Combining Qualitative Coding and Sentiment Analysis: Deconstructing Perceptions of Usable Security in Organisations

Ingolf Becker, Simon Parkin and M. Angela Sasse
University College London
{*i.becker, s.parkin, a.sasse*}*@cs.ucl.ac.uk*

Abstract

Background: A person's security behavior is driven by underlying mental constructs, perceptions and beliefs. Examination of security behavior is often based on dialogue with users of security, which is analysed in textual form by qualitative research methods such as Qualitative Coding (QC). Yet QC has drawbacks: security issues are often time-sensitive but QC is extremely time-consuming. QC is often carried out by a single researcher raising questions about the validity and repeatability of the results. Previous research has identified frequent tensions between security and other tasks, which can evoke emotional responses. Sentiment Analysis (SA) is simpler to execute and has been shown to deliver accurate and repeatable results.

Aim: By combining QC with SA we aim to focus the analysis to areas of strongly represented sentiment. Additionally we can analyse the variations in sentiment across populations for each of the QC codes, allowing us to identify beneficial and harmful security practises.

Method: We code QC-annotated transcripts independently for sentiment. The distribution of sentiment for each QC code is statistically tested against the distribution of sentiment of all other QC codes. Similarly we also test the sentiment of each QC code across population subsets. We compare our findings with the results from the original QC analysis. Here we analyse 21 QC-treated interviews with 9 security specialists, 9 developers and 3 usability experts, at 3 large organisations claiming to develop 'usable security products'. This combines 4983 manually annotated instances of sentiment with 3737 quotations over 76 QC codes.

Results: The methodology identified 83 statistically significant variations (with $p < 0.05$). The original qualitative analysis implied that organisations considered usability only when not doing so impacted revenue; our approach finds that developers appreciate usability tools to aid the development process, but that conflicts arise due to the disconnect of customers and developers. We find organisational cultures which put security first, creating an artificial trade-off for developers between security and usability.

Conclusions: Our methodology confirmed many of the QC findings, but gave more nuanced insights. The analysis across different organisations and employees confirmed the repeatability of our approach, and provided evidence of variations that were lost in the QC findings alone. The methodology adds objectivity to QC in the form of reliable SA, but does not remove the need for interpretation. Instead it shifts it from large QC data to condensed statistical tables which make it more accessible to a wider audience not necessarily versed in QC and SA.

1 Introduction

Information technology has become ubiquitous within organisations. From document management to communications, virtually all aspects of business processes are touched upon by IT. These changes have created systems and data that support a huge increase in productivity which in turn makes them – and the data they contain – a target for attacks. Organisations must invest in an ongoing effort to secure IT assets and electronic data. However, security is a secondary activity for businesses, and security mechanisms that get in the way of users' and employees' business tasks are often circumvented, especially when security responsibilities accumulate over time [6]. The gains that IT affords in productivity are often undone by unusable security solutions that place excessive demands on users. The reasons for ignoring or circumventing security have been uncovered in successive studies since 1997 [1].

In various efforts to understand the elements of security usability, qualitative research methods have been used by a great number of works for the analysis of semi-structured self-reports – by individuals such as home users and company employees – of their per-

ceptions and comprehension around security [2, 5, 27] and privacy [21]. Much of this research is open-ended and investigative, although qualitative methods such as Grounded Theory offer a focused and structured approach to analysing textual data arising from these investigations [13].

Individuals are tasked not only with behaving securely, but with using IT securely and applying security technologies to support their activities. We examine the roles of security and usability in the development of IT security software in three large organisations (between 14,000 and 300,000 employees). All three organisations use a large number of off-the-shelf products, but also develop solutions in-house. In all cases the companies develop products at more than one location. The three organisations have very different customers, both governmental and private. More importantly, they prioritise security and usability very differently. The organizations range from a "security first" corporate culture with a low tolerance for deliberate security violations, to one where security is usually not the primary focus of each business unit. The studies are conducted as part of research by the Institute for Information Infrastructure Protection (I3P), and the QC analysis is published as [8]. Their main research question consider "Why each organization added usability and security elements to its software development process", "how and where the organization added them", and "how the organization determined that the resulting software was usable and secure."

The research presented in this paper builds on the QC conducted by Caputo et al. in [8]. Our contributions are as follows: we lay out our hypothesis of gaining additional insights by combining QC and SA in section 2 and describe the methodology in section 4. We perform a sentiment analysis by additionally coding the data for sentiment (section 5.3), independent of the existing QC annotations. This is followed by our results in section 6, which is finished off with a comparison of our findings with the findings from the QC exercise.

2 Aim

Our new approach combines two existing qualitative methodologies into one statistically validated model. Figure 1 depicts the methodology: Both QC and SA work on the conceptions held by people. These conceptions are concealed within (often large) bodies of text, where both methods have developed to expose specific elements of these conceptions. QC focuses on uncovering a structured theory, attempting to explain the relationships between concepts and artifacts. SA reveals the emotions towards the conceptions, revealing contentious conceptions.

In isolation both methodologies have their limitations.

