



# Using Metaphor to Analyse Qualitative Data: Vulcans and Humans in Software Development

HELEN SHARP

[h.c.sharp@open.ac.uk](mailto:h.c.sharp@open.ac.uk)

*Computing Department, The Open University, Walton Hall, Milton Keynes, MK7 6AA, UK*

MARK WOODMAN

*School of Computing Science, Middlesex University, Trent Park Bramley Road, London N14 4YZ, UK*

FIONA HOVENDEN

*Pandora's Lab, San Francisco, CA, USA*

**Editor:** Marv Zelkowitz

**Abstract.** This paper reports on an experience of using metaphor in qualitative research of software engineering in practice. Our project aimed to uncover non-technical factors affecting the adoption and evolution of Software Quality Management Systems (referred to here as ‘the quality process’). Previously we have reported the tensions we uncovered around the quality process in four companies, based on semi-structured interviews. This paper extends this work by applying metaphor to the results. We show how we were able to produce more general statements regarding the tensions and their amelioration, and then introduce results from a fifth company, which we compare against our general statements. We find that these statements are generally supported by results from this fifth company. Finally we present some reflections on our experience of using metaphor in this way.

**Keywords:** Qualitative methods, metaphor, software process improvement, software quality management systems.

## 1. Introduction

The importance of non-technical factors in the success of software engineering projects has been recognised for many years (e.g., DeMarco and Lister, 1987; Curtis et al., 1988). Few empirical studies have been reported that look at the impact of non-technical factors on software development practices, but work in the area is growing (e.g., Saraph et al., 1989; Black and Porter, 1996; Baddoo and Hall, 2002b; Dyba, 2000) and the significance of studying the human aspects of software engineering through qualitative methods is receiving more attention (e.g., Seaman, 1999; Seaman and Basili, 1998; Wu et al., 2003).

The SoFEA project (Hall et al., 1993) focused on software quality management systems (SQMSs). We examined the non-technical factors affecting the adoption and evolution of SQMSs (called here ‘the quality process’). The project arose because of concerns that technical innovations did not appear to be producing better quality software systems. The project’s particular focus was on the effect of quality initiatives such as ISO9000 (2000) certification schemes, e.g., BS5750 (1987), and the social factors affecting the quality process, in response to those schemes (Hovenden et al., 1994; Sharp et al., 1999).

We performed a variety of data collection activities through our project, including semi-structured interviews with quality managers and ethnographic data about the organisations we visited. This data was analysed using discourse analysis, framed by ethnographic observations, and the results revealed some of the tensions that exist around the adoption and evolution of software quality management systems. A brief description of the original study including the companies involved and a summary of the results from the discourse analysis is given in the Appendix; more detail can be found in Sharp et al. (2003, 2004). These results resonated with findings from other researchers in a variety of fields, e.g., those of Baddoo and Hall (2002); Buono and Bowditch (1989) and Williams et al. (1993), but we wanted to examine the data from a new perspective to identify further insights about the quality process. To do this, we decided to use a novel approach that is complementary to discourse analysis. This approach is the application of metaphor to interpret qualitative data.

In Section 2 we discuss some aspects of metaphor and its uses and introduce our three-step process for applying metaphor to interpret qualitative data. In Section 3 we introduce the specific metaphor we used, its application to our area of investigation, the observations it prompted and the statements about the quality process that the observations lead us to make. Section 4 contains information about a fifth company which we then compare against these statements. Finally, Section 5 presents some reflections on our experiences.

## 2. Metaphor and Its Uses

A great deal has been written about metaphors and their uses. We hardly scratch the surface of this literature here, but some aspects of metaphors and their use are included here to explain our perspective.

Metaphors are a partial mapping from one domain to another that can highlight or characterise similarities between previously uncomparing concepts (Lakoff and Johnson, 1980). In problem solving and in analysis metaphors are used to facilitate reasoning about one, typically unfamiliar, domain through reasoning about another, typically familiar one. Many writers have commented upon the ability of metaphors to uncover the unexpected or implicit in the domain of comparison, e.g., “new slants on the <domain of interest> are illuminated.” (Hesse, 1966, p. 163), “... This may involve... the creation of new similarities...” (Lakoff and Johnson, 1980, p. 154), “... the <domain of interest> acquires new meaning through its involvement with the <metaphorical> one...” (Martin and Harre, 1982, p. 91), “... a metaphorical statement can sometimes generate new knowledge and insight...” (Black, 1979, p. 37).

Gentner (1983) suggests an approach to analysing metaphor called structure-mapping. This says that a metaphor can be viewed as a mapping of knowledge from one domain (base) to another (target) which conveys that a system of relations that holds among the base objects also holds among the target objects. This approach views metaphor as a set of relations and is also called analogical structure-mapping. The relationship between analogy and metaphor is often debated, and we discuss this below.

### **2.1. *Metaphor and Analogy***

Concern is often expressed about the difference between analogy and metaphor, and they are clearly related: both involve comparison between two domains. For example, "... traditionally the metaphor is defined as being a process of the transfer of meaning by the substitution of an analogy..." (Pesot (1975) quoted in Beck (1978, p. 86), "... every metaphor may be said to mediate an analogy or structural correspondence..." (Black, 1979, p. 31). Although authors often go to great lengths to explain how they see the difference, it is quite common for the two terms to be used interchangeably (e.g., Gentner, 1982). It seems that an analogy is more concrete and detailed, with the two domains having obvious and very close similarities, while metaphor is more literary, with the two domains being further apart. For example, comparing the way protons and neutrons move around the nucleus in an atom with our solar system is an analogy, while using the term 'black hole' for very dense masses in space from which light cannot escape is metaphorical.

### **2.2. *Using Metaphor in Analysis***

Metaphor is often discussed in the context of verbal language as a way to reveal a person's underlying thinking or understanding of a domain. For example, it has been used this way in the study of diagrams (Blackwell, 1998), in the study of mathematical thinking (Pimm, 1981), and in the study of information systems development (Hirschheim and Newman, 2001). In (Hirschheim and Newman, 2001) the authors consider metaphors believed to guide systems development, such as the battle metaphor which is commonly used by both users and developers, and they provide evidence of their use from empirical studies. They claim that symbolism used by a community (including metaphors) revolves around shared meaning, which functions as a 'social glue' and binds communities together. Using metaphor in this way is suggested in traditional ethnography as one approach to identifying cultural attributes and to understanding a community's beliefs, values and rituals (Beck, 1978; Alexander, 2002).

