

Process Maturity - Growing older and wiser? Using *Process Advisor* for Process Assessment

A Bryant and J Grogan

Tony Bryant, Professor of Informatics, Leeds Metropolitan University, Leeds LS6 3QS; email A.BRYANT@lmu.ac.uk; fax +44 113 283 3182

John Grogan, Research Assistant, Leeds Metropolitan University, Leeds LS6 3QS; email J.GROGAN@lmu.ac.uk; fax +44 113 283 3182

Abstract

The idea of process maturity has a particular allure for software developers: Not only does it support the argument that there is a central development architecture for software construction; but it also lends an air of credibility to the efforts at improvement that have characterized earlier periods and now extend into the present and future. The software crisis may be no more than symptoms of growth: process puberty and adolescence. Maturity will bring with it a wider and deeper set of foundations for general software practise; and in the not too distant future the period from 1968 to approximately 1998 will be seen in a more favourable light, as a period which had to be endured, but has now been surpassed. Whether this happy situation comes to pass is dependent on a variety of factors and forces, not least of which is the ability of organizations and practitioners to come to terms with their own shortcomings, for instance as demonstrated by the findings of indicators such as those designed to assess process or capability maturity. What follows is based on our initial use of one such indicator.

Keywords

Software Engineering, Process Improvement, Process Maturity, Capability Maturity

1 INTRODUCTION & BACKGROUND

The concept of process maturity is an inherently appealing one, but also derives great strength from an understanding that the route to improvement and perfection consists of many small steps and stages, rather than a single leap. The general idea of maturity, in a quality sense, derives from the work of Deming and Crosby, although the central figure for *software process improvement* is Watts Humphrey and his work on the Software Process Maturity Model (SPMM) which developed into the current programme at the Software Engineering Institute (SEI).

In fact the concept of maturity with regard to IT, if not exactly software engineering, dates back a little further to the work of Gibson and Nolan. Their model of technological maturity was focused more on the application of the technology, but they employed the terminology of growth and a process of development in a fashion similar to that now used with regard to process improvement and maturity. (See Bryant, 1995 chapters 3 and 4 for a more extended discussion of the links between these models.)

Any growth model immediately evokes a number of questions -

- Is it necessary to pass through all the stages in a defined sequence?
- How quickly does the maturation proceed, and can it be accelerated or retarded in any way?
- How can one assess where one is along the route of maturity?

The wide variety of applications modelled around SPMM all seek to offer responses to these questions. The general SPMM model was initially offered as a basis for assessment, internally by the development organization itself. Later variations such as the Capability Maturity Model (CMM) were oriented towards external, customer-based assessment. But the mere existence of such models does not necessarily assist an organization in moving from assessment to improvement. There is also the issue of how applicable the model actually is, given that there is such a wide range of software development contexts. Does a particular model apply only to very specific forms of software development - e.g. real-time or safety critical systems? These issues have been recognized by those developing the maturity models, and much of the innovation and refinement over the last 5 years has been concerned with developing various forms of guidance, extension, specialization and so on.

In the commercial world Process Advisor (PA) has been available for some time as 'a self-directed system for software process improvement' It is clearly derived from the SPMM approach, but is marketed in a slightly different way as 'a low-cost, multimedia product' (Pressman, 199.).

If SPMM is like a health check from your family practitioner; CMM is more like a health check demanded by your insurance company before they will renew your policy: PA is a little like a *Do-It-Yourself* health check that can be done in the comfort of your own home. Given that a large proportion of software is developed in-house by fairly small development departments, or by independent software companies with fewer than 50 employees, the relatively informal, low-cost PA approach can be seen to have an obvious market.

Some organizations probably have used PA in precisely this way. It is certainly cheaper and less externally visible than employing a consultant to do the assessment. On the other hand, there are many who would argue that without some external and disinterested party to such an assessment there is little value to the findings. At Leeds Metropolitan University (LMU) we have been interested in process maturity for some time, both in itself and as part of our work on standards and quality for information systems.

We have used PA as part of our teaching, but have also been keen to show students how PA is used in practice. One way in which we have begun to accomplish this has been to apply PA to the development department of a large organization with whom we collaborated previously. This has demonstrated the practical issues which arise in a process assessment project, and has also provided insight into the PA package itself. In what follows we describe the PA package,

before giving a description of the context in which we used it, and then offer some insights into its use and the general lessons which we drew from the experience.

