Views of software quality: a field report

T. Hall
D. Wilson

Indexing terms: Software quality

Abstract: The authors present findings from a recent study into the quality views of software practitioners in five UK companies. The study explored how issues of software quality have affected grassroots practitioners in ordinary companies; in particular the typical quality experiences of practitioners and what practitioners really think about quality initiatives. It is only by listening to the experiences and views of ordinary practitioners that truly effective approaches to quality can be developed. Indeed various ways are recomended in which the results presented can be used to improve the effectiveness of quality initiatives. One of the main findings is that managers and developers are keen to see software quality improved, and are positive about ways in which that can be achieved. Developers are more enthusiastic than is usually believed to see quality formalism in software development. Developers in almost all of the companies in the study said they wanted a more formal approach to quality; developers in the companies with the least quality formalism were most keen to see it implemented. Although there has been significant progress in the field of software quality, it is shown that this has not yet filtered through to ordinary companies. Where it has, there is evidence of an overzealousness and dogma that have turned practitioners off quality altogether. The result is that many practitioners are frustrated about poor quality but feel they lack the power and information to do anything about it. Indeed, many practitioners believe that the market wants cheap software quickly and is not too concerned about its quality.

1 Introduction

The industry-wide imperative to substantially improve software quality is now well documented. Disasters such as the London Ambulance System [1] amply illustrate this imperative. Solutions put forward to address this quality deficit are usually either technical or organisational. In the last few years we have seen technical quality initiatives such as CASE tools and organisational initiatives such as the capability maturity model [2]. We suggest that whether a quality initiative is technical or organisational, successful implementation is dependent on underlying practitioner attitudes to quality. (We use the term 'practitioner' to refer to the undifferentiated grouping of developers and middle managers. We exclude senior managers from this group.)

There is very little evidence of practitioner views being sought or incorporated into either industry-wide or organisation-specific quality initiatives. However, we believe that the attitudes of individual practitioners directly influence software quality. The attitude of an individual practitioner to software quality, and to the use of specific quality mechanisms, will dictate that practitioner's behaviour towards quality. This behaviour will, in turn, have a direct impact on the quality of the software he or she produces. The impact of practitioner attitude is frequently overlooked during the development of new quality initiatives. This deficit probably contributes significantly to the failure of many quality initiatives. We argue that understanding the attitudes of practitioners to quality is a precursor to implementing successful quality initiatives. In this paper we report on a recent study that explored the attitudes of grassroots software practitioners to quality in five UK companies.

We visited companies and listened to a variety of ordinary software practitioners with the specific aims of

(i) Establishing what their typical quality experiences are.

(ii) Identifying their major concerns about quality.

(iii) Exploring their attitudes to, and experiences of, quality initiatives.

The paper is structured as follows: Section 2 outlines methodology, Sections 3, 4 and 5 present the results, Section 6 provides discussion and analysis and Section 7 summarises the main conclusions and recommendations from the study.

2 The study

2.1 The companies

Five UK companies participated in this study. Three of the five companies responded to a request for participants which we posted via the internet to the BCS Software Quality discussion group. The remaining two were recommended to us and we approached them directly. Predictably, the three 'self-selecting' companies turned out to have a more mature approach to quality

IEE Proceedings online no. 19971072
Paper received 4th November 1996

T. Hall is with the School of Computer Science, University of Westminster, 115 New Cavendish Street, London W1M 8JS, UK
D. Wilson is with the School of Computing Sciences, University of Technology, Sydney, PO Box 123, Broadway, NSW 2007, Australia

Table 1: Company profiles

Company	Company characteristics			Software function size	Main software work (ranked)	Application type
	type	age	size			
A	Software House	6 yrs	120 staff	120 staff	1. Development 2. Maintenance 3. Support	• Business information systems. • External customers
B	Software House	6 yrs	24 staff	24 staff	1. Development 2. Maintenance	• Technical, real time business systems. • External customers
C	Multinational, electronic goods producer	Over 100 yrs	200 staff in the participating division	50 staff in the participating division	1. Development 2. R & D	• Embedded, control software. • External customers
D1	Large, national membership organisation	100 yrs	3,500 staff	75 staff	1. Development 2. Maintenance 3. Support	• Technical, operations support software. • Internal customers
D2	Large, national membership organisation	100 yrs	3,500 staff	75 staff	1. Maintenance 2. Support	• Business and telecom systems. • Internal customers

Companies D1 and D2 are two autonomous divisions of the same company

than the two approached directly. Although we do not claim that this is a statistically representative sample, Table 1 does show that the companies in the study range from very small software houses to very large multinational companies and cover a wide range of application areas. However, we do caution that our small sample makes generalising these results to the whole software industry statistically untenable.

