

The SEI's Capability Maturity Model: a critical survey of adoption experiences in a cross-section of typical UK companies

Kolind, J. P. and Wastell, D.G.

*Department of Computer Science, University of Manchester
Oxford Road, Manchester M13 9PL, England
{kolindj, was}@cs.man.ac.uk*

Abstract

The Capability Maturity Model (CMM) of the Software Engineering Institute has played a seminal role in the field of software process improvement (SPI). Most companies actively interested in SPI will have some familiarity with the model, and indeed CMM assessments are now legally required in the US for government software procurement. The model was, however, developed for defence contract work involving large organisations, and its generalisation to other software development contexts is therefore problematic (e.g. in-house projects, small software houses). This paper concerns the experiences of a representative sample of UK companies, untypical of the traditional application domain of the CMM, who have actively considered the adoption of the CMM. All but one rejected the model. The results are analysed in terms of Roger's seminal work on technology transfer. The main reason for rejection was the lack of a prior engineering culture in the companies, compounded by the failure to achieve short-term gains. External imposition of the CMM was another inauspicious factor. It is concluded that, for the CMM to achieve wider adoption, it must be made simpler, more easily tailored and more readily linked to business goals.

1 INTRODUCTION

This paper is concerned with the technology transfer of software process innovations, specifically the influential Capability Maturity Model (CMM) of the Software Engineering Institute. The inception of the CMM can be traced back to

two seminal papers by Watts Humphrey in the late 80s which described a process maturity framework (Humphrey, 1987a) together with a questionnaire enabling the operational assessment of maturity levels (Humphrey, 1987b). The original purpose of Humphrey's framework was to enable software organisations to identify areas where SPI could be most effectively carried out. The CMM has been strongly influenced by the work of Deming, Juran and Crosby (Paulk et al., 1993a) whose quality principles have been expanded upon in the light of feedback from both government and industry. Since its inception, the CMM has gained steadily in influence, assisted to a significant degree by the increasing use of the model to audit the technical capabilities of software suppliers. Defence contractors must now, for instance, adhere to level 3 of the CMM in order to develop software for the US Department of Defence (Saeidain and Kuzara, 1995). Today most software organisations seriously considering software process improvement (SPI) will be aware of the framework and many will have actively considered its adoption.

Although the CMM is intended to be generally applicable, the model was developed using experiences of government software contractors who do not represent the typical organisation in the software industry. Software development takes place in a wide spectrum of contexts and the adoption of the CMM thus far has been limited. This paper is concerned with differences between CMM adopting organisations and the rest of the software industry. The results will be reported of a survey of a representative cross-section of software development organisations in the U.K. who have considered the adoption of the CMM as a framework for SPI. The paper will describe their experiences, attempting to isolate the key factors underlying the varying fortunes of these organisations.

The next section provides a short background description of the CMM. The paper then continues with a description of the methodology used to select the interviewees. The case studies are then presenting in outline, highlighting the main issues that emerged; more detailed descriptions can be found elsewhere (Kolind, 1996). The paper concludes with a summary of the main themes emerging from the empirical work, including practical recommendations.

2 THE CAPABILITY MATURITY MODEL

2.1 Key concepts

The central concept in the CMM is that of the "maturity level" of a software process. According to Humphrey (Humphrey, 1990), software processes range from the chaotic (level 1) to the ideal (level 5). At level 1, the organisation's software processes are not defined, but are usually improvised by the software practitioners who carry out development. The main concern of management at this level is fire fighting. There are no objective measures of quality, hence quality is unpredictable. Cost and time estimates are usually made ad-hoc and are therefore

typically underestimated, and certainly unpredictable. In a level 5 organisation, on the contrary, the software processes are defined and managed centrally. The processes are updated as necessary, using methods such as those described by Juran for quality improvement (Juran, 1988). Time and cost estimates are based on historical records while quality is measured objectively. Therefore, time, cost, and quality are all predictable. The CMM provides a method of clarifying the process maturity level of organisations, so that lower level organisations may improve their software processes. The full description of the CMM and its different maturity levels has been provided by Paulk et al. (1993a, 1993b).