2 PROCESS ADVISOR

Process Advisor (PA) is a software process improvement method developed in 1992 by Roger Pressman. It consists of a number of stages designed to guide a company through a phased approach to software process improvement. Pressman uses the term 'Software Engineering Implementation Life Cycle' (SEILC) to refer to the six stages he identifies and his advocated progression through those stages.

The philosophy of PA is similar to that of the Software Process Maturity Model (SPMM) developed by Humphrey (1989) and later used to underpin the Capability Maturity Model (CMM) from the Software Engineering Institute (SEI). In summary this philosophy states that in order to improve and sustain the quality of software *products*, the software development *process* must be improved. The overall ideas are drawn from a variety of sources, such as Deming and Crosby; and Humphrey's application of them to software development has spawned a large number of software process improvement and assessment initiatives, including CMM (1993), Bootstrap (1994), Trillium (1994) and SPICE. PA does not directly implement the SEI approach, although many of the questions used in PA have been openly adapted from the CMM questionnaire.

One of the findings of the early uses of SPMM and like was that 80-90% of organizations assessed were at the two lowest levels of the 5 point scale. This proportion has not altered dramatically during that time, and it is for this reason that PA has been designed particularly for organizations at maturity levels 1 and 2 of the CMM; although it can also be used by more advanced organizations. The strategies developed as a result of using PA are also said to be consistent with the philosophy behind general quality standards such as the ISO 9000 series (1991). PA has been marketed as a low-cost alternative to - or basis for - the CMM.

2.1 The Process Advisor Method:

PA is described as a 'self-directed system for software process improvement', which means that it is expected to be utilized by the developers themselves to improve their own software development process. PA aims to offer guidance for users, charting a route through the 'Software Engineering Implementation Life Cycle'. This consists of six stages starting with an initial assessment of current process practices, through selection and justification stages, to eventual implementation of improved practices. The stages are defined as follows:

Assessment:

Objective - Assess the current state of Software Engineering Practice within the organization, a 'look in the mirror'. The software process assessment is used to determine the strengths and weaknesses of current practices. This is accomplished through use of a detailed questionnaire, followed by evaluation of the responses.

Education:

Objective - Increase the level of knowledge of software engineering methods, procedures and tools by those involved with development. An 'Education Planning Matrix' is

coupled with the results of the software process assessment, providing the basis for an education strategy to be developed.

Selection:

This encompasses three elements:

- 1 - establishment of overall goals and criteria for selecting methods, procedures and tools
- 2 - identification of the particular methods, procedures and tools that have a high probability of improving the existing software engineering practices
- 3 - choosing, justifying and acquiring software engineering and CASE technology

Justification:

Use of the Justification Model provided enables previously collected baseline metrics to be used for productivity and quality projections for later assessment of decisions and selections.

Installation:

A Transition Plan is created which identifies the strategy chosen to implement the software engineering methods, tools and procedures and CASE technology selected earlier. This involves the definition of tasks, milestones and deliverables, and also the assignment of responsibilities for the individuals involved.

Evaluation:

Another 'look in the mirror'. Initially at a macro level, by focusing on the success and acceptance of early transition tasks, and finally by undertaking a technical evaluation once an improved process has been applied to one or more projects.

The overall sequence of stages may be altered so that some run in parallel. Following the final evaluation stage, the process is iterative.

For our assessment of PA we devised an action research project involving the use of PA to undertake an assessment of an organization's current software development practices. Therefore we focused almost exclusively on the first stage of the SEILC.

2.2 Process Advisor contents

The PA package consists of three main elements

- a workbook, supported by a disk containing questionnaires, automated models, and standard reports, to take the user through the SEILC stages
- a video of sessions that support the main SEILC stages and activities
- a text book 'A Manager's Guide to Software Engineering'.

The workbook is aimed at the person handling the PA exercise, and it seems to be taken for granted that this would be a (senior) manager of some sort. The workbook contains a detailed description of each of the stages and activities in the SEILC. It also contains the questionnaires, models and reports that are used at the various stages. The intention is that the workbook guides the user through the SEILC, directing the activities, advising on what to look for and making suggestions on how to handle the results. The workbook is also

supported by a computer disk containing automated support aids for the questionnaires and models used, together with a number of templates for the resulting reports.