In the use of metaphor described above, researchers are trying to elicit metaphors from the participants under study, and then they analyse them to understand the participants' view of the world.

An alternative use of metaphor is to help explain a concept to others. For example, metaphor is used in science to explain phenomena, and sometimes to uncover the unexpected (e.g., Hesse, 1966; Beck, 1978). Metaphor has also been used extensively by the HCI community (e.g., Preece et al., 2002) to help users learn a new system or to solve problems when things go wrong. Common metaphors in HCI for personal computers include 'windows,' the 'desktop' and 'trash/recycle bin.' Finally, metaphor is commonly used in education to explain new concepts and to help students relate new information to old.

We believe that metaphor has a further use—as a way of interpreting empirical data. However, very little work has been done in this area.

Building on Gentner's structure-mapping view of metaphor described above, Aubusson (2002) has used analogical structure mapping to interpret the results from an empirical case study focusing on a school science department that was attempting

change. The metaphor he used was the ecosystem, and it was used to explain and explore the nature of science education.

Analogical structure-mapping sees metaphor as a set of relations. Although metaphor in our approach needs to emphasise relations too, i.e., the tensions between two parties or communities, we felt that it would not be appropriate to restrict our focus to this aspect alone. We therefore sought to take a different approach to using metaphor to analyse our data, one that would exploit its characteristics of creativity and explanation. To achieve this, we developed a three-step process that supports this use, called ICE:

1. identify Initial mappings between the metaphor and the domain of interest, i.e., determine whether a particular metaphor is likely to be useful;
2. seek Confirmatory mappings between the metaphor and the domain of interest, i.e., confirm that the metaphor supports existing observations about the situation under investigation;
3. use properties of the metaphor to Explore the domain of interest.

We describe more about this approach and its application in Section 5 below.

### **3. Using Metaphor to Interpret Qualitative Data**

The theme of tensions we identified led us to focus our attention on the relationship between the developers and the quality process. The previous analysis allowed us to make some statements about where tensions were alleviated, and where they were exacerbated. However, these statements were tied very closely to the context of investigation and we wanted to expand our interpretation of the data and to formulate new statements that might be more widely applied. To do this, we employed metaphor, and in particular we needed a metaphor in which relationships were central.

In this section, we describe the metaphor we chose, explain how we ‘applied’ it to further analyse our theme of tensions, and show how this led us to formulate new statements about tensions around the quality process.

#### ***3.1. The Vulcan–Human Metaphor***

The metaphor we use is based on characters in a television series. Specifically we use the relationship between Vulcans and Humans as portrayed in the Star Trek television series, and in particular the relationship between two key characters: Spock (a Vulcan) and Kirk (a Human).

In this section we describe Vulcan and Human characteristics, focusing on First Officer Spock as the key Vulcan figure in the plot, and Captain Kirk as the key human figure. Note that we do not claim that this characterisation of Human behaviour is representative nor scientifically grounded, but simply how human traits are portrayed in

this drama. We then describe the relationship between Vulcans and Humans as represented by the Spock–Kirk relationship.

### 3.1.1. *Star Trek*

The science fiction television series *Star Trek* is about a Starship called *Enterprise* and its crew, and their exploration of space. Their mission includes “scientific investigation and reconnaissance of previously unexplored worlds; providing aid and supplies for Earth colonies; diplomatic courtesy calls on alien civilisations; and the enforcement of laws regulating commerce with the Earth colonies.” (insert at p. 202 in Solow and Justman (1996)). The drama is set in the 23rd century when there is a confederation of civilisations and a large number of space stations throughout space. The series involves a variety of different lifeforms other than humans, including Vulcans (described below). The series focuses on the various adventures the *Enterprise* and their crew encounter on their mission. The series has spawned a number of successful film and further television series.

### 3.1.2. *Vulcans in Star Trek*

Vulcans used to be a passionate, violent race whose civilisation was devastated by terrible wars, until they learned to control their emotions and adopted a philosophy that embraces pure logic. Vulcan society became based entirely on logic, and anything emotional is regarded as socially unacceptable and illogical. Vulcans are thus trained from an early age to master their emotions, but not to deny them. In one episode, we are told that forcing a Vulcan to experience emotion would kill him. It is the belief of Vulcans that everything in life should revolve around logic, and that logic itself is the fundamental constant in the universe.

Spock is the First Officer and the Scientific Officer on the Starship *Enterprise*. He is always regarded as Vulcan, and mostly portrays pure Vulcan characteristics in *Star Trek*, but he is in fact half human, having a Vulcan father and a Human mother.

### 3.1.3. *Humans in Star Trek (as Represented by Captain Kirk)*

James T. Kirk is Captain of the Starship *Enterprise*, and is the key figure in the *Star Trek* series. He is highly successful, and he gained a record number of commendations from Starfleet, yet he was also the first starship captain ever to be put on trial for causing a fellow officer’s death.

Kirk is competent and can cope with anything, but he has a disrespect of authority, often ignoring Starfleet commands and those who challenge him on board his ship, even if they are higher in rank than he. He is dedicated to his ship and its crew, and is always compassionate, avoiding any harm to creatures from other planets when at all possible. Kirk often takes seemingly irrational ad hoc actions which turn out to create unmanageable situations, amenable to no rational, logical analysis. On the other hand, Kirk will and does use logic when it suits his need. He tends to focus on the immediate problem, seemingly not able to grasp all the implications of his actions, and usually turns to Spock to have them explained.

### *3.1.4. The Vulcan–Human Relationship (as Represented by the Spock–Kirk Relationship)*

Spock and Kirk work well together both in terms of commanding the Enterprise, and in terms of serving various dramatic ends. They are close friends, and it seems that Spock does not mind Kirk teasing him on occasions, e.g., trying to provoke him into displaying emotion, or commenting about his pointed ears. Their relationship does include some serious tensions, mainly based around a clash of logic and emotion, but there is also mutual respect and the tensions (in the end) are always productive. Kirk has great commitment to Spock, and vice versa. For example, both would attempt to save the other's life when in danger. For his part, Spock serves his Captain well, offering advice and guidance when asked for it, but accepting Kirk's decision even if he views it as illogical.