The SEI advocates two uses of the CMM, and provides the necessary tools for these uses. First, internal assessments can be carried out to identify areas where process improvement would be most effective; the model provides guidance in selecting a process improvement strategy by suggesting which key processes need to be established. Second, capability evaluations can be carried out by external organisations in order to evaluate the risks associated with a specific contract. The goal is to select the most able contractor to develop the project.

Assessments and evaluations are both based on the CMM Appraisal Framework. This framework ensures that results are consistent across the methods, allowing them to be compared. Internal Process Improvement (IPI) is the methodology advocated for self-assessments, and is aimed at an open, learning-oriented environment. IPI consists of structured and unstructured interviews. Although its primary objective is to assess the need for process improvement, the act of carrying out an assessment can also help to create the buy-in and motivation required to act on the findings. IPI assessments focus on organisation-wide assessment rather than specific projects or project teams. In contrast, software capability evaluations (SCE) are performed in an audit-oriented environment. Therefore, SCE's require more documentation of operational processes. Improvements may be suggested at the end of an evaluation. However, the primary goal is to select the right contractor to develop the product.

2.2 CMM Type Organisations

The Software Engineering Institute has carried out two surveys of how the CMM has affected process improvement in the software industry (Hayes and Zubrow, 1995; Kitson and Masters, 1992). These reports used data from the SEI assessment database, which contains information about all the assessments carried out by the SEI. The purpose of the report by Hayes et al. was to "investigate the experiences of organisations following the CMM as a model for software improvement" (Hayes and Zubrow, 1995). The results show that the CMM is generally successful in improving software activities (73 percent of organisations improved their standing after the second assessment). However, it is important to look at the types of firm studied. These organisations "tend to include relatively more DoD [Department of Defence] and federal government contractors, and have progressed

to a relatively higher software process maturity profile.” (Hayes and Zubrow, 1995). Such organisations are, of course, not typical of the software industry as a whole.

The CMM would thus appear to be far more relevant to large software firms who are primarily concerned with the development of software for a single customer. One may suspect, for instance, that these organisations typically have a strong engineering tradition and may therefore be presumed to understand the need for an engineering-based software process. The relevance of the CMM to the generality of software development organisations would thus seem to be problematic. Broadman et al. have discussed issues relating to small businesses and have found that many of the level 2 Key Process Areas are not relevant to small organisations (Broadman and Johnson, 1994). Another common software development context is that of the in-house IT department. The relevance of the CMM to this situation would also seem questionable. Typically such departments are located in non-engineering companies. One may speculate that such organisations lack the engineering culture needed to understand the issues presented in the CMM, and hence will not understand the need for an engineering based paradigm.

2.3 Technology Transfer and the CMM

In his seminal work on technology transfer, Rogers makes a fundamental distinction between situations where the need for transfer arises from within a social system, and situations where the change is imposed from without. Zerkowitz (1996) refers to the latter as the *rule paradigm of technology transfer*, wherein an outside organisation attempts to impose a new technology on the recipient organisation, which then infuses it internally. The rule model “represents an exportation of technology from one organisation to another”. The imposition of standards by either government or industry exemplifies this model at work. Unfortunately experiences of the rule model are far from positive. Rogers’ parable of water boiling in Peru shows the dangers of the approach. Zerkowitz argues that it is ineffective because “technology generally has to be wanted by the new organisation and cannot be dictated by others” (1996). However, the rule model is important because of its widespread use as a mechanism for technology transfer (e.g. quality standards such as ISO 9000).

