

# Software Quality: Perceptions of Managers and Developers

**Mícheál Ó hAodha**

Library and Information Services and  
Lero - The Irish Software  
Engineering Research Centre  
University of Limerick, Ireland  
Micheal.ohaodha@ul.ie

**Ita Richardson**

Department of Computer Science and Information Systems  
and  
Lero - The Irish Software  
Engineering Research Centre  
University of Limerick, Ireland  
Ita.richardson@ul.

Abstract

*The research presented in this paper focuses on software quality in small to medium sized enterprises (SMEs) in Ireland. We carried out qualitative research with software developers and managers, where we examined the general state of software quality within their company and followed this up with a detailed analysis of the specific software quality and maintenance processes used in their company. Following data analysis we achieved an understanding of the attitudes of management and developers towards a range of issues. Those presented in this paper include the priority given to software process, the ranking of software processes and market-driving considerations. We conclude that there are still two major stakeholders who need to be convinced – management and customers – before software process will be adopted on a wide scale among SMEs.*

## 1. Introduction

The recent downturn in the “dot-com” sector has forced the Irish software sector to turn inwards and examine some strategic ways in which the development of the Irish software industry can be protected and energised. While the Irish economy has the benefit of a comparatively low corporation tax (12.5%) as compared with its European neighbours, its position as one of the most expensive countries in Europe makes it imperative that new strategies for the maintenance of the software industry are explored in their fullness. One such exploratory path is the possibility for Irish companies to embrace the implementation of high-standard software quality processes, an approach which we currently see happening in India, one of Ireland’s largest competitors (Keane and Richardson, 2005).

According to reports over the past number of years the electronics and software sector employs an estimated 92,000 people within 1,300 companies, with a combined estimated turnover of €52 billion for the year 2003 (Central Statistics Office, 2004, ICT Ireland, 2005). As the sector is so important to the Irish Economy, it is imperative that quality is high on the agenda of software development organisations. At the same time many concerns are being expressed about high testing and maintenance costs of

software products, and there are many well-known cases where software has conspicuously failed.

Regardless of this, within the software sector, software quality initiatives may not always be given the required focus, for example, as discussed in He et al., 1996, who present the argument that quality tools have not been discussed systematically in the context of the software sector.

We can look at quality from many different aspects. One approach that can be taken is to consider the four fitnesses of Total Quality Management (Shiba et al. 1993, Bergman and Kelfsjo, 2003) - fitness to standard, fitness to use, fitness of cost, fitness to latent requirement. Fitness to standard and fitness of cost both focus on company issues. Fitness to use and fitness to latent requirements focus on customer issues. Within the software discipline, we consider the first three (standard, use, cost) as a basis for software quality.

## **2. Software Process Factors Under Review**

### **2.1 Fitness to Standard**

Fitness to standard means that the product built conforms to the standards laid down by the designers. When inspecting a product against a standard, it can either pass or fail the inspection. When software is being developed, similar statistically based measurements are often taken as an indicator of whether a software system is a 'quality product' or not. Statistical quality control is another evaluation tool used when determining fitness to standard. A weakness of using this concept as the determinant of quality within an organisation is that, in reality, it is very difficult to determine the quality of a product during inspection. Secondly, the product may pass the 'fitness to standard' test, but may not be the product that the customer wants.

### **2.2 Fitness to Use**

A customer purchases a product for a particular use. The difficulty faced by designers is that at times the customer will use the product for tasks other than those for which it has been designed. A simple example of this is the use of a screwdriver. Screwdrivers are designed to turn screws – however, another use of a screwdriver is to open cans of paint. Designers must be aware of what the customer is doing with their products and design them to fit those uses. Within the software development discipline, requirements engineering, defined by Berry and Ryan (1999), as “the development and use of cost-effective technology for the elicitation, specification and analysis of the stakeholder requirements which are to be met by software intensive systems”, concentrates on the fitness to use of software products.

### **2.3 Fitness of Cost**

Not only do companies need to produce products that comply with the concepts of fitness to standards and fitness to use – they must also comply with fitness of cost. This implies that the product can be produced at as low a cost as possible. The primary means by which this can be achieved is by improving the production process, so that produced goods will pass the inspection process and not be discarded or re-worked. Towards this goal, the seven steps for quality control and the seven tools for quality control have been devised. Since the late 1980s, Software Process Improvement (SPI) has been gaining popularity as an approach to improving software quality. It is expected that when the software process improves, the software product also improves (Humphreys, 1989). It has also been noted that when the process improves, cost is reduced and delivery schedules become more reliable. Software processes followed within software development life-cycles but there are others. Examples include project management, recruitment and training, configuration management and risk management.