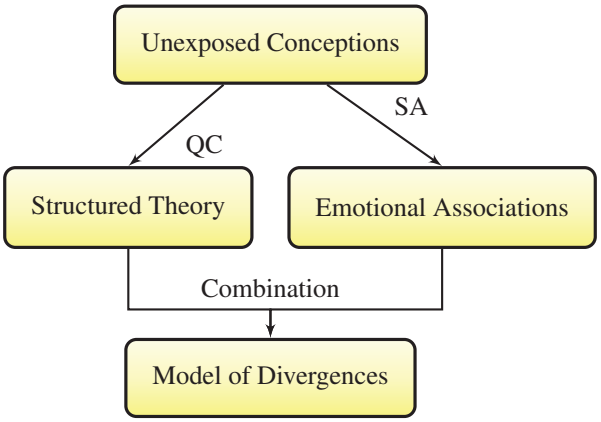

Figure 1: The output structure of our methodology

QC provides a comprehensive overview of themes which are used to construct a grounded theory of the data, yet there is scope to more directly measure the perceived importance of identified themes as according to the individuals under observation. Sentiment Analysis fills this gap: by independently measuring sentiment within the source documents, we get an accurate measure of the expressed sentiment towards each QC concept.

Our methodology combines QC and SA to provide a model of divergences, which reveals the friction points between the themes discovered by QC. The combination approach improves the reliability of traditional QC by providing metrics which further highlight important themes, and can guide application of remedial interventions to critical issues.

3 Background

This paper combines three distinct topics: QC, SA and Usable Security.

3.1 Qualitative Coding

The aim of qualitative coding is the extraction of knowledge from data. Most coding techniques iterate over textual data, where the researcher applies labels ('codes') to sections of the text (also referred to as quotations). These analysis techniques vary in the amount of interpretation of the data that is required, and by describing qualitative coding as a black-box methodology overlook the opportunity to convey the reliability of the exercise. In these cases an additional validation step as described in this research would support transparency.

The first step of the annotation process is *open coding* [25]. While developing QC, the researchers constantly refine their codes in an iterative process, questioning the choice of every code by comparing it to all other in-

stances of the same code throughout the data. The individual codes are further grouped into conceptual categories as part of a process Strauss and Corbin called *axial coding* [25].

It is at this stage that we extract the coding for our analysis. Most QC exercises span around 100 codes, spread across 15-25 categories (e.g. [20]), where these numbers of course vary depending on the size of the study. Each code is applied many times in the source documents, typically leading to many thousands of quotations. The quotations (of different codes) are often overlapping. In some sense the QC represents a very accurate topic model, but compared to statistically-based topic models, the high level of rigour and consistency of the QC represent a reliable basis for further analysis.

3.2 Sentiment Analysis (SA)

The definition of sentiment is remarkably vague: Pang and Lee describe it as "settled opinion reflective of one's feelings" [18]. Yet when identifying conflicts between usability and security, it is essential to consider an individual's sentiment as it reflects the importance of the issue to the individual. It is important to understand the dependencies between issues in order to identify the root cause. In many cases, solving a relatively minor issue that evokes strong emotional responses is a prerequisite for solving more important, but less noticed issues.

Until today, much of the research still restricts the classification space into three scores: positive, negative and objective/neutral [18]. This approach is often sufficient as the sentiment scores are aggregated over the unit of interest, such as all Twitter messages containing a specific hashtag. There are various methods for identifying sentiment. While some approaches attempt to algorithmically determine sentiment on the basis of sentiment dictionaries (where each word has a sentiment score), syntax and semantics, the most successful approaches are based on documents manually annotated for sentiment, in which case Sentiment Analysis becomes a simple annotation task with a fixed code book. These documents then form the training set of supervised learning techniques.

3.3 Complexity of Usable Security

Usable security integrates security and usability considerations with the primary task to form one continuous process [28]. The ideal outcome is an increased level of security with no loss of usability. Preserving usability and security together should enable increases in productivity as barriers are identified and eliminated.

Yet practice is far from this ideal world. Users face a huge number of barriers due to ill-fitting security in their productive processes every day; and their compliance budget [6] — the amount of rules an individual will follow before taking shortcuts — is regularly exceeded. These findings have led researchers to investigate non-compliant behaviour in greater detail, with surprising results: the individual's rationale for non-compliant behaviour can very well be rational. In fact, the non-compliant behaviour is worth studying as it reveals not just organisational failures but alternative approaches to maintaining security and usability developed by individuals themselves [15]. A further dimension to the complexity of security is the cognitive strain placed on the individual when performing security tasks. Security is rarely designed with humans' upper bounds of comfortably performing cognitive tasks in mind [7].

All of this research highlights a divergence from the intention of security to support business processes and the implementation of security. It is this security misalignment that enables those vulnerabilities that have been left unresolved, and introduces new weaknesses. Managing, formulating and implementing effective policies requires understanding of the causes and consequences of this misalignment. Modelling these asymmetric divergences is a challenging task [9], but should be the basis of any new policy considered. Resolution of this misalignment requires insights drawn from divergent fields of research, including behavioural sciences [19].

4 Methodology

This section describes the combination of QC and SA, the final step in figure 1. We present a methodology that combines the structured theory produced by QC and the emotional associations labelled through SA. This produces a more nuanced model of perceptions, crucially associating contributory factors with indicators of the perceived effort and stress associated with interacting with them, as is seen with attempts by employees to complete tasks and navigate security controls.