The companies in the study also had a range of quality maturity levels. Table 2 shows that company A had most quality maturity, having had TickIT accreditation for 2 years. Company B was least mature, and had no quality programme at all.

Table 2: Quality context

Company	Internal software quality system		Certified software quality system		Plans for software quality certification?
	yes/no	age	type	age	yes/no
A	yes	5 yrs	TickIT	2 yrs	n.a.
B	no	n.a.	none	n.a.	no
C	yes	2 yrs	ISO9001[1]	1 yr	no
D1	yes	1 yr	none[2]	n.a.	no
D2	yes	4 yrs	none[2]	n.a.	no

[1] The processes used to design and produce the final products (consumer durables) are ISO9001 certified, but the development of the embedded software is not specifically certified.
[2] Some functions within company D are ISO9001 certified. Software development is not.

2.2 Methodology

We conducted the study over a three month period in late 1995. We spent a day at each company where we discussed quality with a cross-section of practitioners. These practitioners were carefully selected, usually by quality managers, to reflect a variety of skills, experience and quality familiarity levels. In particular we:

(i) Collected demographic information on the company. We did this via a short meeting with the quality manager. We collected mainly objective, context setting information during this meeting.

(ii) Discussed quality with a group of developers. We did this via an informal discussion with peer groups of between four and six software developers.

(iii) Discussed quality with a group of managers. We held a separate discussion with peer groups of three or four managers. The managers were mainly middle managers, although senior management was represented in the session at the smallest company (company B).

During the research sessions we prompted each group of developers and managers to discuss:

(i) The quality of their company's software.

(ii) How software quality could be improved in their company.

(iii) Whether they thought that their company's quality systems were effective.

(iv) What they thought about quality certification.

3 Industry-wide issues

In this Section we present the issues that arose in the group sessions across all the companies in the study. Some issues arose only in the manager sessions; other issues only in developer sessions, but some issues arose in both manager and developer sessions.

3.1 Issues arising in both manager and developer sessions

3.1.1. Positive about quality improvement: All the practitioners in this study had positive attitudes towards quality and all were keen to see software quality in their company improved. In addition, all said they were amenable to ways in which improvement could be achieved. Group norms about quality were also quick to manifest themselves at each company. These norms were generally company-specific and depended on the company's particular approach to quality (company-specific issues are discussed in Section 4).

3.1.2 Growth in formalism: In almost all of the companies in this study the amount of quality formalism in the software development process was perceived to be steadily increasing (this was not the case in company B, and the reasons for this are discussed in Section 4). This increased formalism was in the use of more standards, procedures, methods, metrics etc. On the whole, practitioners were positive about this growth in formalism. They were particularly positive about formalism that they perceived to be reflecting and disseminating existing good practice. Indeed, developers in company C said that their quality procedures had simply formalised the way in which they knew they should be doing things anyway. Practitioners generally felt that the formalism they had been involved in developing, and which was implemented slowly and sensitively, did not meet with resistance.

3.1.3 Impact of experience: Practitioners with the most experience of working within quality systems were generally most positive about quality formalism. Ignorance about what it is to work within a quality system seemed to make practitioners fearful and resistant.

3.1.4 Pressures on quality: Managers and developers alike were very concerned about the impact of tight deadlines and diminishing resources on software quality. There was a strong feeling that the current commercial climate encouraged companies to take a short-term view of quality. Practitioners thought that most companies (and their customers) considered the quickest and cheapest solution to be the best solution. Practitioners were particularly frustrated that tight schedules led to the kind of mistakes which compromised quality. They also said that quality formalism 'went out of the window' when deadlines loomed, and that in their experience, this always meant more rework in the long-term.

In some companies, practitioners felt that they also worked under an implicit job threat. In these companies, practitioners were led to believe that if they could not deliver software on time then they would be replaced by practitioners who could. These problems were particularly acute in company D where practitioners said that senior management insisted that software was delivered on time, believing that quality problems should be sorted out after delivery. Practitioners' general perception was that despite companies *saying* that quality was important, companies *actually* valued deadlines much more than quality.