### **3. The Research Question**

These concepts are fine in theory – but what actually happens in practice? The research presented in the paper looks at software quality from the viewpoint of management and the software engineers – those people close to the coalface of software development. Following our study, one question we ask is whether management is really that close to the coalface? Are there significant differences to the views of quality from the management perspective?

#### **3.1 Research Methodology**

In carrying out the research presented, the primary methodology was qualitative in nature. Data collection was carried out by interviewing over fifty individuals who operated at different levels of the companies in small to medium sized enterprises (SMEs) based mainly in south-western Ireland. The wide-ranging scale of the interviews and the different grades of worker within each company who were interviewed was indicative of the necessity to gain as wide a perspective on the intricacies of the software industry as a whole. In particular we were interested in how viewpoints at different levels in the organization considered the implementation of software process and quality procedures. A particular focus was an analysis of the perspectives of two stakeholders who are often perceived to approach software development from quite different standpoints –the manager and those developers working at the “coalface”.

Data was gathered by interviewing these various individuals within the companies and using a series of semi-structured interviews. This interview method encompassed a combination of open-ended and closed questions, the former of which was used to elicit unexpected types of information. The list of questions used during these interviews was typically divided into two sections – firstly, a section analysing the general state of software quality within the company and secondly a more detailed analysis of

the specific software quality and maintenance processes that were used by that particular company and attitudes towards these processes. Data analysis was then carried out by means of content analysis on the interview transcripts which had been transcribed from voice recording equipment.

## **4. Issues raised**

During this research, we achieved an understanding of the attitudes of management and developers towards a range of issues including the following:

- Was the implementation of software quality or software process measures perceived as a priority in their working environment?
- How did the implementation of efficient software processes rank in relation to the other work processes that the overall development of a new software product necessitated?
- How had the pressure for “on-time” product delivery impacted on the longterm implementation of software processes in their work situation?
- Were market-driven considerations such as the necessity for customer satisfaction the principal drivers behind those software processes had already been implemented in a particular company?
- Was there any mismatch evident between the expectations of what effective software processes could achieve according to where in the company hierarchy somebody worked?
- Were there evident economic threats to the continuance of an ad-hoc approach to software process on the part of many Irish software companies?

### **4.1 Priority of Software Process**

The first and most general observation that was made by many of the interviewees was the fact that software process and the implementation of quality processes were often overlooked entirely or were low on the priorities of many SME’s. Many interviewees were quick to acknowledge this while at the same time admitting that they believed software maintenance and the existence of inadequate documentation, particularly in relation to requirements’ specifications, were almost certainly consuming a major proportion of their company’s total development costs. Many software developers highlighted the probability that unless key players in an organisation were au fait with and supportive of software process initiatives then it was very unlikely that any progress would be made in the implementation of those software processes which would influence an improvement in software quality in the medium or long-term. A significant discrepancy was evident in the attitudes of company managers and the software developers on the importance of software process implementation.

## 4.2 Ranking the Importance of Software Process

We were also interested in where the implementation of efficient software processes ranked in relation to other work processes in the software development industry. Comments such as the following were frequent and indicative of a common opinion amongst management who were often more concerned with the scalability, appearance and performance of a particular software project or product:

*“I’m not so sure about the importance of software quality and software quality processes.....You can definitely have heroes (developers) who will pull the product out of the bag for you at four in the morning. I think the whole purpose of software quality is to let you get away with average developers who will get your product out on the door on time.”*

Many managers complained about the time and cost implications (“Time is money”) - inherent in the implementation of software quality processes. Many argued that the competitiveness of the market meant that there simply wasn’t the time to engage with software process procedures, irrespective of their possible advantages in the broader scheme of things:

*“I’m not so sure about software quality processes...the merits of it can be seen in the fact that there are minimal defects in projects. However, having said that, the time and effort spent ensuring quality is far too excessive. There is too much cost involved in obtaining control, that, in many circumstances, wasn’t needed.”*

The interview process also raised questions regarding the longterm viability of software processes given the pressures of a fast-changing and extremely competitive environment and the attendant emphasis on “on-time” product delivery. Some interviewees, for instance, were dubious as to whether recognised software processes really had a long-term benefit in terms of quality. They cited customer satisfaction, on-time product delivery and good marketing as factors which they placed far ahead of software processes in terms of their priorities:

Yes, it [a good software process] does help, but it is not a tangible link, and an improvement is not guaranteed. The best way I could explain it would be to describe the process as being like a recipe, and the product being the pudding. “The proof of the pudding is in the eating”. The process is a guide, but not descriptive enough to ensure a product of quality. There are other factors that will determine this....you must not only have a good process, but also a good lifecycle and a good methodology.