4.1 The Basic Unit of QC+SA

As input the methodology uses documents that have been annotated using both QC and SA. By analysing intersections between QC quotations and sentiment annotations we compute a distribution of sentiment for each QC code. As every word in the source documents has some sentiment annotated with it, each quotation (as a sequence of words) will be linked to a number of validated sentiment instances as well as to the codes.

For each QC quotation, we aggregate the sentiment annotations that are linked to it. This aggregation is described in the following section. Each QC code has many

quotations linked to it, and hence a distribution of sentiment. This distribution will be different for each QC code. Using statistical tests, we can analyse the differences in sentiment distributions for individual QC concepts, and draw conclusions concerning the emotional state of the QC codes.

Further, we can restrict the sampling space to individual organisations or individual interviews to create a comparison across different aspects. In the case study presented here for example, this allows us to compare the sentiment of developers, managers and usability professionals towards specific factors in the application development process.

Assuming that the QC analysis is conducted with sufficient rigor, a study of this kind can be repeated within and across different organisations over time and allow for direct comparisons with the previous data sets. This will allow researchers to measure the perception within the organisation of a business process before and after changes and compare results - something that has previously been difficult to do reliably.

4.2 Classifying Instances of QC+SA

Given a QC quotation, we retrieve the sentiment associated with the raw text. There may be multiple sentiments attached to a single quotation. Experiments with constructed test data have shown that the most reliable way of averaging the sentiments of quotation text is by weighting on the number of words of overlap between the sentiment text and the quotation text. This is because the boundaries of the sentiment annotations are the most unreliable part of the annotations. Even when sentiment annotators agree on the overall sentiment, the exact location of the boundary of the sentiment remains fuzzy. This gives the following formula for the sentiment of a quotation q, where S is the set of all sentiment annotations:

$$sent(q) = \frac{\sum_{s \in S : |s_t \cap q_t| > 0} s_s * w(s,q)}{\sum_{s \in S : |s_t \cap q_t| > 0} w(s,q)}. \quad (1)$$

Here s_t and q_t denote the words of the quotation or sentiment annotation, and s_s denotes the sentiment score (either -1, 0 or 1). Hence $|s_t \cap q_t|$ denotes the number of words that both the text of the sentiment annotation s and the quotation q have in common, with

$$w(s,q) = \frac{|s_t \cap q_t|^2}{\min\{|s_t|^2, |q_t|^2\}}. \quad (2)$$

The weights in equation (2) are squared to decrease the influence of small overlaps with neighbouring sentiment annotations.

For a given code c, the distribution of sentiments is just the sentiment score of each of its quotations:

$$sent(c) = \{sent(q) : q \in c_q\}, \quad (3)$$

where c_q is the set of quotations associated with c.

4.3 Worked Example

Let us consider a fictitious example which has been annotated for sentiment by two annotators:

Only when you have got a large development team you need usability experts. But they can be useful.

Here, underlines represent negative sentiment and overlines represent positive sentiment. Where there are less than two lines present, a neutral sentiment exists. Independent of the sentiment annotations, this excerpt has been annotated with two QC quotations. The first quotation q_1 spans the first sentence and is linked with the code *Development: Team size*. The second quotation q_2 begins at "usability" and contains the remainder of the excerpt. The second quotation is linked to the QC code *Actor: Usability expert*.

In order to determine the sentiment for each of these quotations, we apply equation (1). Consider the first quotation (q_1, the first sentence) of length 13. There are four sentiments present: The first spans the entire sentence, the other two end and begin to the left and right of the second 'you' respectively. The last sentiment is the neutral and spans only the word 'you' (on line 2). The length of these four sentiments (s_1, s_2, s_3 and s_4, say) are 13, 9, 8 and 1 words respectively. Equation (2) gives us the weights for each sentiment given the quotations. This gives

$$\begin{aligned} w(s_1,q_1) &= 13^2/\min\{13^2,13^2\} = 1, \\ w(s_2,q_1) &= 9^2/\min\{9^2,13^2\} = 1, \\ w(s_3,q_1) &= 3^2/\min\{8^2,13^2\} = 9/64 = 0.14, \\ w(s_4,q_1) &= 1^2/\min\{1^2,13^2\} = 1 \end{aligned}$$

as for $w(s_3,q_1)$ only 3 of the 8 words of s_3 intersect the quotation.

Given equation (1), we can now work out the average sentiment for this quotation as -0.592.

This score is the sentiment for one quotation alone — but each QC code is linked to many quotations, giving us the distributional sentiment to base our further analysis on.

5 The Three Case Studies

We applied our methodology to a study on usable software development. The study details are presented in [8]

and [20]. Three large US based organisations were selected to be studied by a team of 5 researchers with the aim of characterising usable software development activities. The study was driven by three research questions: 1. Why has an organisation added usability and security elements to its software development process? 2. How and where were they added? 3. How did the organisation determine that the resulting software was usable and secure? Supporting hypotheses examined the effect of individual roles on the development process. Research questions were then investigated by way of semi-structured interviews with individual employees of the organisations.

5.1 Data Collection

Interviews were conducted on the organisations' sites. At least three researchers were physically present at each interview. Audio recording devices were not permitted, hence three interviewers orthographically transcribed the conversations verbatim. Inconsistencies across transcripts were then reconciled to produce a merged transcript for each interview. In total, 21 interviews were conducted, with a combined length totalling 87496 words. Seven interviews were conducted at organisation A and nine and five interviews at organisation B and C respectively.