3.1.5 Need for flexibility: All practitioners stressed the need for flexibility within quality systems. Practitioners within companies with mature quality systems stressed how the success of those systems depended on the flexibility inherent in them. Practitioners in the companies which had less quality maturity thought that quality systems were not able to provide the necessary flexibility. Practitioner perceptions of quality flexibility were related to their levels of quality experience. Many practitioners who had experience of comprehensive quality systems said that they had been surprised at how flexible those systems could be in reality (they too had expected them to be more rigid). Many practitioners were quite ignorant about how unprescriptive quality systems can be.

3.1.6 Dangers of bureaucracy: Practitioners all commented on the dangers of introducing unproductive bureaucracy in the name of quality. Again, views fell into two camps: those practitioners with experience of quality systems and those without. Most practitioners with experience of working in a quality system said that, although they had worried about the prospect of excessive bureaucracy, their quality system had not proved to be full of 'red-tape'. However, one or two of the experienced practitioners had known highly bureaucratic quality systems that had made them very negative about quality systems generally. Practitioners without experience of quality systems were much more concerned about bureaucracy. Many were convinced that the smallest task would become a bureaucratic nightmare. This perception was particularly acute in company B, where practitioners seemed to have a pathological hatred of anything to do with quality formalism. This was taken to the extreme of a senior manager with the specific brief of 'running interference' on a major customer's quality system. Practitioners in most of the companies felt that some companies had taken quality formalism too far, and that this had scared some companies off quality altogether.

Table 3 provides a summary of the spread of the general issues discussed above across the companies in the study.

Table 3: Occurrences of industry-wide issues across the companies

	Issue occurences across the companies				
Issues	A	B	C	D1	D2
1. Positive about quality improvement	✓	✓	✓	✓	✓
2. Growth in formalism	✓	✗	✓	✓	✓
3. The impact of experience	✓	✗	✓	✓	✗
4. Pressures of quality	✓	✗	✓	✓	✓
5. The need for flexibility	✓	✓	✓	✓	✗
6. The dangers of bureaucracy	✓	✓	✗	✓	✗

3.2 Issues from manager sessions

3.2.1 Top-down pressures: Middle management generally felt under pressure from senior management to improve quality. However, the managers also felt that senior management did not fully understand the difficulties of software development. This was a particular issue in companies whose primary product was not software. In some companies, such as company D, the managers felt that senior management had a 'band wagon' approach to improving quality. The managers in company D described how senior management had half-heartedly implemented a sequence of different quality 'silver bullets' (most of which were subsequently abandoned). The managers in one or two of the companies were concerned that the approach of senior management had made practitioners cynical about all quality initiatives.

There was a general perception that, although senior management were keen to be seen implementing quality initiatives, they were only really interested in software being delivered on time and in budget. Middle managers thought that delivering high-quality products was the secondary concern of senior managers; delivery on time was perceived to be much more important to senior managers. There was a general feeling that senior managers only really paid lip service to quality.

3.2.2 Bottom-up pressures: Middle managers also felt that they were under pressure from developers about quality. Managers in three companies said that developers were keen to see software development improved and were active in advocating ways of improving quality. Managers also said that grassroots quality improvements were the most long lasting and successful.

3.2.3 Quality empowerment: A recurrent theme in the sessions with managers was their attempts to create a quality culture. Managers said that their priority was to create a grassroots quality culture where individual developers took responsibility for the quality of their software. In some companies (company A in particular) managers emphasised the importance of a 'no-blame' ethos. Managers in company A were trying to create a culture where people could be open about their mistakes so that quality could be improved in the long term. Managers also thought that professionalism was highly significant to software quality. This meant that managers viewed developers as responsible for the quality of the software that they personally produced. Indeed managers and developers both felt strongly that good quality software was ultimately dependent on good developers.

3.2.4 Quality infrastructure: Managers in some of the uncertified companies felt strongly that it was important to get the quality infrastructure right before thinking about quality certification. They were aiming to develop a quality system which first and foremost improved quality and only when that was working effectively to think about having the system certified. They thought that too many companies had gained quality certification with minimal procedural change, but no real quality culture. They also suspect that these companies did not have quality systems that improved quality.

3.2.5 Information and guidance: Managers felt strongly that there was not enough information and guidance on implementing quality systems. This was a particular issue in the two companies (A and C) with the highest quality maturity. Managers in company A felt that this lack of proper information and support had initially led them to implement an overzealous quality system in their attempt to be sure of achieving certification.