## **3.2. Initial Mappings to the Metaphor**

The purpose of applying this metaphor to our data is to illuminate the tensions identified in our initial analysis. But before we can do this, we need to establish some mappings between the elements of our metaphor and the subject of study. What initially attracted us to this metaphor were the parallels we saw between Vulcanism and the quality process on the one hand, and between the developers and Star Trek Humans on the other. We believe that this is a reasonable mapping on the following grounds:

### *3.2.1. Mapping Between the Quality Process and Vulcanism*

The main basis for this mapping is that both Vulcan society and the quality process are (or are portrayed as being) founded on rational logic.

The quality process was often talked about by our interviewees as though the need for an SQMS was 'clear' and 'obvious,' i.e., putting one in place was logically right and unquestionable (Sharp et al., 2003). For example, in the data we gathered for our initial analysis, there were many references to the rational language used to express quality procedures. In addition, the SQMS itself embodies a logical or rational approach to situations. It contains a set of standards, rules or guidelines for software quality which are formally expressed. It represents a rational view of the complicated world of software development.

As explained above, Vulcans revere logic, reject emotion and view the world in rational terms, no matter how complicated it may be. Other Vulcan traits that are mirrored in the intentions underlying the quality process are: reliability, loyalty, and an unbiased approach.

### *3.2.2. Mapping Between the Developers and Star Trek Humans*

A key aspect to this mapping is that Kirk can (when required) make decisions that rely on his intuition, without stopping to logically analyse a situation, but he is also competent enough to make sensible decisions. In fact, it is often this intuitive course of action that is portrayed as providing the 'winning edge.'

Software developers are often put in situations where they too need to make decisions under pressure, responding to tight deadlines and unexpected circumstances. They need to be free to use their own judgement and trusted to make sensible decisions.

### *3.2.3. Mapping Between the Spock–Kirk Relationship and the Quality Process-Developer Relationship*

We are grounding the mapping here on the tensions that exist between the two parties in each situation. In the case of Spock and Kirk, the tensions are friendly and are productive, while in the quality process situation, we have found that the tensions can often be confrontational and counter-productive. The purpose of making this mapping is to try and identify characteristics of the Spock–Kirk relationship that might provide insights into how the quality process-developer relationship might be improved, i.e., made more productive.

### **3.3. Confirmatory Mappings to the Metaphor**

The purpose of this step is to go beyond the superficial impressions that led to the initial belief that the metaphor is suitable, and to find out if the similarities are more substantial. So we need to look at the observations we already have about the situation under study and confirm that the metaphor is not in conflict with these observations.

Taking each observation in turn (indicated by italics), we can consider the metaphor circumstances and see how they relate.

Circumstances where tension appears to be present:

- *Inadequate or obviously unreasonable explanation is given by the quality process to developers regarding the need to introduce an SQMS.* Spock always explains his conclusions, even if Kirk decides to reject them.
- *There is a general lack of consultation between the quality process and developers concerning the introduction and subsequent development of the SQMS.* Spock and Kirk readily discuss decisions and situations whenever pressures allow. If the situation does not allow for discussion then they trust each other to take appropriate actions. However both Spock and Kirk recognise the importance of communication.
- *Procedures which ignore current practice are imposed with no explanation of their rationale.* Spock always gives explanations of his assessment of a situation that justifies his suggested course of action. There is no evidence from the programmes to suggest whether or not he ignores day-to-day practice, but the implication is that he is sympathetic to this.
- *Existing pride in the company is questioned or undermined by the quality process undervaluing developers' expertise.* The commitment of both Spock and Kirk to the mission of the Enterprise and all aboard is unquestionable.

- *Quality process procedures are written in a rigid fashion, ignoring the practicalities of day-to-day working.* We have no direct information about Spock's attitude or behaviour towards this aspect. He provides logical interpretations of Starfleet rules but does not rigidly enforce them.
- *Company commitment to quality falters.* Every member of the Enterprise's crew is portrayed as being a member of a strong team. This includes Kirk and Spock. Neither of these ever shows lack of commitment to each other, the ship's crew, or to their mission.

Circumstances where tension appears to be resolved or at least alleviated:

- *The need for a quality manual and for quality initiatives is accepted by all staff.* No-one on the Enterprise questions the worth of Spock and his assessments in any serious way. Other senior crew members do sometimes show exasperation with him but they still accept that what he says is valid.
- *Staff who have to implement the procedures in the quality process write them, i.e., they are trusted to 'write their own rules.'* We observed above that Kirk shows disrespect for authority and wants to operate under his own rules sometimes.
- *The manuals themselves include explanation and varying levels of insistence, or are not entirely prescriptive.* It is difficult to map this directly to the Spock-Kirk relationship, but this point is about flexibility. Both Spock and Kirk work within a flexible framework, although when the framework isn't flexible enough for Kirk, then he bends it.
- *The manual is flexible and allows decisions to be taken 'on the ground'.* Spock's assessments are logical and rational, and are therefore not flexible in themselves, but the way in which they are interpreted is flexible. He doesn't prevent Kirk from taking decisions 'on the ground.'
- *The company's commitment to quality is visible and maintained.* Spock and Kirk are always committed to the ship, its mission and its crew.

From this discussion, we can see that the metaphor is generally in tune with the situation under study. It is particularly important that we have found no conflicts. We can therefore conclude that the metaphor does indeed have a more substantial mapping with the quality process situation we are attempting to analyse, and it is worth pursuing its application in our analysis.

### ***3.4. Exploratory Mappings from the Metaphor to the Quality Process***

In the previous step, we looked for similarities between the observations from our previous analysis and the metaphor, to establish it as a viable vehicle for further exploration. We now turn to the metaphor itself and look for aspects of the Spock-Kirk

relationship that might provide insight about how to alleviate the tensions around the quality process.