5.2 Qualitative Coding

The team used the Atlas.TI qualitative analysis software [3]. The coding was carried out by five expert coders, one of whom is one of the authors of this paper.

There is little guidance on distributing QC analysis in the literature. While Glaser and Strauss describe the procedure of QC as an iterative refinement of code/category/theme by comparison of all instances to each other, this becomes counter-productive when sections of the source text are distributed across a number of annotators.

As a trade-off between methodological accuracy and efficiency the annotators began by all coding the same interview individually. An entirely open approach was chosen, in line with Strauss and Corbin [25]. The coding choices of all five coders were discussed in a dedicated session to expose the biases of individual coders early on. Subsequently, every interview was independently coded openly by at least three coders, expanding the code book as necessary. After the open coding of each interview, the annotations were discussed in plenum to identify and resolve all differences in the meaning of open codes across coders. This step was intended to mimic the constant comparison [14] of all quotations of one code to each other though, rather than comparing all instances, here coders' code definitions were compared and unified. This process aligned each coder's individual code book, giving a unified QC that is in agreement with every coder.

This process was repeated for the axial coding of the data. Here the discrete codes were grouped into conceptual families that reflect commonalities among codes. 76 codes span 3737 quotations, roughly equally distributed between the three organisation. The codes span topics such as usability, security and topics related to the organisational structure and business processes of each organisation. The identified code families show the focus of interviews on the interactions of security and usability within organisations. There are also a significant number of codes concerning decision drivers and the focus of these decisions (i.e. goals, methods and solutions).

5.3 Sentiment Annotation

The source documents under analysis have a high degree of specialised language, being as they are driven by subject-specific expertise. Without verifying the accuracy by annotating at least a section of the primary documents manually, applying an off-the-shelf machine learning approach does not satisfy our desire to ensure accuracy.

Since sentiment annotations are inherently subjective, multiple independent annotators are required to ensure consistency and provide a score of inter-annotator agreement across annotations. As we already need to annotate a section of the source documents to verify the quality of the annotations, we decided to manually annotate the entire set of raw documents, providing us with a set of gold standard documents to base further research on.

5.4 Methodology of Manual Sentiment Annotations

The methodologies of sentiment annotations in previous work vary significantly. Strapparava and Mihalcea ask their annotators simply to annotate the given title for sentiment [24]. No further guidance or training was carried out. Annotators were free to use any additional resource. The instructions given to annotators by Nakov et al. are similarly short [17], but a list of example sentences with annotations is given.

In light of the issues presented by these two approaches we have chosen to give more detailed instructions but have refrained from giving explicit examples. The instructions given to the annotators can be found in figure 2.

For each of the documents please annotate each sentiment occurrence as either positive or negative. If you think no sentiment is present, just leave the text as it is. We are interested in the underlying, implicit sentiment. This is what the interviewee thinks about the topic at the given expression. Be generous when annotating, annotating sentiment is inherently subjective. As you are annotating transcribed speech, it may be very possible that sentiments change abruptly. Make sure that the content of the annotations should preserve the context, but this may not always be possible.

Figure 2: Annotation instructions given to annotators

5.5 Analysis

As part of our practical contribution 21 transcribed interviews have been manually annotated by 3 annotators, two of which are authors of this article. The annotations have been carried out using Atlas.TI [3] with a code book limited to *positive* and *negative*.

Annotator	total	#pos	#neg	avg #words
1	1450	731	719	17.129
2	1677	873	804	32.291
3	870	419	451	15.380

Table 1: Annotation statistics per coder

The combined length of the 21 transcribed interviews is 87,496 words. Table 1 lists the distribution of annotated phrases for each of the annotators. The difference in the number of phrases that have been annotated is surprising: annotators 1 and 2 have each annotated many more phrases with sentiment than annotator 3. While annotators 1 and 2 have a similar number of sentiment annotations, each annotation of annotator 2 spans nearly twice as many words.

These divergent results highlight the difficulty of clearly annotating sentiment. As we gave no examples of sentiment annotations to the annotators, the annotation lengths varied.

5.6 Cross Annotator Agreement

In the literature there is wide spread disagreement about the choice of metric and its interpretation [17, 26]. The two measures widely used in literature are K [22] (also called Multi-π [12]) and Fleiss' Kappa [10] (also called Multi-κ).

The annotation task described above is a multi-coder boundary annotation problem with multiple overlapping categories. The issue for this class of annotation problems is that the reliability should not be calculated token wise (unitise, as K and Kappa do), but should rather respect the blurry nature of their boundaries — annotators may agree that a specific sentiment is present, but have different begin and end tokens. One measure that does not overly penalise on non-exact boundary matches is Krippendorff's α_U [16], a non-trivial variant of α. Unfortunately we could not find a single use of this measure in the literature.

For K (or Multi-π) and Fleiss' Kappa (or Multi-κ) we can report agreement figures of 0.59 and 0.60 respectively.