Table 4 provides a summary of the spread of the manager issues discussed above across the companies in the study.

Table 4: Occurrences of manager issues across the companies

	Issue occurences across the companies				
Issues	A	B	C	D1	D2
1. Top down pressure	✗	✗	✓	✓	✓
2. Bottom up pressure	✓	✗	✓	✓	✗
3. Quality empowerment	✓	✗	✓	✗	✗
4. Quality infrastructure	✓	✗	✓	✗	✓
5. Information and guidance	✓	✗	✓	✗	✗

3.3 Issues from developer sessions

3.3.1 More formalism: Developers in almost all companies said they wanted a more formal approach to quality. The exceptions to this were company B, where an antiquality culture existed, and company A, which already had a certified quality system and where developers were generally happy with existing levels of formality. Developers in the companies with the least quality formalism were most vocal in their desire for more. Developers' craving for more formality was illustrated when developers in company D were enthusiastic about their company recently introducing timesheets.

Developers said they wanted more quality formality because they felt that:

(i) They wanted to know the 'right' way of doing tasks.

(ii) Explicit standards and procedures would enable them to show managers exactly what is involved in producing a new system. They felt that this would put them in a stronger position to argue for realistic schedules and resources. Developers in company C described formalism as 'a good stick to beat managers with'.

(iii) If things went wrong they would have more protection from blame if they could show that they had followed the appropriate procedures.

(iv) New staff could be integrated into the company more easily.

Developers said they were disappointed at how slow companies are to formalise and disseminate good practice. There was disappointment in company D that senior management were not keen to nurture the grassroots quality initiatives that developers valued. Developers in company D thought that their senior management were most keen on formalism that had little real value. They perceived that senior managers had a 'band wagon' approach to quality formalism and that had led to dogma-driven quality initiatives. Developers in company D also said that they would like to see a more formal approach taken to senior management planning and decision-making, as this was perceived to have very little transparency currently.

The general developer consensus on formalism was that, although developers were keen on quality formalism, they were not often keen on the insubstantial formalism that senior management usually introduced.

3.3.2 Appropriate levels of detail: Developers felt that it was important to be aware that formalism can quickly become outdated and that a quality system needs constant maintenance. Developers felt that maintenance was often neglected and that this led to quality systems degrading.

3.3.3 Quality system evasion: Developers felt that some of the procedures within quality systems were too easy to evade (including developers in the companies with most quality maturity). Indeed there was a general feeling that when 'the going got tough' managers condoned quality system evasion.

3.3.4 Threat to creativity: Very few developers in the study mentioned conflicts between quality formalism and creativity. Only in company B was it considered an issue, although a manager in company C mentioned it as relevant to R&D developers.

Table 5 provides a summary of the spread of the developer issues discussed above across the companies in the study.

Table 5: Occurrences of Developer Issues across the Companies

	Issue occurences across the companies				
Issues	A	B	C	D1	D2
1. More formalism	✗	✗	✓	✓	✓
2. Appropriate levels of detail	✓	✓	✗	✗	✗
3. Quality system evasion	✓	✓	✓	✓	✗
4. Threat to creativity	✗	✓	✓	✗	✗

4 Organisation-specific issues

4.1 Opposition to quality

Company B had a company culture which was diametrically opposed to any quality formalism. This company had many manifestations of a classic CMM [2] level one company. It was a very small software house headed by a charismatic managing director who believed that quality depended only on the calibre of the developer. He claimed to produce good quality software because he only employed highly creative able developers who were not restricted by any bureaucracy (in an attempt to meet this end many of the developers he recruited were physics rather than computing graduates). He felt that quality systems depress and demotivate highly creative people.

Practitioners in this company were vitriolic in their opposition to any quality formalism, perceiving it only as a restriction on their creative freedom. Practitioners in this company thought that quality systems did not improve quality, but instead created work in quality departments for less able developers and 'empire building' managers. They also felt that quality certificates were pointless, as they are awarded on the basis of consistency rather than quality. During the study we did, however, come across anecdotal evidence to suggest that the quality of the software produced by this company was not as high as the practitioners believed.

4.2 Damaging senior management styles

Quality initiatives in company D2 were overshadowed by a hard-line senior management style and a highly political organisation. Practitioners felt that a 'watch your back' way of working was necessary. They felt that such an environment made tackling quality difficult and made implementing quality initiatives at the grassroots risky for individuals.