The following is a set of observations that may be made about the Kirk–Spock team. We identified these through discussions among ourselves, by asking friends and family, by watching episodes of the series and by reading books and websites:

- *Kirk and Spock show mutual respect, friendship and trust*: Kirk and Spock are friends and they trust each other. For example, Kirk recognises the value of Spock’s analyses even if he decides to ignore them, and Spock accepts that the Human system of resolving problems works well, and is sometimes superior to the Vulcan logical approach. Spock is always surprised by the positive outcome to Kirk’s illogical behaviour, although he does learn from and value the experience.
- *Kirk and Spock are independent*: If Kirk is indisposed, or is off the ship, Spock takes over command. Spock can operate independently from Kirk when the situation demands it. In the television series, this situation is always temporary because Kirk always returns, but the implication is that Spock could take over command more permanently if necessary.
- *Balance of power*: Kirk can overrule Spock because he is his superior in rank, but not vice versa unless Kirk is deemed unfit by the doctor. As scientific officer, Spock has authority in scientific matters, but Kirk, as Captain, can still overrule him.
- *Balance of skills*: Kirk and Spock have complementary approaches to problem-solving. We are led to believe that neither one on their own would be so successful.
- *Spock as rule-keeper*: Spock provides information and logical interpretations of situations, and reminds Kirk when he is about to break rules or do something which Spock regards as dangerous.
- *Spock is half human*: Spock is accepting of Kirk’s ‘irrational’ behaviour. For a true Vulcan this might not be possible, but although Spock always behaves in true Vulcan fashion, his mother was a Human. We can speculate that his reaction may be more than simple tolerance. It may be that he is influenced at a deeper level because he has inherited some human traits himself.
- *Kirk’s irrationality*: Kirk’s seemingly ad hoc, irrational approach to problems often provides the winning edge. We are led to believe that without his irrationality certain situations would not be resolved as easily nor as happily.

Having identified these characteristics of the metaphor, we now use these observations to consider our original data from a new perspective. The purpose of this step is to identify new insights into how tensions around the quality process may be alleviated. We now discuss each observation in the context of the quality process and our original data, and suggest statements about the quality process that arise from the discussion.

*Kirk and Spock show mutual respect, friendship and trust.* It doesn't require us to inspect any metaphor to understand the importance of these in a successful relationship. In our previous results, there is evidence that these qualities were not present around the SQMS process where there were tensions. For example unreasonable justification, lack of consultation, and ignoring current practices all indicate a lack of respect, friendship and trust. Similar issues are reported in literature on organisational change, e.g., Buono and Bowditch (1989), Williams et al. (1993) and Deal and Kennedy (1988).

*Kirk and Spock are independent.* Interpreting this in the domain of software quality, the implication here is that the quality process should be able to operate independently from (i.e., is not dependent upon) any one individual or group of individuals. For example, the quality system would continue to fulfill its purpose even when the quality manager is not available. Another example might be that the quality manual would be comprehensive enough to cover all eventualities in the event that a particular, maybe specialist, developer is not available.

*Balance of power.* It may be too risky for developers to be able to overrule any element of the quality process. Similarly, saying that the quality process is never able to overrule developers' actions is also undesirable. In our previous work we found that tensions were alleviated when the manual is flexible and allows decisions to be taken 'on the ground.' At the same time, however, it is also important that the quality process be recognised as authoritative. In our metaphor, Kirk always has the last word, so should developers be allowed to have the last word? There are different interpretations possible here, but a key aspect seems to be that both are authoritative under certain circumstances.

*Balance of skills.* Developers are good at making competent decisions, when presented with a set of facts. Processes and documents are good at maintaining detailed records of procedures and rules. This observation is linked to the next one.

*Spock as rule-keeper.* Reminding developers about conventions, previous experience, legal obligations and so on is a key role for SQMSs, and is therefore central to the quality process. They are intended to embody the 'rules' or the correct ways of doing things, and to provide guidance for developers.

*Spock is half human.* Not only does the quality process need to be tolerant of developers, it needs to 'inherit' some of their aspirations and intentions. Exactly how this is interpreted is difficult to say, but in our metaphor, Spock is pragmatic. One conclusion may therefore be that tensions around the quality process will be alleviated if the quality process is not founded exclusively on logical, rational grounds, but integrates pragmatism as well.

*Kirk's irrationality.* This is one example where the observation is not terribly helpful in our new context, and so we abandon it.

### *3.4.1. Summary of Statements*

The discussions above lead to some new statements about how tensions can be alleviated. These are summarised below. Tensions around the quality process will be alleviated if:

1. it is operating in an atmosphere of mutual respect, friendship and trust.
2. the process is comprehensive and its operation is not reliant on any one individual or group of individuals.
3. a suitable balance of power is maintained between the process and the developers, i.e., neither one is in charge at all times, but both are seen as having authority.
4. the quality process and the developers are both entrusted to achieve what they are best suited to, i.e., the quality process should allow developers to make decisions for specific situations, while the quality process should maintain suitable rules and regulations.
5. the quality process is not founded exclusively on logical, rational grounds, but integrates pragmatism as well.

## **4. Testing the New Interpretation**

In order to test the validity of the insights identified using the metaphor, in this section we draw on new data taken from a fifth company where we spent a week working with and interviewing developers. We then relate our new statements to the information gleaned at this fifth company.

This data was collected before the metaphor analysis had been done, and so our interview questions were not related to the issues we identified, but were guided by the same concerns as the original semi-structured interviews (see Sharp et al., 2003).

In the next section we introduce the fifth company, and then we compare the five statements made above with the situation from this new context. For confidentiality reasons we refer to the company as Company A.

### *4.1. Company A and Its Quality Process*

Company A was a large government-owned organisation producing and maintaining safety-critical software. It had gone through a number of changes in the recent past and was in the early stages of preparing for stock market flotation.

We spent a week at this company; one of us stayed with the developers for the whole week, observing and interviewing them, and the other two visited for one day each, on different days. We interviewed numerous people, including the quality manager and all

staff involved in software development from different groups within the company. These groups were responsible for particular products and had very strong individual identities. The staff did not view themselves as being primarily software developers although this was an integral part of their main work.

At the time of our visit, Company A had developed their quality management system and had been using it for a while, although it was still undergoing significant changes. The SQMS was developed by an outsider—a consultant who had previously worked as a software engineer with one of the groups. He subsequently became the quality manager of the company, but it was important to him to continue to “get my hands dirty,” i.e., to work on the software which the company produces, in order to maintain his credibility with the other developers.

Initially, the introduction of the SQMS had met with resistance, although some felt that there was a definite need for a quality manual of some kind. The following quotes from three different developers illustrate the perceptions:

I think in the early days, ... it was considered these things were tablets of stone, people down there had produced them. We, in the initial stages, didn't give any input at all, they were just dumped on us. I think that galled us a bit, you know, here I am, I've been coding, oooh, in excess of twelve years, somebody comes in downstairs, writes a procedures manual, throws it at me, says do it this way and I think ... no. You know [laughing]. If you want to know how to code you ask the people who've been doing it for years. We may not have been doing it right, but we're doing it in a way that works.