	% by phrase	% by words
Perfect agreement	48.60	47.49
Majority agreement	95.55	96.42

Table 2: Agreement statistics

An alternative measure used widely is an agreement table [11]. Table 2 represents an accumulation of an agreement table. Perfect agreement represents the percentage of all-negative, all-neutral and all-positive tokens. Here agreement rates are weak, with 48% of phrases showing full sentiment agreement. To soften the measure slightly, we include a figure for majority agreement, where at least 2 of the annotators agree on the sentiment assignment of a phrase or token.

5.7 Discussion

The reliability values presented here can be classified as reasonably consistent. A recent publication does not report annotator agreement metrics but reports "accuracy bounds" without specifying their meaning or derivation [17]. It seems likely that the reported bounds of between 77% and 89% represent majority agreement, which would fit well with our findings. Interpreting K and Fleiss' Kappa is more difficult. The literature does not agree on strict bounds for these measures, but only values > 0.67 are generally seen as reliable, but other researchers argue that $0.40 < K \leq 0.60$ indicates moderate agreement [11]. Our divergence may be a result of the ambiguity of sentiment annotations. Emotions are perceived differently by individuals, partly due to different life experiences and partly due to personality issues [4] and gender [23]. Secondly, the annotation process itself may be responsible for these variations. By phrasing the annotation instructions vaguely, a large number of weak sentiments have been annotated. Rather than penalising the process for these inaccuracies, we argue that, in this subjective context, this level of uncertainty

Code	Label	Freq	p	t	r	mean
67	Usability problem	48	0.000	−6.333	0.612	−0.222
69	Usability problem: tradeoffs	13	0.000	−4.854	0.510	−0.395
45	Security problems	36	0.000	−4.452	0.478	−0.161
68	Usability problem: difficult to understand	8	0.000	−3.678	0.410	−0.380
14	Development: conflict	12	0.001	−3.424	0.386	−0.259
27	Other conflict	13	0.013	−2.479	0.290	−0.143
49	Security success criteria: Better than before	6	0.024	2.258	0.266	0.480
22	Education, training, skills	32	0.032	2.141	0.253	0.268
51	Team	36	0.043	2.028	0.240	0.252

Table 3: T-test comparing the distribution of sentiment of one code to the distribution of the sentiment scores of all other codes of organisation A. The overall mean sentiment is 0.1226 with 67 degrees of freedom.

is beneficial. Instead of aggregating the sentiment annotations, preserving the uncertainty for the down-stream applications will lead to an enhanced understanding as these tasks will utilise the existing measures of uncertainties in the statistical tests. This will allow us to be fully confident of the results of the analysis given the limitations presented here.

6 Results

In this section we discuss the results of the methodology. Results are presented in three categories: first, each organisation is analysed in isolation; second, the three organisations are compared and third, the different interviewee groups are compared across the organisations.

Each individual organisation has internal security experts, developers and usability experts for the creation of internal and commercial security management products. The development processes of these products are the focus of the interviews, specifically how the products are designed to be usable and secure, and what – if any – criteria have been used to measure usability and security.

6.1 Per-Organisation Analysis

For each QC code 'c' a t-test was conducted comparing c's distribution of sentiment (equation (3)) against the distribution of sentiment of all other codes (i.e. the union of equation (3) for all other codes) applied to that organisation. These distributions of sentiment are distributed approximately normally in $[-1,1]$. If the distribution of c differs to an extent that is statistically significant, the opinions expressed in the quotations linked to c are significantly different than the opinions expressed on average. It is these codes that tell us what the issues and concepts are that the interviewees feel strongly about.

Tables 3 and 4 shows the output of such an experiment for organisation A and B (similar results are produced for organisation C, but not presented here). Only those codes that exhibit a statistically significant variation are listed. Columns p and t give the significance of the correlation. This has been converted into Pearson's r value, indicating the strength of the correlation. The last column lists the mean of sentiment distribution of that code.

From Table 3 it can be seen that conflicts and problems are prevalent in the organisation. Usability in particular is causing a significant amount of negative emotion, as the trade-offs made between usability and security leave a negative impact on the development process. This highlights the problem of adding usability as an add-on to an existing product (as is the case in organisation A). The three codes with statistically significant positive scores contrast this: organisation A prides itself in providing better security. The provided education and training are well regarded and employees like working in their existing team.

The issues found in organisations B (table 4) and C share similarities with organisation A: Usability is an add-on to existing products. This creates conflict in the development process, as developers struggle to understand the usability problem (as the significant negative emotions for the code *Usability problem: difficult to understand* highlight). But we can identify some positive messages from these organisations too: interviewees of organisations B and C agree that *usability as a goal* is desirable and some *usability success criteria* are seen to be statistically significantly more positive. While the development process struggles to integrate usability, there are positive instances (such as *user satisfaction* in B and *better functionality* in C) where the benefit of making the product more usable is bearing fruits. Yet the *funding of resources* and the *organisational structure* (in B) as well as the *corporate culture* (in C) wear heavily on the development process.