The video provides sessions on each of the stages of PA, helping the user to prepare for each before it is begun. It also contains a question and answer section on each stage providing help for the most common questions. The book also uses a question and answer format to providing further advice on adopting the PA approach.

3 STUDY OVERVIEW

The application of PA came about as a result of LMU approaching a local financial organization, with whom we had previously collaborated, with the suggestion that we use the PA package to assist them in their own internal deliberations regarding process improvement. The work was not done on a consultancy basis, but on the understanding that we could use the experience of our use of PA for our teaching and research. The organization is the headquarters of a group of financial companies, with a centralized software development staff of around seventy people, dealing with systems and projects for the group. Many of the systems involve large daily volumes of transactions. One of the development personnel had previously used part of the SPMM; seeking to incorporate ideas of 'best practice' in a specific project and then producing a report rating the actual practices used, and giving an overall calculation of the 'cost of quality'. The results of this indicated that the processes used on that project could be rated at level three of the SPMM, and that the payback expressed in the form of a ratio of 'expected rework' to 'actual rework' was in the order of 10:1. (The approach used to measure 'rework' was based on Error Source Accounting, see Bryant & Chan, 1992). These fairly impressive results had raised the visibility of process improvement issues amongst the personnel, and probably acted as a deciding factor in the organization's decision to allow outsiders to apply the PA assessment.

The initial task was to identify the organization personnel who would liaise with the external project team (the authors). Although the bulk of the effort would be supplied by the external team, an internal group was required to initiate and oversee the project. Three internal staff were selected, including a senior manager and the staff member who had previously applied SPMM to a project. The group met and decided on a project schedule. The internal members were responsible for selecting interviewees, and arranging a briefing meeting at which the aims and objectives of the project could be discussed with all those involved. The scope of the project was to be limited to the first task of PA; the assessment stage of the SEILC. This involved assessing the current practices, tools and methods of the organization, using the questionnaires provided by PA as the main instrument for data gathering. The PA Workbook advises that this, like all other stages, should involve a team of managers and technical staff to ensure different perspectives are taken into account, and also to promote involvement by the staff generally.

It was decided that a team of ten people would be asked to complete the questionnaires and be interviewed. Their job functions included -

- Project Leader
- Senior Development Manager
- Analysis & Design
- Program and Build
- Development (Quality) Manager

Business Development Manager

The team also included two senior managers from user areas of the company, to represent the customer view of the products resulting from the software development process. This combination provided a wide range of views from those participating in the software development process.

4 ASSESSMENT EXERCISE

The interviews were conducted by one of the authors, who also undertook the processing of the results from the questionnaires. The assessment exercise involves responding to three questionnaires; one being done in the respondent's own time, with the results recorded on paper, and two completed during individual face-to-face interviews. The paper questionnaire consists of a number of questions requiring either "Yes" or "No" responses - "Don't Know" and "Not Available" are also allowed. All ten participants completed this: They did this prior to the interviews, so that they knew the sorts of topics that the later interviews would cover. The questions are quite detailed and are designed to elicit information concerned with whether various activities are performed or standards followed. These questions are similar to those of the CMM (Humphrey 1987), and cover a number of aspects including 'Organizational policies' and 'Training and software development'. Since the answers are either Yes or No, they can be combined for all respondents, and give an overall score for each section. (PA includes, on disk, a template for a spreadsheet which produces the relevant ratings when the scores are entered.)

A qualitative set of questions is put to the interviewees to gain further details of the practices, methods and tools currently in use. This is an extension of the comment provision in the CMM questionnaire, which itself is mostly in a Boolean Y/N form. Each interview for this section took about one hour, and resulted in a great deal of information complementing the earlier responses. PA also offers guidance for follow-up questions and inferences to be drawn from the detailed information that is acquired.

Finally, there is a set of quantitative questions regarding matters such as budgets and future plans. It was felt that this required specialized knowledge not available to most of the target group. Only the Senior Development Manager, who was responsible for most of these areas, answered these questions. Again, a set of inferences and follow-up questions are supplied to guide the PA user.