Practitioners felt that senior management were also keen for the company *to be seen* implementing the latest quality initiative. This left practitioners lurching from one half-hearted quality initiative to the next. The result of this working environment was a very disillusioned and frustrated IT section who could see all the quality problems but felt powerless to improve them.

4.3 Outsourcing

An important influence on the quality views of practitioners in company D1 was outsourcing. A major part of their department's work had been outsourced and this led to an interesting quality juxtaposition. Not only were practitioners producers of software, but they were also customers of another software producer. As customers they were unhappy with the quality of the software that they were receiving, but felt powerless over the quality of their outsourced software. This made practitioners very keen on quality formalism for their software suppliers. Indeed, they wanted quality certification for their supply company, but were less keen on it for themselves.

5 Quality certification

Generally, most practitioners were both cautiously positive about quality certification and worried about bureaucracy and flexibility. Practitioners seemed to be torn between wanting the discipline of a certified quality system and wanting plenty of flexibility within the system. Practitioners felt that although certification *per se* did not improve quality, the process of certification had motivated many companies to take quality seriously. However, some practitioners believed that it was more important to put the organisational quality structures in place before starting to think about quality certification. These practitioners said that too many companies only implemented a quality system to gain quality certification. A minority of practitioners, predominantly in company B, thought that quality certification was a 'bandwagon' with nothing to recommend it at all.

There was a great deal of ignorance about how prescriptive certified quality systems need to be. Practitioners with no experience of certified quality systems had an exaggerated notion of how rigid and bureaucratic those systems were. Practitioners who had experienced certified quality systems were, on the whole, positive about those systems. However, two practitioners were very negative about their experience of certified quality systems at previous workplaces. They recounted how these quality systems had been very heavy handed and they felt put off certification because of this.

Practitioners within the certified companies thought that software quality had improved since certification (although no company had measured this perceived improvement). On the other hand, practitioners in the uncertified companies all voiced varying degrees of scepticism as to whether certification really could improve quality. In particular, practitioners were concerned that certification only meant that software was developed to be consistent rather than to be of high quality. Many practitioners believed that the money companies spend on seeking certification would have a bigger impact on quality if it was just spent on improving the development process. Practitioners at company D believed the reason they did not have certification was simply that senior management did not want to spend the money on the process of certification (although the company had done so to certify a publicly visible aspect of its service). Senior management's approach to certification in company D led us to suspect that certification was being used mainly as a marketing tool in that company.

Company C seemed to be most satisfied with their approach to certification. Although software at company C was not specifically certified, ISO9001 principles had been applied to software development within the company-wide certificate. Practitioners felt that they had the best of all worlds with their minimalist but helpful quality system. They felt they had useful quality structures without a massive bureaucracy. Furthermore practitioners felt that the quality system in company C had been implemented with a lot of staff consultation and input.

Positive comments on software quality certification made by practitioners at a number of the companies included the following:

- it looked good to customers
- it improved the status of IT within the company
- it motivated staff to do things correctly
- it helped new people get up to speed.

Negative comments on certification made by practitioners included the following:

- it is an irrelevance to quality
- it is less important than having a quality infrastructure and culture
- it is just about producing software consistently not necessarily of high quality
- it is nothing without practitioner professionalism and ability
- it is "*more about putting shelves on the wall to house the paperwork than anything of real value*" (manager in company D2)
- "BS5750 *tries to guarantee quality despite incompetence*" (manager in company B)
- it is "*for companies that employ monkeys*" (manager in company B).
- it is "*A way for certification bodies to hold small companies to ransom*" (developer in company B)

Particular certification problems that company A had experienced during TickIT certification included a lack of information and guidance about what is really necessary to gain certification and what is over the top. In addition there was poor performance from their quality consultants. In their experience, some consultants did not know enough about certification and their advice was positively damaging. On the other hand, they were very complimentary about the help and advice that they had received from their TickIT auditors.

6 Discussion and comments

6.1 Impact of the company

Although there was a broad consensus between developers and managers across the companies in this study, each company did have its own particular quality culture. The biggest single influence on a company's quality culture appeared to be the senior management's attitudes to quality. Senior management seemed to set the quality tone in every organisation. Furthermore, a company's approach to quality seemed to be directly related to the state of the organisation generally. Companies with a chaotic internal organisation also had a chaotic approach to quality; companies with streamlined and effective internal bureaucracies also had streamlined and effective quality systems. This adds weight to the view that companies should only look to certification after they have sorted out their organisational framework.