Organisation B stands alone in the positive view of their ability to *measure security*, although their metric is

Code	p	t
Usability problem	0.000	−6.1
Security problems	0.000	−5.2
Development:conflict	0.000	−4.1
Usability problem: difficult to understand	0.000	−3.9
Usability success criteria: user satisfaction	0.001	3.5
Usability problem: tradeoffs	0.003	−3.0
Resources: Funding	0.004	−2.9
Usability goal	0.009	2.6
Other conflict	0.012	−2.5
Security methods: Measurement	0.013	2.5
organisation: structure	0.021	−2.3
Security methods: active monitoring	0.030	2.2
Security problems: access control	0.043	−2.0

Table 4: T-test of organisation B. Mean sentiment is 0.08489.

defined unscientifically as *'it is secure if we can't break it ourselves. And we continuously try'*. By pushing security as far as possible, it supersedes all other stakeholders in the product – including usability. This issue is amplified by the positive view towards this approach: the organisation is proud of their security. In contrast, employees suffer the shortcomings of the organisation's approach to security every day, as the negative view on *access control* highlights.

6.2 Cross-Organisation Comparison

With our methodology a t-test can be conducted to compare the distribution of sentiment scores between the organisations for each of the codes. In table 5, an arrow pointing upwards represents a statistically significantly more positive sentiment score compared to the sentiment scores of the other organisations; similarly an arrow pointing downwards represents a statistically significant more negative score (with $p < 0.05$). A horizontal arrow represents a non-significant change towards positive or negative. A field that is left empty represents insufficient data for this organisation and code.

The data in table 5 shows some strong trends. For the majority of statistically significant variations the quotations belonging to organisation A have more positive emotions attached. Similarly organisation C does not exhibit a single code which is more positive than in the other organisations. This pattern may point to the overall morale of the organisations in question: the sentiment portrayed by the interviewees at organisation A was a lot more positive than at organisation C. This is reflected

Code	A	B	C
Actor: developer	⟷	⟷	↓
Actor: salesperson	↑		↓
Actor: usability specialist	⟷	↑	↓
Development	↑	⟷	↓
Development: process	↑	⟷	↓
Development: requirements	⟷	↑	↓
Education, training, skills	⟷	⟷	↓
Organisation: corporate culture	↑	⟷	↓
Other conflict	↑	⟷	
Resources	↑	⟷	⟷
Resources:funding	↑	⟷	⟷
Security	↑	⟷	⟷
Security methods: measurement	↓	↑	
Team	↑	⟷	↓
Tools: development	⟷	↑	⟷
Usability	⟷	⟷	↓
Usability problem	↑	⟷	⟷

Table 5: T-test comparing the three organisations. The up, horizontal and down arrows indicate positive, neutral and negative variation respectively at $p < 0.05$.

in codes such as *Team*, *Organisation: corporate culture* and *Development*, where both A is uniquely more positive and C uniquely more negative than the other organisations.

The negative morale in C may seem unsurprising. However, the existing conflict between usability experts and the rest of the organisation is further highlighted by the relatively negative views towards the three actors types *developer*, *salesperson* and *usability specialist*. The actor *salesperson* did not show up in the analysis of each organisation in isolation, but here it suggests another source of conflict.

For organisation B only four codes display significant variations and all of these are positive. The fact that codes such as *Usability problem* are not significantly more positive for organisation B than for A and C conflicts with the significantly more positive code *Actor: usability specialist* in B. This reinforces the assessment that different aspects of usability have been accepted to different degrees. The same conclusion can be drawn for organisation A. While *Usability* is seen more positively than in the other organisations, *Actor: usability specialist* is not. The understanding of what usability means in practice is a point of contention.

6.3 Cross-Role Comparison

Here we explore the potential sources of conflict from the perspective of those who live them, by assessing the interviews according to the three interviewee role categories illustrated in table 6.

	A	B	C
Number of interviews	7	9	5
Number of developer	5	2	2
Number of security specialist	2	8	0
Number of senior usability expert	0	0	3

Table 6: Distribution of interviewee types over the organisations

All of the interviewees aligned with one of the three categories apart from in organisation B where one interviewee was classified as both a developer and security specialist. Note that the distribution of roles varies, despite an original intention for there to be an equal split [20].

Code	D	S	U
Actor: developer	⟷	⟷	↓
Actor: salesperson	⟷	⟷	↓
Actor: usability specialist	⟷	⟷	↓
Decision driver	⟷	⟷	↓
Development	⟷	↑	↓
Development: process	⟷	↑	↓
Development: requirements	⟷	⟷	↓
Education, training, skills	⟷	⟷	↓
Organisation: corporate culture	⟷	↑	↓
Security goals	⟷	↑	⟷
Security goals: preserve reputation/funding	↓	⟷	
Security success criteria: better than before		↑	
Team	↑	⟷	↓
Usability method	⟷	⟷	↓
Usability method: testing	↓	⟷	⟷
Usability success criteria	⟷	⟷	↓

Table 7: T-test comparing the three interviewee types. The up, horizontal and down arrows indicate positive, neutral and negative variation respectively at $p < 0.05$.

Table 7 follows the format of the previous section but with *D, S & U* standing for *Developer*, *Security expert* and *Usability expert* respectively. The table summarises the perspectives across all interviewees who share each role classification. In the case of usability experts all statistically significant variations are more negative, more so than for developers and security specialists. This may be linked to the results of the analysis between organisations: the only three usability experts in our data set were at organisation C.

It is clear that security experts have the most positive view. This reinforces our assertion that the focus of product development remains on security and that the development process is tailored towards security over integrating usability. The positive feeling towards *corporate culture* supports this. The negative emotions of the developers towards *usability method: testing* highlight an additional shortcoming, in that developers fail to see any benefit in usability testing and instead regard it as adding additional strain.