Zerkowitz characterises the Capability Maturity Model as embodying the ‘rule model’ paradigm of technology transfer (1996). Certainly, to the extent that CMM compliance is a requirement to win Government and other contracts, the CMM does embody the rule model. However, this reflects a rather narrow view of the CMM. The model is more than an audit mechanism; an important motivation is to provide a framework for progressive software improvement and organisational learning. In this second mode, the CMM can be seen as providing a ‘positioning framework’ (Earl, 1989). Positioning frameworks provide a benchmark to enable a

firm to assess its own capabilities. This is important in order to create the impetus for improvement. To understand how to improve, a firm needs guidance as to where it should plan to move next and which are the most feasible improvement strategies. Using a term borrowed from Vygotsky, this 'zone of proximal development' aids a firm in choosing between the different strategies (Wastell and Sowards, 1995). The CMM helps define this zone because each level represents "a measure of improvement that is reasonable to achieve from the prior level" (Humphrey, 1990). The CMM's key process areas help divide the levels into manageable chunks, and pinpoint more precisely where a firm needs to improve. The assessment report usually includes specific activities, which must be carried out in order to reach the next level.

3 RESEARCH METHODOLOGY

The research reported here was carried out in order to examine attitudes and experiences in relation to the CMM across the British software industry. We have argued that the CMM appears to be applicable to a rather narrow range of software development contexts, and there is a serious worry that firms which require SPI most of all may be those for whom the engineering ethos of CMM is least well suited. The purpose of the research was to evaluate empirically the reactions of a range of typical software organisations to the CMM, and in particular to attempt to identify whether there were any common factors that bore on whether the CMM had been adopted or not. Organisational culture was one factor of high *prima facie*

Table 1 Organisations surveyed compared to the typical CMM organisation

| <i>Characteristic</i> | <i>Archetypal CMM organisation</i> | <i>Organisations in survey: typical characteristic</i> |
|----------------------------------|------------------------------------|--|
| <i>Culture</i> | | |
| IT as percentage of organisation | Large | IT not the major part of the organisation's business |
| Maturity Level | High | Low |
| Background | Software Engineering | Other Background |
| <i>Projects</i> | | |
| Size | Large | Small |
| Complexity | Technically Complex | Information systems |
| Requirements | Well-defined | Fuzzy |

interest. Another key issue was the locus of the stimulus to adopt CMM, i.e. whether the motivation to adopt was imposed externally (i.e. the rule model) or arose through an internally recognised desire for SPI (the organisational learning paradigm).

The companies interviewed were chosen from the UK Software Process Improvement Network (SPIN); this is a British Computer Society (BCS) special interest group, which draws together quality personnel from a wide variety of companies across the UK. The purpose of the group is to share SPI experiences and help disseminate good software engineering practices. The SPIN list contained a total of 378 names; this was narrowed down to companies in the North West region and around London for logistical reasons. The final number of applicable companies for a telephone survey was 52.

The initial telephone survey was carried out on a sample of 16 of these companies in order to select the final group. Of the 16 companies contacted, 6 had significant direct experience of the CMM, or a close derivative such as Trillium (Bell, 1992). These companies were then followed up by in-depth interviews with one or two key staff, typically a senior member of the quality department or someone else with a senior technical role in the company (such as the Technical Director). The 6 companies included a wide range of industry sectors, including banking, insurance, telecommunications, package software development, and software consulting. Table 1 summarises the typical characteristics of the companies in the survey contrasted with the archetypal CMM organisation, as suggested by the Hayes and Zubrow survey.

The interviews were structured across key areas of discussion, which included:

- Company backgrounds (this covered the main areas of development work as well as a more general background to company policies)
- CMM experiences (including reasons for using or not using the CMM, implementation strategies, difficulties found with implementation, and any other critical comments)
- Other software process improvement activities.

To ensure that interviewees gave a more open picture the interviews were not recorded. However, the interview notes were written up within 24 hours of the interview to ensure no important information was lost. The notes were later checked with the interviewees; this gave a chance for some interviewees to add to the information already given in the interview. All companies insisted on a confidentiality agreement. Hence, the names of companies and interviewees have been disguised.