I think at the time they [standards] were introduced, when there wasn't much in the way of ownership, they were sort of readily imposed. Many people didn't think much of them and ... [there were] few people who saw some merit in having some planning ... but in general I think the acceptance level was very low.

... people would admit there's an obvious need for it. I think in <Company A> in the past there was a very sort of loose culture, people tended to go away and do their own thing, without sort of communicating enough of it. There's loads of instances where people've produced software which hasn't been documented properly, they've left, and people haven't been able to use it properly since.

Company A had discovered that it is more effective to involve as many people as possible and to write procedures which directly respond to work as it is done. There was an ongoing process of adoption, involving negotiation and balance between what actually happens and what management would like to happen, but there was also a very clear idea of why this process was useful. Again, here are some illustrative quotes:

... we involved about 12 people in writing these procedures and we've got some more checking, so we probably involved a third of the department in actually preparing it. That actually was helpful because the more people get involved the more ownership there is.

... there's supposed to be documents that describe what you do and in some cases they describe what we'd like to do, but will not opt to do and you have to get that balance right as well.

One interviewee told us of arrangements for releasing software to customers that had not been subject to full quality procedures because of practical considerations. Data from other interviewees confirmed that flexibility was clearly important to the business, and the SQMS was being modified to increase its flexibility. Although there was not yet agreement on the details of the document, there seemed to be agreement on what the SQMS should achieve.

All code always goes out—well, that's not quite true, ... what we term <quality-checked> releases go out through them, but sometimes we're working with customers who need very quick bug fixes and things like that and then they go out as so-called development versions and the customer accepts that they haven't been through all the standard QA tests, or whatever.

... you get it engrained in the brain of the Project Managers what they really must do and then come up with half a page of recommendations hopefully which will be relevant to projects across the board.

... really people want to get to the stage where quality isn't something you add on at the end but is intrinsic in the way you do things.

Overall, the company seems to have experienced some tensions when the quality process first started, but the situation had improved through a variety of measures and the tensions were lessened.

#### ***4.2. Company A and the New Interpretations***

We now return to the statements made at the end of Section 3 to see how the information gleaned from Company A relates to them.

##### ***4.2.1. Mutual Respect, Friendship and Trust***

Initially, when the SQMS was “imposed” upon the developers, there was considerable resistance and distrust between the parties involved. Although the manual had been developed by someone who had worked with them in the past, the developers were wary of the quality process and had to be convinced of its worth. Hence the need to increase ownership. As a result of this early resistance, the quality manager had recognised the importance of respect because he knew that he must maintain credibility with the other developers. In fact, we found that the influence of one or two highly-regarded technical staff members was key to acceptance of the whole process. If these staff members did not support the process, then others followed their lead.

As the quality process continued, and developers became more involved, levels of trust and mutual respect appeared to be growing. For example, the developers appeared to trust the quality manager and to understand the need for what he was doing. It is worth noting though that this atmosphere had to be nurtured; it did not exist initially. The impression we had was that this had been an up-hill struggle during the early stages of the process.

#### *4.2.2. Process is Comprehensive*

The process was ongoing when we visited the company, so we have no evidence in this data relating to comprehensiveness. There was also no indication of a desire for the process to be comprehensive, just that it be practical and helpful.

#### *4.2.3. Balance of Power*

Initially, the quality process attempted to take power by telling the developers what would happen. However the developers had enough self-confidence and belief in their ability to produce good code that they were not prepared to accept such imposition and they took control. The quality process had no authority. In order for the quality process to be successful, the balance of power needed to shift so that both parties had authority. This was being achieved by further involvement of developers.

#### *4.2.4. Balance of Skill*

The quality manager learned that the manual had to support the developers, not impose things upon them. He also learned that the quality process had to give developers enough flexibility to make decisions for specific situations. This need for flexibility was recognised and accepted by all the developers we spoke to.

#### *4.2.5. Integrates Pragmatism*

We saw clear evidence of the need for and the desire to include pragmatism in the quality process. The flexibility embodied in the SQMS meant that the developer didn't feel the need to overrule the SQMS. Initially, when the manual was rigid and told developers how to behave, this caused tension, but increased flexibility resulted in decreased tension.

### **4.3. Conclusion**

Experience from this fifth company supports most of the statements made as a result of our metaphorical analysis. We can therefore conclude that the statements above have validity and are worth further investigation. Consequently, we believe that the metaphor analysis has provided us with a valid and useful alternative perspective.

## 5. Reflections on the Approach

We end this paper with some reflections on the use of metaphor described above. We believe that the application was successful, but this approach was not always easy, and we highlight here some of the main issues that we struggled with.

### 5.1. *How to Identify a Suitable Metaphor*

Identifying a metaphor for this kind of application is not simple, and we feel that it is still difficult to give concrete guidance on how to choose or even to recognise a suitable metaphor. In HCI, conceptual design includes considering whether a metaphor would be appropriate and if so, which one. Guidance here points to studying users' discourse to see what metaphors they use themselves, since the intention is to find something familiar, e.g., Erickson (1990). This would not be suitable for exploiting metaphor's creative property, as in this case a metaphor that is 'orthogonal to' rather than 'arises from' the area of study is more likely to provide a different perspective.

In our case, the choice of this metaphor arose serendipitously: one of us was attending an event for fans of the series with a younger member of their family, and was talking about this at one of our project meetings. Through the discussion, we noticed a series of parallels between the data we had been analysing and the situations portrayed in Star Trek.

However, one dimension that seems to be significant when choosing a metaphor to use in this way is the notion of 'distance' between the domain of interest and the domain of the metaphor. For example, a different but similar situation where management standards were being introduced into an organisation, such as in the education sector, would not have provided the necessary 'distance' from our situation. This would then be more of an analogy. Analogous comparisons can also shed light, but the kind of insights would have been different. For us, the key thing was to concentrate on the characteristic we wanted to investigate—relationships and tensions—and then to find a situation which had sufficient similarities.