While powerful comparison tools, tables 5 and 7 do suffer from a potential bias due to the lower number of interviews that make up the separate organisations and employee types. Further, for each of these tables up to 228 t-tests were performed which raises the chance of false positives. Yet even with a conservative Bonferroni correction, some interesting artifacts remain statistically significant, shrinking the number of significant variations in tables 5 and 7 by approximately one third. Further, as described in the following section, the results are mostly in line with the pure qualitative analysis, validating the approach chosen.

6.4 Comparison to QC Findings

Here we summarise the findings from the complementary quantitative coding work [8] and discuss the additional benefits of our approach. Caputo et al. hypothesised three distinct explanations of why changes in the software development process might lead to more usable security (from [8]): 1. The "key individual" theory: Improved outcomes resulted not from the process changes but instead from the efforts of a single individual who cares about usable security; 2. The "experienced team" theory: Improved outcomes resulted not from the process changes but instead from the team's prior experience in building usable security, and; 3. The "incentives" theory: Improved outcomes resulted not from the process changes but from incentives placed on team performance with respect to usable security.

Of note is that none of these theories were confirmed in their analysis. Our results agree: organisation C is the only organisation with usability experts, and for this organisation the positive codes are *usability expert to develop software* as well as *use cases* (see section 6.1). As we stated previously, negative codes such as *Usability-security trade-offs* and *development conflicts* highlight that their impact is small. In general when comparing the three organisations in table 5 or the three different

employee types in table 7 we do not find indications to support any of the three theories. Hypothesis 1) can be analysed particularly well by our methodology as it investigates emotions towards security — exactly what our methodology focuses on through its use of SA. The original quantitative analysis did not consider the use of sentiment.

Rather, the authors present a list of five findings: 1. Usability is a grudge sale: only when losses in sales could be linked to a lack of usability, did the organisation respond; 2. The negative effects of a lack of usability occur at the organisational level and are not passed on to the developers. Hence there are no incentives to deliver usable software. This is the exact opposite of the third hypothesis above; 3. Wildly varying definitions of usability; 4. Lack of knowledge by developers of capabilities and limitations of human perception and cognition, primary task, and context of use, and; 5. Developers think they know users because they use the software themselves.

Our methodology provides a more detailed picture, detailing the extent of the "divergences" illustrated in figure 1. We are able to assert that the interviewees in fact acknowledge the importance of building a usable application (see the analyses of table 3 above in section 6) - but when it comes to security, they lack knowledge on how to reconcile what they conceive as competing demands. Our analysis shows this stems from a number of factors: there is no definition of the usability problem, and there is an existing belief that security 'comes first' in the organisations' priorities, and hence in the development processes. There are some positive notes however: in organisation C, *personas* were perceived positively as a usability tool to aid the development process.

The methodology also facilitated analysis of the differences between the three organisations. While the original study [8] attempted to identify exemplary development processes that integrate security and usability, the authors did not find practices that could be recommended. Yet our analysis detects positive differences — in terms of partial improvements that can serve as building blocks for an integrated development process. One could argue that these may have been found with a more rigorous qualitative analysis, but a quantitative approach simplifies the task of comparing across three organisations.

Caputo et al. finish with some open questions which can be answered by our methodology. They speculate that the integration of usability into the software development process is less important than having motivated developers and usability specialists. We can support this hypothesis: Our data has shown that the conflict between usability and security centers around the individual employees and the organisational culture, rather than the software development process. The addition of usability experts to organisation C has shown positive effects on usability tools, as well as codes such as *personas*. Resolving the misconception of a security-usability tradeoff will go a long way to improving usability of security.

Caputo et al.'s second open question concerns cultural barriers to usable security. This manifests in different perceptions of usability throughout the organisation. Our analysis between the different types of employees certainly answers this open question: there are clear differences between the developers, security and usability experts that we described in section 6.3 and table 7.

7 Conclusion

We have introduced a new methodology: by performing an additional level of analysis on Qualitative Coding (QC) with Sentiment Analysis (SA), we can gain additional insight into the emotional colouring of statements.

As a proof-of-concept, we performed this analysis on QC text from 21 interviews with developers, security experts and usability experts in 3 organisations. Whilst the QC analysis uncovered that all 3 organisations were able to 'talk the talk', 'walking the walk' of usable security was a different matter. There were no usability criteria, and few usability methods were employed during the development of the products we discussed.

Our analysis agrees with many of the original QC findings, but from the QC exercise condenses the data requiring interpretation into a number of tables of statistically significant rows. This mechanism serves as a filter for pointing out specific findings that were missed in the original QC analysis. We are able to approach the original dataset from different angles, and compare aspects across organisations and employee types allowing us to draw additional conclusions through cross-comparison.

Through our methodology, security and policy managers can pinpoint friction points and conflicts in organisation processes not only through interview studies, but also other shared communications platforms such as corporate forums and dedicated support channels. For security researchers, this repeatable method offers a powerful tool that generates verifiable quantitative results to harden the results of qualitative analysis. We also explore the potential to transfer findings across organisations to different teams, where the methodology can identify aspects of professional cultures shared across separate organisations.