4 OVERVIEW OF THE INTERVIEWS

The organisations interviewed fell into two categories. The first category included companies (Small Software Developer Ltd. and Multinational Inc.) that were focused on developing a moderately technical product, which is sold to multiple external customers. The other software organisations were concerned with developing information systems to be used internally in the organisation. This section highlights the different experiences of these diverse companies, focusing on two issues in particular: a) the role played by culture and b) whether the motivation for CMM was endogenous, motivated by the need for SPI, or exogenous, following rule model.

4.1 Multinational Inc

Multinational Inc. is a large telecommunications company, developing mainly telephone switching systems and the software to support them. It employs around 4000 people in the UK. The organisation is both ISO9000 and TickIt certified and has strong links with Philip Crosby, a quality guru, through its parent company. The organisation is currently working towards the Malcolm Baldrige award for business excellence and has based its business processes on the Europe for Quality Management (EFQM) model.

Multinational Inc. had a highly developed engineering culture because of its previous work in developing hardware for switching systems. Although the IT division now concentrates on software based work, this understanding of the need for engineering based processes has meant that the engineering paradigm had transferred well to software. The software processes were highly controlled, and developed through experience and historical metrics. This has resulted in an informal assessment of maturity level 4 against the CMM. Multinational Inc. conforms well to the profile of the sort of organisation (table 1) which has traditionally been successful in using the CMM to improve software quality. These organisations typically see quality management as a vital part of the production process prior to the improvement program and hence do not face the cultural problems faced in other organisations. Hughes Aircraft is the most cited example of CMM success (Humphrey et al., 1991).

4.2 Small Software Developer Ltd

Small Software Developer (SSD) Ltd. is a small software house developing Laboratory Information Management Systems (LIMS) and Chromatography Data Systems. It consists of approximately 100 staff, 40 of whom develop software. The company has been certified to the ISO9001/TickIt quality standard. SSD faced two major problems. First, most of its American pharmaceutical customers were required to carry out extensive audits in order to satisfy government regulations.

Second, customers were encouraged to suggest changes and improvements to the software. The result was a huge problem of prioritising requests when trying to decide what to put into the next version of the software. The same was true with defect removal. As one interviewee commented, if a severe defect only affected one customer it would not get fixed before the minor defect that affected 90.

Prompted by the requirement of its customers for quality audits, SSD had looked into using the CMM to 'certify' its processes. However, the quality team had found that many of the questions were arbitrary or irrelevant to their circumstances. CMM was thus felt to impose a large overhead for the quality department, who typically assisted in over three audits a months. CMM had thus been dropped. The company can be seen as fire fighting with no understanding of the need for an engineering based software process. This is illustrated by the following quotes:

'It is often said: If you can't measure it then you can't control it. However, I don't agree. When you have to make a decision you always instinctively know what to do.'

'Processes change so quickly that you cannot gain meaningful information when you measure them.'

There was consensus in SSD that software process improvement was important, but only to decrease the time to market; SPI in terms of defect prevention was not felt to be an important issue. SSD provides a vivid example of a company which urgently needs process improvement, but simply lacks the engineering culture to understand this. Although the company had implemented inspections, and seen some improvements in terms of reduction in defects, it was not felt that this had decreased the time to market and hence failed to meet the company's primary objective.

4.3 Highstreet Bank plc

Highstreet Bank plc. is a major high street bank, providing a wide range of services, both to the general public and to other financial institutions. A team of 1500 software developers, create the software required to 'support the business' of the bank. This includes systems such as Management Information Systems (MIS) and Account Management Systems. In their interview, Highstreet Bank highlighted two major problems. First, there was a need to decrease development time ("increase productivity"). Second, they wanted to reduce the number of defects ("increase quality"). To tackle these problems the quality department had attempted to develop a Quality Management System (QMS). Initially, an informal CMM-based assessment had been carried out in order to identify the best practices within the organisation. The assessment had produced a variety of maturity levels

(levels 1 to 3). The goal of the QMS was to standardise practices so that all project groups would be assessed at level 3.