The interviewing itself caused some anxiety on the part of the target personnel, but this was dissipated when the confidentiality of their answers was emphasized: But this had to be restated a number of times. Respondents expressed some difficulty in deciding upon their answers to the Boolean questions, but the PA guidelines account for this, suggesting that such questions should only be answered with a 'Yes' if -

'A - The question is true for a significant majority of the cases or situations (normally 70% or more); B - the majority of knowledgeable people within your organization would also respond 'yes'; C - an independent party could substantiate that the answer is true by using specific project evidence.'

As is common with projects requiring co-ordination of many people, with different priorities and objectives apart from those of the project itself, the project faltered at this interview stage. The project team had been keen to work to a tight schedule. It had been hoped to be able to go from the initial briefing, through to completion of the interviews in a three week period: And completion of this first stage project within one month of that. The idea was to demonstrate to participants that their efforts had been useful, and the overall results of the project would yield meaningful results and tangible effects. Unfortunately various people were unavailable at the pre-arranged interview times, and the period for this activity had to be extended. The Senior Development Manager was the hardest to get hold of, an early indication perhaps of lack of management commitment to the project.

5 PROCESSING THE RESPONSES

Upon completion of all the questionnaires and interviews the results were then processed. For the Boolean questions PA does not state whether the overall ratings should be reached by consensus or by averaging across all the results. The project team decided that an average should be taken, although the range of responses would be noted. The average is then plotted against scores supplied by PA for 'common' and 'best practice'. (These scores derive from Pressman's use of PA on a range of software development organizations.)

The attributes measured by this particular questionnaire correspond roughly to the sections covered by the CMM. (CMM for instance uses terms such as 'organizational and resource management', 'training, standards & procedures', 'software engineering process metrics' and so on.) An average score for each process attribute is obtained, which results in a grade corresponding to the maturity rating for that attribute. A summary of the scores obtained in our example is shown in Figure 1. The column headed '*actual*' gives the average score attained for that process attribute, or sub-section. The CP column indicates 'common practice'; the BP column 'best practice'.* The resultant grade for each aspect is shown in the final column.

Pressman warns against using the attribute scores and the overall ratings too simply. Each software development company is unique, has different demands from its own customers and its own management structure and personnel. Scores and comparisons must not be treated as absolute and objective measures, but at best as indicators. They can provide a guideline, but should not be allowed to form the focus of the exercise, with the sole aim of improvement being to reach a higher score as an end in itself.

Using PA for a group drawn both from users (customers) and developers is slightly problematic. It is unrealistic to expect users to have detailed knowledge of the software development process. Therefore many of their responses will be "don't know". But for PA scoring a "don't know" scores '1', the same as that for a "no". The net result of a large proportion of "don't knows" would be poor comparison for that attribute against common or best practice. PA offers no suggestions to deal with this, probably on the assumption that users/customers are not expected to be involved. In our case study there were a large number

* Common practice is typical of most software development. Best practice represents the top 10-15 per cent of software developers. Common practice is inadequate in the software industry today, whilst best practice is still short of state-of-the-art practices and thereby has scope for improvement

	Section	Actual	CP	BP	Grade
1	Organizational Policies	1.35	1.8	3	E
2	Training	1.49	2.2	3.7	E
3	Software Development Process*	2.17	1.4	3.1	D
4	Quality Assurance Activities	1.93	1.4	3.1	D
4.1	Documentation	2.78	1.6	3.9	
4.2	Reviews & analysis of results	1.77	1.4	2.6	
4.3	Quality assurance functions	1.23	1.3	2.8	
5	Project Management*	2.0	2.0	3.6	D
5.1	Organizational resources	2.56	1.6	4.4	
5.2	Oversight	2.33	2.7	3.7	
5.3	Planning	1.67	2.0	3.1	
5.4	Monitoring & tracking	1.84	1.4	3.4	
5.5	Configuration management	2.12	2.0	3.8	
5.6	Subcontracts	1.61	2.5	3.0	
6	Methods and Techniques	1.98	2.3	3.8	D
6.1	Customer communication	1.85	2.7	4.5	
6.2	Software engineering methods	2.1	2.2	3.1	
7	Tools	1.9	1.4	3.0	D
7.1	Categories	2.5	1.7	3.3	
7.2	Environment	1.38	1.0	2.8	
8	Metrics & Measurement	1.09	1.0	2.4	E
	Overall	1.74	1.69	3.21	D

Figure 1 Grading Software Engineering Practice.