It is also unlikely that you will be able to find a metaphor that is helpful across the whole domain. For example, in our case, the aspects which are not covered by the metaphor include the internationalization of standards, the relationship between the quality process and the organisation's customers, and the different processes required by different businesses. Our metaphor says nothing about these aspects, but we would have chosen a different metaphor if we wanted to highlight those things.

### 5.2. *What Maps to What?*

Having identified what may be a suitable metaphor, one of the challenges then is to scope the necessary mappings. In our discussions, we spent a lot of time trying to pin down the metaphor mapping to precise elements of the two domains. For example, at one point we were considering whether Spock should be mapped to the quality manager, and then we could bring in Starfleet Command as the quality manual (or maybe the book of Starfleet rules and commands?). This led to quite complicated and confused con-

versations about the details of the two situations. In the end, we pulled back from such detailed mappings and used instead a more abstract mapping, of the ‘quality process’ with ‘Spock’. At times this felt awkward because Spock is a specific person, while the quality process includes different people, documents, events and so on. However this difficulty was also a strength because it allowed for flexibility in interpretation within our domain of interest.

It is not necessary for every aspect of a metaphor to map well to the domain of interest. Areas where a metaphor breaks down or simply raises issues regarding its suitability can also be of real use. For example, the superiority in rank of Kirk over Spock seems inappropriate if it is assumed that some or all developers should always have authority over the quality process. But this leads us then to consider the possibilities around this suggestion.

It is important to remember that metaphors are intended to be a partial mapping, since they both hide and highlight certain aspects of the phenomenon they are refiguring (Lakoff and Johnson, 1980). The fact that the metaphor does not sensibly map completely to the domain of interest is not a problem. Any metaphor is only useful insofar as it provides different perspectives. Once the new analysis has been done, the metaphor can be abandoned anyway.

This fluidity can be difficult to accept because of the level of freedom it supports. However as software engineers, we can bring to bear a lot of knowledge and experience of the domain throughout the analysis process, and thus produce solid results that have been interpreted within an appropriate framework.

### **5.3. *Making Useful Observations***

The list of observations we made about Star Trek and its characters was clearly not exhaustive. For example there are aspects of the Vulcan/Human diversity that we did not dwell upon here, such as their physiological similarities and differences. Highlighting these would not have helped our discussion, and it would have taken considerable time to factor these comparisons into the metaphorical mapping.

We did consider a wider variety of observations than those reported here, but we have presented the main characteristics as identified from a variety of sources. In this case, we only abandoned one observation that seemed to be irrelevant to the quality process (the fact that Kirk could be irrational).

### **5.4. *Using ICE***

We did not initially set out to use this three-step process. It emerged as the analysis progressed. However, once these steps were identified, it provided clarity to our process and enabled us to separate concerns about the mapping’s suitability from the new insights that we thought we could gain. We have considered whether to combine the first two steps, but feel that it is useful to keep them both because it emphasises the difference between initial impressions and a deeper suitability, and we contend that it is important to identify confirmatory mappings and not to rely just on initial impressions.

The exploratory stage involves interpretation and application of domain knowledge. Not everyone would draw the same conclusions that we did, but group discussion helps to moderate any extreme suggestions. Because of this, we would recommend that this technique be used by a group rather than by a single researcher.

## 6. Conclusions

The use of metaphor-based analysis is an innovation in qualitative analysis methods. We believe that using a metaphor has provided us with a way to place a different perspective on the situations we've identified, specifically allowing us to abstract from the individual instances we observed and present our findings in a more generic way. For example, instead of concluding that tensions are alleviated where the quality manual includes explanation and varying levels of insistence, which is a specific solution, we can say that tensions are alleviated where pragmatism is incorporated into the process. Exactly how an organisation may choose to apply this suggestion is up to them and their particular environment.

We then considered conclusions from a further company to explore whether these more generic statements had any support in this new situation. We found that in general they did.

Field studies are often criticised for being too reliant on context, and therefore ungeneralisable. We have described a case where applying a metaphor to a set of field study situations has resulted in the development of more general statements that have been shown to be applicable to a new situation. We therefore conclude that using metaphor to interpret qualitative data can give rise to new insights that may be more generalisable than the original individual observations.

## Acknowledgments

We would like to thank all those who took part in our studies for their time and patience. The work reported on here was partly funded by The Open University's Research Development Fund.

## Appendix: A Summary of the Original Study

Below, we briefly describe the background to the original study, each of the four companies involved, and a summary of the overall results. Further details are in the papers Sharp et al. (2003, 2004).

### *A.1. Software Quality Initiatives*

A variety of software quality initiatives have arisen from the works of gurus, such as Deming (1982); Juran (1988); and Crosby (1979). For example, certification schemes

such as BS5750 (1987) and ISO9000 (also known as ISO EN BS 9000), emerged. Other, more explicit models have also been developed, such as the Capability Maturity Model (Paulk et al., 1997) and later the People CMM (Curtis et al., 2002) and CMMI, SPICE (Khaled El Emam et al., 1997) and Total Quality Management (Oakland, 1994).

A 'quality system' (SQMS) is "The organisational structure, responsibilities, procedures, processes and resources for implementing quality management" (ISO8402, 1994). It is intended to make the process of software development visible and public. This is partly about accountability, but, more pragmatically, makes knowledge (especially that relating to the practice of developing software) communal, minimizing the problems arising when team members are away, or have left. It also helps to make the process visible to those who have not been involved in the development, such as senior managers, auditors and customers. An SQMS makes the process auditable because it provides guidelines, or benchmarks which can be tested against, and it provides some assurance of adherence to house style. This fulfills a range of requirements, including indicating to customers that certain standards are being met, strengthening a market image by 'branding' through conformance to certain standards. Hence the introduction of an SQMS can reinforce management control as well as enforce internationally established guidelines.

## ***A.2. The Companies Involved***

### *Company A*

Company A provided computing services and equipment for scientists to use in the field. At the time of the study, Company A was undergoing some restructuring and relocation. They had just started to develop their SQMS, and a group of engineers, including our interviewee, were writing down the procedures already in operation. No quality manager had been appointed, nor was there any evidence that one would be. Our main interviewee was a management-level employee involved in the development of the SQMS.