Only company B was opposed to quality formalism: all the other companies were implementing quality formalism to varying degrees with varying levels of success. Company B's stance was partly a function of it being a very small company (and partly because of the maverick approach to formalism taken by the managing director). With only 24 practitioners the managing director can probably keep a reasonable grip on quality without any formality. Whether their approach can sustain quality as the company continues to grow is doubtful.

Practitioners all agreed that quality will only be significantly improved by radical culture change in companies. The companies in the study were approaching culture change with varying degrees of enthusiasm: company A was succeeding, company C was trying hard, company D had not really realised that it was necessary and company B was determined not to do it.

Again, commitment to culture change seemed to be determined by senior management. Companies with senior management who wanted a culture change were changing and those which did not were not.

An issue that managers consistently raised was the impact on quality of professionalism. Managers in certified companies said that professionalism was important to quality, and managers in the companies with the least quality formalism said they relied exclusively on staff professionalism for quality. It is, however, of concern that some managers seem to be trying to put responsibly for quality solely on the individual practitioner, while at the same time putting practitioners under intense pressure to deliver software on time and to budget.

We were surprised that our results do not show a relationship between approaches to quality and whether companies produce software for internal or external customers. The determinants of an effective quality system seem more complex than that.

6.2 People factors

Previous quality experiences had the biggest impact on practitioner's attitudes to quality. There was a strong correlation between practitioners having experience of quality systems and practitioners being positive about quality systems. Ignorance about quality systems made practitioners anticipate the worst. This was compounded by practitioners feeling that there is very little accurate and detailed information and guidance on quality, as this left them even more ignorant and anxious about the unknown realities of quality systems. Practitioners who lacked knowledge about quality and quality systems worried predominately about the prospect of lots of bureaucracy and 'red-tape'. They did not, however, seem anxious about the curtailment of creativity.

Several practitioners in companies with an immature approach to quality had moved from companies with a lot more formalism. These practitioners were, on the whole, the loudest advocates of quality formalism. They thought that producing software was easier when quality structures are in place.

During the study we also got the feeling that practitioners were more positive and knowledgeable about quality if they were trained software engineers. Practitioners with a science rather than computing background, and a hardware rather than software background, were generally more negative about quality.

Most middle managers were aware of the dangers of implementing quality initiatives too quickly and too severely. Some were so sensitive about this that they appeared to be using it as an excuse for not tackling quality at all. Everyone seemed to blame someone else for their company's quality problems. Although there was an overriding feeling that developers wanted improvement to quality working practices, senior management were not really helpful to quality improvements and middle management were in the middle trying to satisfy everyone. However, it is interesting to note that, although our results portray senior management negatively, they were the only group not represented within the study. Had we known beforehand how important senior management is to quality we would have also targeted them. Clearly this is an area of the research that needs to be followed up.

It was also interesting that the developers in company D1 (who had some of their software supplied via an outsourcing contract) felt they had no control over software quality. This is probably a common feeling among ordinary buyers of software, and the reason why procurers are so keen on certification. Perhaps if quality software was delivered, then the industry would not be under the existing pressure to certify.

6.3 Resources dilemma

A significant issue for developers was the pressure on software quality from unrealistic project planning. Crosby's 'quality is free' argument [3] had not really penetrated the companies in this study. Deadlines were perceived as much more important to senior management than quality. It was also clear that managers implicitly condoned evasion of the quality system when the pressure was on. Indeed only in company A did managers not know that the quality system was evaded at times of high pressure (although developers said it was). Several practitioners complained that some quality procedures were too easy to get around and that they would prefer that not to be the case.

7 Conclusions and recommendations

Our results show that software practitioners are keen not only to see quality improved but, also to explore ways in which this can be achieved. Developers in almost all of the companies in the study said they wanted a more formal approach to quality, and the developers in the companies with the least quality formalism were most keen for more. Although the myth persists that most practitioners are negative about software quality, the positive practitioner attitudes that we found are already well documented [4, 5].

Our results show that companies have different approaches to quality improvement and unique quality cultures. We can illustrate this by comparing companies A and B. On paper both companies look similar, but in reality their quality cultures could not be more different. The two cultures are at opposing ends of a quality spectrum. We show that a company's quality culture is partly moulded by practitioner attitude to quality, but that practitioner attitude is profoundly influenced by senior management's attitude. Our results suggest that the factors traditionally believed to be significant have no discernible impact on a company's approach to quality. For example, whether software is developed for internal or external customers does not seem relevant to a company's approach to quality.