Software developers who were often lower down the chain-of-command in the organisation often echoed this finding. Ironically these were often developers working in a company where some form of systemised software process structure was already in place:

*“The main problem that I have encountered would have to be the company’s heavy-handed approach to documentation and procedures. This can be extremely excessive at times, especially if the change involves something small, such as changing a heading. It wastes a lot of both the developers and the company’s valuable time”*

### **4.3 Market-driven Considerations**

Some developers pointed to the “quick-fix” and market-driven nature of most companies, a fact that indulged the twin demands of slick marketing and customer satisfaction, even to the detriment of the product’s quality. For example,

No [Product quality is not dependant on a good software process], but it helps. A bad product could be marketed well and as long as it sells to the customer and the customer is satisfied then the management see this as a success.

Many developers indicated a change of perception taking place within their own companies towards an increased reliance on software process procedures, however, something which they attributed to changing or increasingly specific demands from their customers:

*“Well, software from my own experience has always been important but I find that there is a lot more focus now on recording the level of software quality you have achieved...there is a lot more time spent with project planning and getting it right at that level.... a couple of the customers that I am particularly involved with at the moment...they are becoming more involved.....some of it goes back to the initial outlay for some people.....they say they want something.... say a website... and they want to see where their money is going”.*

While the management in software companies often considered software process a waste of time, the opposite was the case for many software developers who saw better communication and realistic marketing of the product – i.e. a marketing strategy that was realistic about time-frames and customer expectations – as two very beneficial outcomes associated with better software process procedures:

*“Our own problems were communication and timing of products...the time we were allowed to do something and the actual time it took was a lot different. Most of the time the problem was due to the marketing....When they were selling a product, they would promise that it would be done in two weeks when we would need two months to do it. That’s the big thing, it was all about communication....we have resolved some of this now seeing as we are homing in on the whole idea of CMM and using processes to ensure that everything is traceable and all that”.*

#### 4.4 Mismatch in Expectations

Comments such as those previously mentioned pointed to the appearance of a major contradiction that became evident as we collated the results of our research. This was a serious mismatch between the perceived benefits of software process procedures as outlined by both management and the software developers. Time and again, interviewees, whether management or employees highlighted that the most pressing problem in relation to their company's software development was almost invariably requirements-related, in particular the absence of correct requirements' specifications. This was a cause of poor data gathering in the first instance, the poor collation and archiving of such data, changing requirements' needs and an inability on the part of the software company to pin down what the exact requirements needs of the customer were in the first instance. While both managers and workers acknowledged that the absence of specifics in relation to requirements' gathering and late changes in requirements needs on the part of customers were a contributory factor towards spiralling costs and huge over-estimations in terms of time-management; it was frequently the developers alone who highlighted the obvious fact that correct, seamless and easily-recordable software process procedures could be of huge benefit in this situation. While Management frequently focussed on marketing ("It'll be done in two weeks (in reality- two months!") and quick fix solutions in terms of documentation, requirements' problems and project estimation, it was the developers who highlighted the inevitable and unnecessary time-wasting and outlay associated with such an approach. Developers frequently highlighted the dangers inherent in such a "macho-management" approach, an approach which put very little emphasis on medium to long-term planning and was frequently either unaware or "in denial" about the realities on the ground. Developers often pointed to the potential benefits that could be gained from the standardization of software quality processes and the (ROI) return on investment advantages this approach would engender, but were still dubious as to whether the management culture would change so as to buy into a new "quality process" culture:

*"Again, from my own experience it is getting senior management to buy in, making sure that they know what is involved from start to finish and that's where software processes come in....they have to realise the scope of the project from the beginning.....Like, I was working in Dublin for one project and it started off as a 6 weeks project and two-and-a-half years later it was still going ahead. It kept rolling and rolling on as other bits were added to it. This can happen, particularly with smaller companies....So it is really about letting everybody know from day one...being clear about what can realistically be done within a specified time-frame".*

#### 4.5 Economic Threats

A number of developers also pointed to the longterm dangers inherent in the "fly-by-night" and last-minute/four o'clock-in-the-morning processes, particularly with the

decline in number of qualified personnel in the Irish software industry and the fact that many SME's were often relying on one individual to keep a huge proportion of the project's development process in their "memory" alone.