* denotes an average score based on a wide range of individual scores

of “don’t knows”, which did affect scores for some attributes; but did not alter the grade for that attribute, or the overall grade.

The attribute grade gives an indication of the organization’s current practice for that attribute, and also an intimation of how best to start to improve it. The overall grade is based on the profile of all the attributes. PA suggests that a grade ‘D’ indicates that the organization has -

- few controls in place
- a relatively informal software development process
- some procedures in use, but these are incomplete, not fully defined, and allow a range of interpretations across and within projects
- some methods have been identified, but they are not always used (correctly)
- some CASE tools might be in use, but are not integrated into the development process

This is a fairly accurate description of the situation existing in the organization studied.

The descriptions of the five possible grades offered by PA are not meant to be more than broad approximations. They are based initially only on the one form of questionnaire. But this does permit a simple form of validation, since there should be some correspondence between the description offered by PA, and the context in which the questionnaire has been applied. In our case the project team were satisfied that the ‘D’ rating was reasonably accurate.

For each process attribute PA offers an outline of initial improvement activities. For example, with regard to the grade ‘E’ for ‘Organizational Policies’: PA suggests that this indicates a lack of top down direction for software engineering practices. This is usually because senior management has not identified such practices as a critical aspect. The suggested remedies include: educating senior management on the importance of software to their business; affirming this view to middle and line management; getting senior management to commit resources to process improvement initiatives. These suggestions from PA are then combined with the findings from the other two questionnaires; the ‘qualitative’ and ‘quantitative’ interviews.

5.1 Sample Findings

The processing of the interview results is time consuming, and requires skill and judgement on the part of the project team. The disinterestedness of people external to the organization itself is critical: Both earlier in the interviews themselves, and in this processing of the responses. Essentially what is required at this stage is for the interview material (tapes, notes, etc.) to be analyzed, highlighting key points and issues. If this were to be done by someone within the organization, there would be a high possibility that individual prejudices would interfere with the process. One of the authors worked through all the interview material, and produced a draft report with a number of findings.

The initial list of findings was then used as a basis for a second pass through the material, correlating responses to the findings, whether they supported that finding or not. (It can be as useful to know that people are unsure or differ about something as it is to know if something is working well or not.) The attribute scores were also a useful guide, particularly where there were a wide range of individual scores. A total of 40 findings resulted as shown in Figure 2.

FIGURE 2 - Findings from PA Assessment**Organizational Policies**

1. Lack of management commitment to improving the development process.
2. No clearly defined software development process exists.
3. No clearly identified quality and productivity improvement program exists.

Training

1. Training is ad-hoc for staff when new practices, methods or tools are implemented.
2. The lack of a training strategy is having a detrimental effect on staff.
3. Training is not clearly specified for software engineers and project managers.
4. Project planning does not take into account training requirements for project staff.
5. Staff find it hard to keep up-to-date with latest developments in the industry.
6. Project managers are seen as having little experience of handling large-scale projects.
7. Management are seen as requiring training in staff management and motivational skills.

Software Development Process

1. The development process is not defined and monitored to ensure consistency of approach.
2. Standards that have been established are not followed by all staff.
3. Testing seems to be planned for some projects and not for others.
4. Projects on different platforms employ different testing methods.
5. Requirements capture is seen as an area that is not given great enough emphasis.

Quality Assurance Activities

1. There are specific documentation formats defined, but not all staff are aware of this.
2. Configuration management is not used for all documentation.
3. Formal technical reviews are neither formal nor conducted regularly.
4. There is no analysis of the review data.
5. The Q. A. procedures are not applied consistently.

Project Management

1. The estimating process does not identify risk areas.
2. Timescales for projects are not seen in the same way by users and developers.
3. Project management is not consistent over all projects.
4. Project initiation stems from a number of sources, projects vary in importance and size.
5. The project management process is seen as de-motivating staff.
6. Users do not understand their role in the project management process.
7. There is little emphasis on a review of projects upon completion.
8. Project Management of subcontracted work is seen as managed differently.
9. The development effort spent on maintenance activities is 23.8% of the total workload.

Methods and Techniques

1. No established team to monitor and assess the development process on a long-term basis.
2. The user is not involved regularly during the development process.
3. The methods employed are not as comprehensive as they could be or utilized consistently.
4. Software developers are lacking in knowledge of best practices in software engineering.
5. Development staff have different views upon who the customer is.