Three problems had surfaced when trying to implement the QMS. First, bonuses had been directly linked to key performance indicators. This had introduced 'fudging' of the statistics in order to generate higher bonuses. The metrics collected were therefore useless when trying to analyse where improvement was needed. Next, the quality department had attempted to improve processes in one big leap. This had caused inappropriate measures to be taken in the first place. Level 3 metrics were being applied instead of those more appropriate to level 1. Last, it was felt that the CMM required a level 1.5 because the jump from level 1 to 2 was seen as "too large". The lack of an engineering orientation showed up elsewhere in the bank. For example, the software group had recently been broken up into smaller units in order to bring software development closer to the end user. Although this is a noble idea in itself, it had introduced a lack of communication between the different software divisions. In general, it was indicative of how software development was being managed as another business function, with the need for an engineering discipline being accorded a low priority.

4.4 Telecommunications Ltd

Telecommunications Ltd. is a large telecommunications company. The department interviewed (OSS) existed to support the networks' infrastructure side (a much larger IT department was responsible for other IT functions such as the billing systems). There was a total of 8000 people employed by the company, and 100 people were employed in OSS spread over 3 locations. The process improvement initiative in the OSS department had been driven externally, after the company had merged with a Canadian company. The external nature of the initiative had had two effects on the assessment outcome. First, it had become a political exercise rather than a genuine process improvement exercise. Second, the results of the assessment had not been widely disseminated throughout the organisation with "only a handful of people having read the final report". The assessment was based on the Trillium model and was carried out by a team consisting of two Canadians and four employees internal to OSS.

The final report had consisted of five key improvement recommendations, of which 2 aspects had been partially worked on. As a result the department have developed a quality management system (QMS), and in that process have attempted to formalise and communicate best practice. However, there was no real evidence of cultural change. It was reported that project teams had not co-operated on documenting processes, nor had any metrics been put in place. Moreover, the long term approach to quality management was going to be reactive rather than proactive: rather than looking for process improvement initiatives, the quality department were relying on project managers to request improvement activities.

4.5 We Insure Ltd

We Insure Ltd. is a part of a larger company selling insurance services to the general public. We Insure provide the life insurance and pensions services of this larger organisation. From a total of 2500 staff, 300 staff are involved in software development “to support the sales of life insurance and pensions.” In the early 1990’s there was a strategic review of business activity which had led to dramatic downsizing of the organisation and a total of 800 staff were lost. As part of this review, it was felt that technology development was key to improving the market position of the company. The technology would provide the company with much needed flexibility in an ever changing marketplace. A special task force called QUEST (Quality and Excellence in Service and Technology) had been formed to drive this initiative forward.

QUEST had developed a two-pronged strategy to improve software processes. First, the CMM was chosen as an external model upon which to benchmark current and future processes. Second, there was a generic education of the employees in the principles of quality management. The CMM had been chosen because it was felt that it had been independently validated and would provide an objective external measure. However, a decision was made to focus on only relevant key process areas and ignore others such as the subcontracting key process. An initial informal assessment put the organisation at level 1. The goal of QUEST was to improve the organisation to reach level 2 by September 1996. The second strategy, to educate employees in quality management, was important in order to create the need for process improvement.

However, there was still resistance towards QUEST. Project managers were reported as complaining they had of little or no time to spend on writing process definitions, and identifying metrics and valid process variants. We Insure had had a long history of attempted process improvements which had engendered this resistance because of bad experiences in the past. However, there was appreciation of how important the cultural change was. Although We Insure still manage software development by traditional methods, there was an underlying understanding of engineering quality methods which had been reinforced by the CMM experience. This gave some optimism that We Insure will ultimately succeed in their process improvement program.

4.6 Big Bank Ltd

Big Bank Ltd is major international bank, providing a variety of financial services. UK Systems Development (UKSD) is the division which provides software and technology support to all the UK high street branches. Out of 900 staff, 700 were involved in the development and maintenance of software.

