. We were interested in speed and did not want to interfere with instinct. However, rationale would have been invaluable, may have mitigated any randomness in the rankings and would have facilitated deeper analysis. In addition, while we suggest that this activity could provide a basis for discussion on the requirements, we did not extend our experiments into this follow-on activity.

6. Conclusions and Future Work

This paper has proposed the concept of requirements scouting using visual representations of emerging requirements documents in conjunction with simple heuristics to guide a more cursory form of quality inspection. It has examined the use of a Wordle for this visual representation and early studies suggest that this holds promise, especially as the size of a requirements document increases and to include stakeholders with little prior exposure to writing or reviewing requirements in this process. The results of a quick quality assessment gained from use of a Wordle over use of its requirements document did not vary so significantly in our studies, which is an encouraging sign. We suggest that Wordles can concurrently act as a shared communicative artifact about which to conduct a directed requirements quality discussion and that there may be further value in using Wordles to visualize other software development artifacts. One area we are currently exploring is their use for the assessment of horizontal and vertical traceability.

Wordles are inevitably restricted in those quality properties they can help to highlight and are not a visualization to support all our software development task needs [18]. But, given it is such a critical and often-omitted task, is recommended that alternative visual representations be explored to further assist requirements inspections. For example, the connectivity and dependency of requirements within a requirements document could perhaps be illustrated using a TextArc [16], while the structure could be conveyed using a Word Tree or Treemap (visualizations also supported by Many Eyes [13]). Our ultimate vision is a dashboard of visual representations that act as a trigger for much needed requirements-related discussions between many parties.

7. References

- [1] Alexander, I.F. and Stevens, R. *Writing Better Requirements*. Pearson Education Ltd., 2002.
- [2] Ambler, A. *Agile Modeling: Effective Practices for Extreme Programming and the Unified Process*. John Wiley & Sons, 2002.
- [3] CMMI Product Team. *CMMI for Development, Version 1.2*. CMU/SEI-2006-TR-008, August 2006.
- [4] Davis, A.M. *Software Requirements: Analysis and Specification*. Prentice-Hall, Inc., 1990.
- [5] Fagan, M.E. Design and Code Inspections to Reduce Errors in Program Development. *IBM Systems Journal*, Vol. 15, No. 3, pp.182-211, 1976.
- [6] Feinberg, J. *Wordle*. Online at <http://www.wordle.net/>, Accessed March 2009.
- [7] Gilb, T. Agile Specification Quality Control. *Cutter IT Journal*, Vol. 18, No. 1, pp.35-39, January 2005.
- [8] Gilb, T. and Graham, D. *Software Inspection*. Addison-Wesley, 1993.
- [9] Glass, R.L. Inspections—Some Surprising Findings. *Communications of the ACM*, Vol. 42, No. 4, pp.17-19, April 1999.
- [10] Gotel, O. In Search of the System Concept. *IEEE Software*, Vol. 23, No.1, pp.102-103, January-February 2006.
- [11] Gotel, O.C.Z., Marchese, F.T. and Morris, S.J. On Requirements Visualization. In *Proceedings of the 2nd International Workshop on Requirements Engineering Visualization (REV'07)*, New Delhi, India: IEEE, 2007.
- [12] Gotel, O.C.Z., Marchese, F.T. and Morris, S.J. The Potential for Synergy between Information Visualization and Software Engineering Visualization. In *Proceedings of the 12th International Conference on Information Visualisation (IV'08)*, London, UK, 8-11 July 2008.
- [13] IBM Watson Research Center. *Many Eyes*. Online at <http://manyeyes.alphaworks.ibm.com/manyeyes/>, Visual Communication Lab, Collaborative User Experience Research Group, Accessed March 2009.
- [14] Kovitz, B. *Practical Software Requirements: A Manual of Content and Style*. Manning Publications Co., 1998.
- [15] Martin, J. and Tsai, W.T. N-Fold Inspection: A Requirements Analysis Technique. *Communications of the ACM*, Vol. 33 No. 2, pp.225-232, February 1990.
- [16] Paley, W.B. *TextArc*. Online at <http://www.textarc.org/>, Accessed March 2009.
- [17] Parnas, D.L. and Lawford, M. The Role of Inspection in Software Quality Assurance. *IEEE Transactions on Software Engineering*, Vol. 29, No. 8, pp.674-676, August 2003.
- [18] Petre, M. and de Quincey, E. A Gentle Overview of Software Visualisation. *Psychology of Programming Interest Group Newsletter*, September 2006.
- [19] Porter, A.A., Votta, L.G. and Basili, V. Comparing Detection Methods for Software Requirements Inspections: A Replicated Experiment. *IEEE Transactions on Software Engineering*, Vol. 21, No. 6, pp.563-575, June 1995.
- [20] Schneider, R.J. System and Software Requirements Validation Through Inspections: Constructive Reading and Mining Requirements from Natural Language Requirements Documents. *Information Knowledge Systems Management*, Vol. 3, No. 2-4, pp.173-194, 2002.
- [21] Shull, F., Russ, I. and Basili, V. How Perspective-Based Reading Can Improve Requirements Inspections. *IEEE Computer*, Vol. 33, No. 7, pp.73-79, July 2000.
- [22] The Standish Group International, Inc. *CHAOS Chronicles Version 3.0*. 2003. Online at <http://www.standishgroup.com/chaos/toc.php>, Accessed March 2009.
- [23] Steinbock, D. *TagCrowd*. Online at <http://tagcrowd.com/>, Accessed March 2009.
- [24] Wiegers, K.E. *Peer Reviews in Software: A Practical Guide*. Addison-Wesley, 2002.
- [25] Wikipedia. *Tag Cloud*. Online at http://en.wikipedia.org/wiki/Tag_cloud, Accessed March 2009.