For future work, appropriate reliability metrics for QC are needed, to ensure that future studies can be compared by the quality of the annotations. There is also room to explore the analysis further - cross-linking different codes and the sentiment annotations could potentially create a powerful deductive tool for researchers, although visualising multi-dimensional relationships is non-trivial. As analysis becomes more elaborate there is

the challenge not just of gathering more source data, but also annotating it. Future research may then explore how machine learning can be used to automate annotation.

8 Acknowledgements

We would like to thank the LASER program comittee, and in particular our shepherd. The authors are supported in part by UK EPSRC grants, no. EP/G037264/1 and no. EP/K006517/1. We would like to acknowledge the contribution of Adam Beautement, toward the manual sentiment annotations, as well as Deanna Caputo, Shari Lawrence Pfleeger, Paul Ammann and Jeff Offutt who performed the interviews and coded the transcripts as part of the I3P project funded by NIST and DHS.

References

1. Adams, A., and Sasse, M. A. Users are not the enemy. *Commun ACM*, 42(12), 1999: 40–46.
2. Ashenden, D., and Sasse, M. A. CISOs and organisational culture: their own worst enemy? *Computers & Security*, 39, Part B, 2013: 396–405.
3. ATLAS.ti GmbH. ATLAS.ti. Berlin, 2013.
4. Barrett, L. F. Valence is a basic building block of emotional life. *J Res Pers*, 40(1), 2006: 35–55.
5. Bartsch, S., and Sasse, M. A. How users bypass access control and why: the impact of authorization problems on individuals and the organization. *Proceedings of the 21st European Conference on Information Systems*. 2013, Paper 402.
6. Beautement, A., Sasse, M. A., and Wonham, M. The compliance budget: managing security behaviour in organisations *Proc. NSPW '08*, 47–58.
7. Benenson, Z., Lenzini, G., Oliveira, D., Parkin, S., and Uebelacker, S. Maybe Poor Johnny Really Cannot Encrypt: The Case for a Complexity Theory for Usable Security *Proc. NSPW '15*, 85–99.
8. Caputo, D., Pfleeger, S., Sasse, A., Ammann, P., and Offutt, J. Barriers to usable security: three organizational case studies. under Review. 2016.
9. Caulfield, T., Pym, D., and Williams, J. Compositional security modelling. In: *Human Aspects of Information Security, Privacy, and Trust*. Ed. by T. Tryfonas and I. Askoxylakis. Vol. 8533. LNCS. Springer, 2014, 233–245.
10. Davies, M., and Fleiss, J. L. Measuring agreement for multinomial data. *Biometrics*, 38(4), 1982: 1047–1051.
11. Di Eugenio, B., and Glass, M. The kappa statistic: a second look. *Comput. Ling.* 30(1), 2004: 95–101.
12. Fleiss, J. L. Measuring nominal scale agreement among many raters. *Psychol Bull*, 76(5), 1971: 378–382.
13. Glaser, B. G., and Strauss, A. L. The discovery of grounded theory: strategies for qualitative research. Chicago: Aldine Transaction, 1967.
14. Harry, B., Sturges, K. M., and Klingner, J. K. Mapping the process: an exemplar of process and challenge in grounded theory analysis. *Educational Researcher*, 34(2), 2005: 3–13.
15. Kirlappos, I., Parkin, S., and Sasse, M. A. Learning from "shadow security": why understanding non-compliance provides the basis for effective security. *Proc. USEC*. 2014.
16. Krippendorff, K. On the reliability of unitizing continuous data. *Sociol Methodol*, 25, 1995: 47–76.
17. Nakov, P., Rosenthal, S., Kozareva, Z., Stoyanov, V., Ritter, A., and Wilson, T. SemEval-2013 task 2: sentiment analysis in twitter. *Proc. SemEval*. 2013, 312–320.
18. Pang, B., and Lee, L. Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval 1-2. 2008.
19. Pfleeger, S. L., and Caputo, D. D. Leveraging behavioral science to mitigate cyber security risk. *Computers & Security*, 31(4), 2012: 597–611.
20. Pfleeger, S. L., and Sasse, M. A. Studying usable security: how to design and conduct case study. under Review. 2016.
21. Renaud, K., Volkamer, M., and Renkema-Padmos, A. Why doesn't jane protect her privacy? *Privacy Enhancing Technologies*. 2014, 244–262.
22. Siegel, S. Nonparametric statistics for the behavioral sciences. 2nd ed. In collab. with N. Castellan. New York ; London: McGraw-Hill, 1988.
23. Stoppard, J. M., and Gruchy, C. D. G. Gender, context, and expression of positive emotion. *Pers Soc Psychol Bull*, 19(2), 1993: 143–150.
24. Strapparava, C., and Mihalcea, R. Learning to identify emotions in text *Proc. SAC '08*, 1556–60.
25. Strauss, A., and Corbin, J. M. Basics of qualitative research: grounded theory procedures and techniques. Thousand Oaks, CA, US: Sage Publications, 1990.
26. Thelwall, M., Buckley, K., Paltoglou, G., Cai, D., and Kappas, A. Sentiment strength detection in short informal text. *J. Am. Soc. Inf. Sci.* 61(12), 2010: 2544–2558.
27. Wash, R. Folk models of home computer security *Proc. SOUPS '10*, 11:1–11:16.
28. Yee, K.-P. Aligning security and usability. *IEEE S&P*, 2(5), 2004: 48–55.