The quality unit of UKSD had chosen a two stage strategy to introduce process improvement initiatives into the organisation. First, a business case had been made

for improvement activities and presented to senior management. This had created the necessary 'buy-in' and support necessary to carry out any improvement. Second, a process assessment model had been chosen based on twenty one selection criteria. These criteria had been identified through consultation with two business units and an external consultancy, and included such items as the cost of implementing the model and conformance to SPICE, an international standard for process assessment models. The four models to be evaluated were CMM, Trillium, Software Technology Diagnostic, and SPICE.

The CMM was not chosen as the model for process improvement. UKSD felt it was too "scientific" in its approach, was too costly, and had limited support in the UK. It was perceived that CMM had been developed "with a skew towards development of large, real-time scientific applications, developed in 3rd-generation languages" and was therefore not seen as appropriate to developing information systems. It was also felt that the CMM would be very costly to administer regularly because of the degree of rigour required and the breadth of its scope. Instead the Software Technology Diagnostic (STD) was chosen. Three perceived advantages had apparently motivated this decision. First, STD's scope embraced business processes rather than just software development. A second advantage was that STD can readily be linked to business goals, therefore enabling cost-effective process improvement. Thirdly, STD explicitly supports a selective rather than a global approach, allowing improvement efforts to be focused on only those processes felt to be critical.

Three reasons for the success of SPI at Big Bank were highlighted in the interviews. First, management had been convinced of the need for process improvement. Although senior management support had been hard to maintain in the long run, "process owners" themselves (project managers) had been strongly committed. Therefore, the usual problems of under-funding were avoided and focus had been maintained. Second, the improvement model had been chosen based on criteria defined by the staff who were going to use it. There was therefore little mismatch between the model and reality. Third, business goals and process improvement had been inter-linked. This had meant that only cost effective process improvement had been made, thus sustaining management interest.

5 DISCUSSION AND CONCLUSIONS

Although Rogers' seminal model of technology transfer applies to adoption processes at the individual level, the model, with some adaptation in the detail, nevertheless provides a useful conceptual framework for interpreting the present results (summarised in table 2) which pertain to organisations. In essence, Rogers characterises technology transfer as a process in which various *factors* bear on the outcome (i.e. the decision to adopt or to reject the new technology). These factors can be grouped into two broad areas, namely those factors intrinsic to the recipient

Table 2 Summary of the case studies

| <i>Company</i> | <i>Cultural Antecedents</i> | <i>Stimulus for change</i> | <i>CMM Adoption experience</i> | <i>SPI Impact</i> |
|-----------------|---|----------------------------|--|-------------------------------|
| Multi-national | Strong Engineering Culture | Internal | Adoption and use | Moved from level 3 to level 4 |
| SSD | Sales oriented culture, fire-fighting approach to quality | External | CMM rejected | None |
| Highstreet Bank | Consumer orientation paramount rather than engineering rigour | Internal | Engineering discipline of CMM not understood, misapplied | Limited |
| Telecoms | Young company, no prior quality culture, essentially reactive approach to quality | External | CMM resisted and marginalised | None |
| We Insure | Non-engineering but receptive to engineering approach, history of SPI efforts | Internal | CMM tailored, goal to reach level 2. Some resistance. | Moderate, good prognosis. |
| Big Bank | Strong management commitment to SPI, but subordinate to business goals (must be cost-effective) | Internal | CMM too costly and not linked to business goals. STD chosen instead. | High |

and those which reflect features of the technology. Rogers also emphasises the importance of the locus of the change initiative (exogenous or endogenous) as a critical prognostic indicator. The results of the present study will now be discussed in terms of these three factors, i.e. locus of initiative, recipient attributes and technology characteristics.