From analysis of Company A's data, we made the following conclusions. The SQMS was being developed because of a belief at the higher management level that commercial pressures required that the organisation be certified to ISO9000. However, the staff had a strong feeling of pride in the service that they offered, and the implication that they needed certification to prove quality was resented, and the introduction of an SQMS was seen as an imposition 'from above.' The current activity of writing down existing procedures was seen as unproblematic in itself, but there was concern about what would happen when certification requirements that the company did not meet were identified (which was viewed as inevitable).

### *Company B*

Company B was a large avionics organisation that had a long-term involvement with the defence industry. Company B's SQMS had just started to be used. Our interviewee was responsible for quality procedures within Company B.

Company B had a variety of quality standards and due to its involvement in the defence industry, had been conforming to such legislation for many years. However, a unified SQMS was developed because of a need to conform to ISO9000. The staff felt that there had been a lack of consultation in the development and that the new quality system was being imposed upon them. Posters exhorting the staff to improve quality were regularly defaced. Staff felt that they knew what was needed to improve quality, but their opinions had not been sought. Initially, the quality manual was particularly difficult to read and use, and its relevance to the engineers who were intended to implement it was not recognised or explained. After several iterations, the final document included different levels of direction. The mandatory information, phrased as “this shall happen in these circumstances,” was accompanied by comments that explained the rationale behind it.

#### *Company C*

Company C was a large software house with international links. Its SQMS was mature and had been operating for many years. Our interviewee was the quality manager for one of the company’s sections. The company chairman was committed to quality and exerted great influence on staff.

Company C’s SQMS emerged over many years from a document that captured useful information, to a more formal quality regime that was introduced later on. The SQMS outlines the implementation of quality management at a high level of abstraction, and leaves the details to staff in the specific areas. There is a clear belief that staff need to be involved in developing the SQMS because they understand best how to do the job, and they need to be free to do it. Staff are encouraged to make suggestions for quality improvements in their part of the company. The quality manual should also reflect actual practice, rather than idealized intentions. Any new procedure which is suggested is trialled first, and if it is successful, it can become part of the quality manual.

#### *Company D*

Company D was a subsidiary of a large, non-UK company. It was the only subsidiary of this company whose main business was software development. Company D had a mature SQMS which had been operating for many years. We interviewed the subsidiary’s quality manager who had been in the company for 16 years.

Company D’s SQMS was developed iteratively and was continually improved. The SQMS was written by staff ‘on the ground’ because ownership was viewed as important. Because Company D were the only software house owned by this company, there were many influences brought to bear on the development of their quality procedures. The SQMS currently in use at the time of the study was divided into prescriptive procedures, and guidelines. Our interviewee emphasised the need to keep a high profile for quality, the importance of encouraging an attitude so that quality never slips and of maintaining a long term view of the company’s business.

### *A.3. Tensions Around the Adoption and Evolution of SQMS*

The data used in our initial analysis was explored using discourse analysis and one of the themes we identified was that of tensions around the quality process (see Sharp et al. (2003) for more details.) Specifically, we found that tension appears to be present where any or all of the following are true:

- Inadequate or obviously unreasonable explanation is given to staff regarding the need to introduce an SQMS;
- There is a general lack of consultation with staff concerning the introduction and subsequent development of the SQMS;
- Procedures which ignore current practice are imposed with no explanation of their rationale;
- Existing pride in the company is questioned or undermined by undervaluing expertise;
- Procedures are written in a rigid fashion, ignoring the practicalities of day-to-day working;
- Company commitment to quality falters.

Tension appears to be resolved or at least alleviated where any or all of the following are true:

- The need for a quality manual, and for quality initiatives, is accepted by all staff;
- Staff who have to implement the procedures write them;
- The manuals themselves include explanation and varying levels of insistence, or are not entirely prescriptive;
- The manual is flexible and allows decisions to be taken 'on the ground';
- The company's commitment to quality is visible and maintained.

### **References**

- Alexander, C. E. 2002. 'One of the boys': Black masculinity and the peer group. In S. Taylor, (ed.), *Ethnographic Research: A Reader*. London: Sage Publications Ltd.
- Aubusson, P. 2002. Using metaphor to make sense and build theory in qualitative analysis. *The Qualitative Report* 7(4): [www.nova.edu/ssss/QR](http://www.nova.edu/ssss/QR).
- Baddoo, N., and Hall, T. 2002a. Motivators of software process improvement: An analysis of practitioners' views. *Journal of Systems and Software* 62: 85–96.