*"We follow ISO9000 but we aren't certified. We don't get audited.....From a quality point of view we definitely need a guideline on how to do things. It would reduce the time to market our products.....Once you get to a certain size, it's too dangerous just to have one person with all the information in their head....when there are more people involved you need to be following something especially for sharing information"*

Not all management surveyed had a negative view of the potential of software quality processes to drive success in an ever-competitive market. However:

*"The main reason the company follows an approved software process model is economic. The more efficiently we do something the faster we get it on the market, which satisfies our customers and allows us to focus on new products rather than spend time and resources on maintaining poor quality products that can be released prematurely....Process also helps us to adhere to better practices and gives the customer faith in our software so you could say that it is market driven as well....."*

Other managers indicated that they were beginning to move towards a view of software process whereby quality, market success and customer satisfaction were all seamlessly interconnected.

*"...I definitely see software quality as important and the point that we always drive home, and I think it's valid, is that, the later down the line that problems are found, the more costly they're going to be to fix.....Once problems get out into the field, they're a lot more hassle to fix and it's more difficult to keep the customer happy."*

## **5. Concluding Remarks and Future Directions**

While the latter comments generate a certain optimism as regards the potential integration of market-efficient software process and quality procedures in the future there is still a long way to travel before the Irish software community are likely to implement software processes on a wide scale.

Based on the outcomes of this research, there are two major stakeholders in the software equation who need to be convinced for the adoption of software process on a wide scale to take place. The first of these stakeholders are those at a management level within the software industry. As noted by Richardson and Varkoi (2003), software process adoption will not take place without leadership and sponsorship. Managers need to be convinced of the benefits which software process can bring to their organisation in terms of revenue, return business, seamless flow of work processes within the organisation, customer satisfaction, marketing and the retaining of an organisation's "institutional" memory.



The research on which this paper was based indicated a frequent lack of awareness on the part of management of any of these tangible benefits with the widescale “stereotype” that software processes were an excuse for shoddy workmanship or time-wasting continuing as prevalent amongst the more senior echelons of many companies.

Another point which became increasingly evident during the course of the research was the fact that some people working at management level in Irish companies were unaware of the flexibility and relative ease with which many software processes could be fine-tuned or tailored so as to meet the specific needs of their own organisation.

The bulk of the research strongly indicated that the second stakeholder i.e. the customer - is undoubtedly the most influential in the long-run and there were strong signs exhibited throughout the research results that the customer is already beginning to initiate a change towards an increased demand for proven quality processes. The attitudes of this customer-base could have a snowballing effect if customers begin to consider it the norm for software quality processes to be seen as a given in future marketing and business initiatives. With the rhetoric of transparency, EU standardization and “value-for-money” becoming common buzzwords Irish society as a whole it seems only a matter of time before customers insist on the “normalisation” of such processes and standards within the software industry.

## **Acknowledgement**

This research was funded by Science Foundation Ireland through the Global Software Development for SMEs Project.

## **References**

1. Bergman, Bo and Bengt Klefsjo, (Translated by Karen Ashling) 2003. Quality from Customer Needs to Customer Satisfaction. Studentlitteratur, Sweden.
2. Berry, Daniel M. and Kevin Ryan, 1999. Presentation in RENOIR URL [www.cs.ucl.ac.uk/research/renoir](http://www.cs.ucl.ac.uk/research/renoir), Requirements Engineering Network of International Co-operating Research Groups, Accessed in July, 1999.
3. Central Statistics Office: Information Society Statistics – Ireland 2004. Central Statistics Office, Government of Ireland. Dublin, Ireland. (2004).
4. ICT Ireland: ICT sector costs have increased by almost 20% in 2 years. <http://www.ictireland.ie/ibec/press/presspublicationsdoclib3.nsf/wvICT-News/BC8ADB4FFD87A57680256FC4003EB612?OpenDocument> 05/04/2005. Dublin, Ireland (2005).
5. He, Z., G. Staples, M. Ross, I. Court, 1996, “Fourteen Japanese quality tools in software process improvement”, The TQM Magazine, Vol. 8, No. 4, pp40-44.
6. Humphrey, Watts S., 1989. Managing the Software Process. Addison-Wesley, Reading, M.A., U.S.A.

7. Keane, Brendan and Ita Richardson, 2005. Quality: Attitudes and Experience within the Irish Software Industry, European Software Process Improvement Conference, EuroSPI05, pp 49-58, Budapest, Hungary.
8. Richardson, Ita and Timo Varkoi, "Managing for Change when Implementing Software Process Improvement Initiatives", European Software Process Improvement Conference, EuroSPI' 2003, University of Music and Dramatics Arts, Graz , Austria, 10<sup>th</sup>-12<sup>th</sup> December, 2003, pp X.1 - X.11.
9. Shiba, S., A. Graham and D. Walden, 1993. A New American TQM: Four Practical Revolutions in Management, Productivity Press, Oregon, U.S.A.