Regarding the source of the transfer initiative, the present findings strongly suggest that the outcome for CMM implementation is somewhat inauspicious when the initiative is exogenous, i.e. the CMM is externally imposed in an audit-oriented mode. The two cases of the rule paradigm in table 2 (SSD and Telecom Ltd) both led to CMM rejection, and ultimately no tangible improvement in software quality. When the change initiative arose endogenously, on the other hand, two of the cases led to adoption of an SPI approach (not CMM ultimately in one case) and to significant gains in performance, and in the other two cases there were some benefits at least from the experience.

Regarding recipient features, the present results strongly reinforce the decisive role played by organisational culture. Many other studies of technology transfer have underlined the importance of this factor. McMaster and Kautz (1994), for instance, attributed the failure of a city council to assimilate a structured methodology (SSADM) to the lack of fit between the formal structure of the methodology and the informal work practices of the development staff. To take another example, Le Quesne (1988) found that software houses with individualistic work practices were less successful in utilising IPSE tools than those having a more co-operative, team-based ethos. The present results strongly suggest that receptiveness to CMM depends critically on the prior existence of an engineering culture (or at least sympathy to engineering methods at a senior level in the recipient). Where this culture does not exist, the prognosis is poor. The irony of this will not be lost on the reader; those organisations most in need of CMM (i.e. level 1) will be those most likely to resist and reject it.

This difficult situation is exacerbated by the characteristics of the technology (i.e. CMM) itself. Rogers argues that several key attributes of an innovation contribute to the decision to adopt or reject: *relative advantage* (the degree to which the innovation is perceived to be beneficial), *compatibility* (i.e. the issue of cultural fit discussed above), *complexity* (the degree to which the innovation is perceived as difficult to understand and use), *trialability* (ease of experimentation with the technology), *observability* (the degree to which benefits are readily visible). In terms of these attributes, the outlook for the CMM is inauspicious. All but Multinational Inc. commented on the size and complexity of the CMM, observing that many aspects of the model did not apply to their circumstances. Broadman et al's study reports such a view to be typical of the majority of organisations in their survey. Adoption of the CMM entails a significant investment of organisational effort, and several of our respondents commented on the need for cogent evidence of the gains that would result (e.g. from published case studies of comparable organisations). The need of managers to achieve short term performance improvements poses particular problems in the case of complex technologies, such as those associated with SPI where the gains may take many years to work through. Studies of structured development methods indicate that many organisations give up on SPI because of the failure to achieve short term gains (Wastell, 1996; McMaster and Kautz, 1993).

In summary, the present results suggest that major problems need to be overcome if the CMM is to be more widely adopted by software development organisations. The need for a prior engineering ethos, the perceived complexity of CMM, its lack of trialability and the inherent difficulty of demonstrating benefits, are all serious impediments to its wider adoption. It is striking that only one organisation in our sample had adopted the CMM, and that this organisation appeared to be somewhat anomalous (a high level of maturity and a strong engineering culture, arising from its historical role in hardware development). The other five companies, whose need for SPI was rather more acute, had all largely rejected the CMM. That the present sample represents companies who may be supposed to be more committed to SPI than the typical software organisation (they are SPIN participants for instance) makes this result all the more stark. Clearly major issues need to be addressed if the CMM is to achieve more widespread application. One helpful measure would be to make the CMM more readily tailored to the circumstances of the recipient organisation, in particular in relation to the company's business goals. The two non-engineering organisations that had achieved some SPI success had achieved this either by customising the CMM (We Insure) or by adopting what they felt to be a less monolithic, more business-oriented approach (Big Bank).

Another promising way forward is provided by Humphrey's Personal Software Process (Humphrey, 1994). This is a training course for individuals which aims to inculcate disciplined engineering techniques derived from key practices of levels 2 and 3 of the CMM. Whereas the CMM is a top down approach targeted at changing the organisation, PSP is a bottom up approach aimed at changing the individual. Laboratory studies have shown that significant improvements in productivity/quality can be achieved with PSP in a relatively short time (Humphrey, 1994; Sherdil and Madhavji, 1996). Implementation of PSP could thus be used to facilitate the introduction of organisation-wide SPI (i.e. CMM) in two ways. First, by helping to develop more of an engineering culture in the organisation by fostering a more rigorous approach to software development at the individual level. Second, by producing some short-term benefits, the PSP will help to build up and sustain the high levels of motivation and commitment that are required over the much longer period required for the profound changes in organisational practices that the CMM entails.