- Baddoo, N., and Hall, T. 2002b. Software process improvement motivators: An analysis using multidimensional scaling. *Empirical Software Engineering* 7: 93–114.
- Beck, B. E. F. 1978. The metaphor as a mediator between semantic and analogic modes of thought. *Current Anthropology* 19(1): 83–97.
- Black, M. 1979. More about metaphor. In A. Ortony, (ed.), *Metaphor and Thought*. Cambridge: Cambridge University Press, pp. 19–43.
- Black, S. A., and Porter, L. J. 1996. Identification of the critical factors of TQM. *Decision Sciences* 27(1): 1–21.
- Blackwell, A. F. 1998. *Metaphor in Diagrams*. University of Cambridge.
- BS5750 1987. *Parts 1–3 Quality Systems*, HMSO.
- Buono, A. F., and Bowditch, J. L. 1989. *The Human Side of Mergers and Acquisitions*. San Francisco: Jossey-Bass Publishers.
- Crosby, P. B. 1979. *Quality is Free*. New York: McGraw-Hill.
- Curtis, B., Krasner, H., and Iscoe, N. 1988. A field study of the software design process for large systems. *Communications of the ACM* 31(11): 1268–1287.
- Curtis, B., Hefley, W. E., and Miller, S. A. 2002. *The People CMM: Guidelines for Improving the Workforce*. Wesley: Addison-Wesley.
- Deal, T., and Kennedy, A. 1988. *Corporate Cultures: The Rites and Rituals of Corporate Life*. London: Penguin.
- DeMarco, T., and Lister, T. 1987. *Peopleware: Productive Projects and Teams*. Dorset House Publishing.
- Deming, W. E. 1982. *Out of the Crisis*. Cambridge MA: MIT Press.
- Dyba, T. 2000. An instrument for measuring the key factors of success in software process improvement. *Empirical Software Engineering* 5: 357–390.
- Erickson, T. D. 1990. Working with interface metaphors. In: B. Laurel, (ed.), *The Art of Human-Computer Interface Design*, Boston: Addison-Wesley, pp. 65–73.
- Gentner, D. 1982. Are scientific analogies metaphors? In: D. S. Miall, (ed.), *Metaphor: Problems and Perspectives*, Brighton: The Harvester Press, pp. 106–132.
- Gentner, D. 1983. Structure-mapping: A theoretical framework for analogy. *Cognitive Science* 7(2): 155–170.
- Hall, P. A. V., Sharp, H. C. and Woodman, M. 1993. What is behind software quality management systems? In M. Ross, C. A. Brebbia, G. Staples, and J. Stapleton, (eds.), *Software Quality Management*. Southampton, UK: Computational Mechanics and Elsevier, pp. 225–232.
- Hesse, M. B. 1966. *Models and Analogies in Science*. Notre Dame, Indiana: University of Notre Dame Press.
- Hirschheim, R., and Newman, M. 2001. Symbolism and information systems development: Myth, metaphor and magic. *Information Systems Research* 2(1): 29–62.
- Hovenden, F. M., Yates, S., Sharp, H. C., and Woodman, M. 1994. The use of ethnography with discourse analysis in the study of software quality management systems. In: M. Ross, C. A. Brebbia, G. Staples, and J. Stapleton, (eds.), *Software Quality Management II*. Southampton, UK: Computational Mechanics Publications, pp. 557–572.
- ISO8402 1994. *Quality Vocabulary*.
- ISO9000 2000. *Quality Management*.
- Juran, J. M. (ed.) 1988. *Quality Control Handbook*. New York: McGraw-Hill.
- El Emam, K., Drouin, J.-N., and Melo, W. (eds.), 1997. *SPICE: The Theory and Practice of Software Process Improvement and Capability Determination*. London: John Wiley.
- Lakoff, G., and Johnson, M. 1980. *Metaphors we Live by*. Chicago and London: The University of Chicago Press.
- Martin, J., and Harre, R. 1982. Metaphor in science. In: D. S. Miall, (ed.), *Metaphor: Problems and Perspectives*. Brighton: The Harvester Press, pp. 89–105.
- Oakland, J. S. 1994. *Total Quality Management*. Oxford: Butterworth-Heinemann Ltd.: Oxford.
- Paulk, M. C., Curtis, B., Chrissis, M. B., and Weber, C. V. 1997. Capability maturity model for software. In: M. Dorfman, and R. H. Thayer, (eds.), *Software Engineering*, IEEE.
- Pesot, J. 1975. Un probleme de semiologie linguistique: Mobilite du signe et monosemie profonde. *Canadian Journal of Research in Semiotics* 3: 5–15.
- Pimm, D. 1981. Metaphor and analogy in mathematics. *For the Learning of Mathematics* 1(3): 4–50.
- Preece, J., Rogers, Y., and Sharp, H. 2002. *Interaction Design: Beyond Human-Computer Interaction*. New York: John Wiley & Sons Inc.

- Saraph, J. V., Benson, P. G., and Schroeder, R. G. 1989. An instrument for measuring the critical factors of quality management. *Decision Sciences* 20(4): 457–478.
- Seaman, C. B. 1999. Qualitative methods in empirical studies of software engineering. *IEEE Transactions on Software Engineering* 25(4): 557–572.
- Seaman, C. B. and Basili, V. R. 1998. Communication and organization: an empirical study of discussion in inspection meetings. *IEEE Transactions on Software Engineering* 24(7): 559–572.
- Sharp, H., Robinson, H. M., and Woodman, M. 1999. *The Role of Culture in Successful Software Process Improvement*. Milan, Italy: EUROMICRO '99, pp. 170–176.
- Sharp, H., Hovenden, F., and Woodman, M. 2003. Tensions in the adoption and evolution of software quality management systems. In *Proceedings of PPIG/EASE 2003*, UK: Keele, pp. 297–312.
- Sharp, H., Woodman, M., and Hovenden, F. 2004. Tensions in the adoption and evolution of software quality management systems: A discourse analytic approach. *International Journal of Human Computer Systems* 61(2): 219–239.
- Solow, H. F., and Justman, R. H. 1996. *Inside Star Trek: The Real Story*. New York: Pocket Books.
- Williams, A., Dobson, P., and Walters, M. 1993. *Changing Culture: New Organisational Approaches*. 2nd edition. London: Institute of Personnel Management.
- Wu, J., Graham, T. C. N., and Smith, P. W. 2003. In: *A Study of Collaboration in Software Design*, International Symposium on Empirical Software Engineering, Rome, Italy, pp. 304–313.



**Helen Sharp** is a Senior Lecturer in the Computing Department of the Open University, and a Senior Visiting Fellow at the Centre for HCI Design at City University, London. She is also co-director of the Department's Centre for Empirical Studies of Software Development. Since working as a developer in the early 1980s, Helen has been keen to understand software practice, i.e., how software is developed in reality, and thereby to inform the development of better support for software engineers. This has led her to look at a variety of research areas, from software design support environments to ethnographic studies of developers, and from process modelling to quality assurance. Throughout her investigations, she focuses consistently on the people and their social interactions, rather than on the technologies available and where to apply them. She received a BSc in mathematics and an M.Sc. and Ph.D. in computer science from University College London. She is an affiliate of the IEEE Computer Society, and a member of the British Computer Society, the Engineering Council and the ACM. She is also a chartered engineer.



**Mark Woodman** is founder and head of Middlesex University's e-Centre that researches and consults on all aspects of e-technology and e-strategy. His current work is primarily to do with the alignment of strategy and technology solutions in organisations, both large and small. This encompasses the way, for example, that complex web-based software systems are produced, deployed and managed. One area of study concerns how end-clients engage with agile, incremental approaches to both strategy and software development. Another area is the reorganisation of businesses around software solutions, such as those of Web Services technologies. In his strategic work he routinely uses tools such as discourse analysis and metaphor in the ethnographic study of people and organisations.



**Fiona Hovenden, Ph.D.** is an organizational consultant specializing in the use of ethnography and psychology to understand contemporary business and technical cultures. Her interests include: work and research processes; innovation and creativity; the roles and influences of belief and value systems in the creation of products and services; the relationships between producers and users as mediated by a product or service; organizational cultures in transition; working with implicit and explicit hierarchies; and the analysis of online experience. Based in San Francisco, Fiona can be reached at [fiona@pandoras-lab.com](mailto:fiona@pandoras-lab.com).