We also de-bunk the myth that practitioners are worried that quality initiatives impinge on creativity. We found only a few practitioners who thought that this was an issue.

The experiences, views and concerns of developers and managers were also not clear cut. Managers and developers had different primary quality concerns, but those concerns did not conflict. Both groups were also aware of, and sensitive to, each other's concerns. Developers were sympathetic to the pressures that middle managers were under and middle managers were actively trying to address the issues of importance to developers. Senior management came in for the most criticism. There was a general impression that many senior managers were out of touch. This united developers and middle managers in their frustration with senior management.

Our results lead us to suggest that for real quality improvement companies should do the following:

Encourage and nurture grassroots quality initiatives: Practitioners at all levels wanted to see quality improved. However, many developers and middle managers felt that they were powerless to effect improvement. There was a strong feeling among practitioners that grassroots initiatives had the most to offer and were those that succeeded in the long-run. Many practitioners felt that senior management quality initiatives were usually misconceived and the result of 'jumping on the next bandwagon'.

Start small and slow: Practitioners thought that the most successful quality initiatives had started small and unobtrusively.

Limit the pressure on practitioners: Some practitioners seemed to be in a 'no-win' quality situation. Quality was being made their professional responsibility at the same time that they were under tremendous pressure to deliver to schedule.

Understand the problems of software development: Many practitioners felt that senior management's failure to understand software and software development made developing quality software that much more difficult.

Quality bodies should do the following:

Emphasise a 'horses for courses' approach to quality: The most successful quality initiatives not only had flexibility within them, but also had different approaches for different types of project. This flexibility was popular with practitioners.

Publish more guidance: Most practitioners complained about the lack of really useful quality guidance, which meant that many practitioners lacked knowledge about quality systems and quality standards. This lack of knowledge encouraged practitioners to be pessimistic about quality. It also meant that companies undergoing certification were unsure about what they really needed to implement to gain certification. Indeed, the problem of companies taking quality formalism too far could be corrected by more support guidance from quality bodies. However, it is possible that practitioners were simply unaware of the existing quality literature, but even this situation must be tackled. It may be that access to information is the real problem, but either way ignorance was the single most damaging aspect of practitioner attitude to quality.

Disseminate positive quality experiences: Many practitioners knew apocryphal horror stories about quality. These rumours and exaggerations encouraged negative attitudes in practitioners who had no first-hand experience of quality systems. Very few positive stories of quality are reaching the practitioners on the ground. This is important, as positive experiences of quality played a significant role in making practitioners enthusiastic about quality initiatives.

Prevent quality standards being used solely for marketing: Our results suggest that quality standards like ISO9001 are being undermined by companies gaining the certificate only for marketing reasons. Many practitioners thought that marketing was the major motivation for gaining the standard and that quality was not improved by the standard.

Our results show that the experiences and views of grassroots practitioners are not often sought or listened to. We show that, although in the last 5-10 years there has been significant *theoretical* progress in the field of software quality, this has not always permeated ordinary companies. Many practitioners are left frustrated by poor quality but lack the power and information to do much about it. This is despite the fact that practitioners have very clear feelings about how things could be improved. The real experiences of many practitioners is that the market wants cheap software as quickly as possible and is not too concerned about the quality of it. Furthermore, the current commercial climate makes it difficult for companies to stop 'firefighting' and plan for long-term quality improvement.

8 Acknowledgments

We would particularly like to thank Dr. Isobel Nicholson (University of Central Lancashire) for her input to this study. We would also like to thank Professor Norman Fenton (City University) and Dr. Barbara Kitchenham (Keele University) for their valuable comments.

9 References

1 BEYNON-DAVIES, P.: 'Information systems failure' *in* MYERS, HALL and PITT (Eds.): 'The responsible software engineer' (Springer, 1996)

2 HUMPHREY, W.S.: 'Managing the software process' (Addison–Wesley, 1989)

3 CROSBY: 'Quality is free' (McGraw–Hill, 1979)

4 DEMARCO, T.: 'Controlling software projects' (Prentice–Hall, 1982)

5 PFLEEGER, S.L.: 'Lessons learned in building a corporate metrics programme', *IEEE Software*, 1993, **10**, (3), pp. 67–74