6 REFERENCES

- Bell (1992). TRILLIUM - Telecom software product development capability assessment model, draft 2.1, Bell Canada, July 1992.
- Broadman, J. G. and Johnson, D. L. (1994) What small businesses and small organisations say about the CMM. In *16th International Conference on Software Engineering*, pp. 331-340.
- Earl, M. J. (1989) *Management Strategies for Information Technology*. Prentice Hall International (UK) Ltd.
- Humphrey, W.S. (1987a) Characterizing the software process: A maturity framework. Technical Report CMU/SEI-87-TR-11, Software Engineering Institute.
- Humphrey, W.S. (1987b) A method for assessing the software engineering capability of contractors. Technical Report CMU/SEI-87-TR-23, Software Engineering Institute.
- Humphrey, W. S. (1990) *Managing the Software Process*. Addison Wesley Publishing Company Inc.
- Humphrey, W. S., Snyder, T. R., and Willis, T. R. (1991) Software process improvement at Hughes aircraft. *IEEE Software*, 8(4):11-23.
- Humphrey, W. S. (1996) Using a Defined and Measured Personal Software Process. *IEEE Software*, 13(3):77-88.
- Hayes, W. and Zubrow, D. (1995) Moving on up: Experience doing CMM based process improvement. Technical Report CMU/SEI-TR-95-008, Software Engineering Institute.
- Joseph M. Juran. (1988) *Juran's Quality Control Handbook*. McGraw-Hill Inc., 4th edition.
- Kitson, D. H. and Masters, S. (1992) An analysis of SEI process improvement assessment results: 1987-1991. Technical Report CMU/SEI-92-TR-24, Software Engineering Institute.
- Kolind, J. P. (1996). The Capability Maturity Model in the United Kingdom. Master's thesis, University of Manchester.
- Lequesne, P. N. (1988). Individual and organisational factors in the design of IPSEs. *Computer Journal* 31(5):391-397.
- McMaster, T. and Kautz, K. (1993). The failure to introduce system development methods: a factor-based approach. In *Proceedings of the IFIP conference on Diffusion, Transfer and Implementation of Information Technology*, Pittsburgh.
- Paulk, M. C., Curtis, B., Chrissis, M. B., and Weber, C. V. (1993a) Capability Maturity Model for software, Version 1.1. Technical Report CMU/SEI-93-TR-24, Software Engineering Institute.

- Paulk, M. C., Weber, C. V., Garcia, S. M., Chrissis, M. B., and Bush, M. (1993b) Key practices of the Capability Maturity Model, version 1.1. Technical Report CMU/SEI-93-TR-25, Software Engineering Institute.
- Rogers, E. M. (1971) *Communication of Innovations: A Cross-cultural approach*. The Free Press, New York.
- Sherdil, K. and Mahhavji, N.H. (1996) Human-oriented improvement in the software process. *Lecture Notes In Computer Science* 1149:145-166.
- Saeidain, H. and Kuzara, R. (1995) SEI Capability Maturity Model's impact on contractors. *Computer*, 28(1):16-26.
- Wastell, D. G. and Sowards, A. (1995) An information system profile of the UK manufacturing sector. *Journal of Information Technology*, 10(3):179-189.
- Wastell, D.G. (1996). The fetish of technique: methodology as a social defence. *Information Systems Journal*, 6(1):25-40.
- Zelkowitz, M. V. (1996) Software engineering technology infusion within NASA. *IEEE Transactions on Engineering Management*, 43(3):250-261.