Tools

1. CASE tools are not being used to any great extent.
2. Some tools are not being used to their fullest capabilities.
3. There is no strategy for the evaluation, purchase and implementation of tools.
4. Prototyping is sometimes used, but not often.

Metrics and Measurement

1. Metrics are employed but have only recently been introduced.
Staff are largely unaware of the benefits to be gained from the use of metrics.

Each finding was then stated in the assessment report, together with a discussion and recommendation. An example entry for the Software Engineering Process attribute is given below.

FINDING: *The software development process is insufficiently defined and monitored to ensure consistency of approach.*

DISCUSSION: *The software development process is partially defined in that procedures exist for certain elements, e.g. configuration management. These elements are not always employed as they should be, e.g. independent software quality assurance is not employed on all projects, through lack of publication, auditing and management commitment in varying degrees. It is interesting that the score given for this section by the users involved in this exercise was markedly lower than that given by the development staff who participated. Even development staff who identified the development methods employed as being a strong point in the company's favour also pointed out that these methods were not consistently employed.*

RECOMMENDATION: *There is a need for the software development process to be defined and documented. The documented process should identify milestones, deliverables (and their assessment) and control points (i.e. reviews, review procedures, user involvement) that ensure consistency of approach across all projects. Then a quality management system should be applied to the whole software development process. This would give a process based on established procedures, a manual that all staff could have access to, or copies of the relevant parts, and a solid base for auditing projects. It would also provide a good platform for decisions on how to improve the process, methods and tools employed.*

At this stage Pressman recommends that all the findings are prioritized, forming the basis for an action plan for later stages of PA. The prioritization was carried out by the entire project team, since some internal steering was essential. The team, however, had great difficulty in accomplishing a list of priorities based on the findings; and found it easier to focus on the recommendations. This was a slight departure from the prescribed PA route which provides a template for the prioritization report. One of the key factors in the decision to present the results in this way was that it would enhance the accessibility to those unfamiliar with the technicalities of the exercise, particularly senior management. This change in emphasis caused the project to overrun, but resulted in an assessment report that met with the approval of all of the project team, with a prioritized list of recommendations that would provide a strategy for process improvement for the next eighteen months (a time frame recommended by PA).

6 ASSESSMENT EXERCISE COMPLETION & RESPONSES

The exercise was completed with briefings to senior management and the participants; outlining the findings and recommendations. The assessment report was made available to both groups. At the outset, there had been the expectation that the earlier 'level 3' rating for the individual project would be matched across the department as a whole. The 'D' roughly

equates to the lower level 2, and some attributes only rated 'E' (or roughly level 1). This failure to match expectations had extensive repercussions. The senior development manager who had initiated the project, and was part of the project team, was in a particularly difficult position. The findings showed that recent initiatives to create a more open and quality-minded organization had proved ineffective. Also that improvements in methods and practices had been introduced, but there had not been any check to see if they were understood, working effectively, or indeed being used properly or at all.

This meant that the process envisaged by PA was undermined. Instead of continuing to the education and selection phases, the report led to an overall questioning of recent policy and control. Various personnel felt threatened by the findings, and many questioned the wisdom of allocating resources to accomplish what should already have been achieved by earlier initiatives. Rather than providing a mechanism for increased awareness and resource commitment, the assessment stage threatened to subvert the whole idea of process improvement.

At the time of writing, this combination of factors has meant that there is no immediate prospect of continuing the project to later PA stages. Officially this hiatus is caused by the arrival of a manager responsible for all quality matters. Internally some have seen this as indicating that management have started along the path to process improvement, but again demonstrated that they are not yet fully committed to allocating the necessary effort and resources. The effect of this failure to act on the software development staff, and particularly those involved in the exercise, has been severe. The danger is that any future attempts will be clouded by this negative experience. This aspect has been grasped by the Information Systems Director, who seems determined to maintain some momentum, and so the project may progress in the near future. If it does so, the intention is to continue with PA, and our case study will move on to the Education and Selection stages.

7 EVALUATION OF PROCESS ADVISOR - ASSESSMENT STAGE

PA is marketed as a self-directed system for software process improvement. The workbook and video are meant to guide people through this, and their structure and content are well suited to the purpose. On the other hand, the earlier metaphor of a *Do-It-Yourself* kit is valid. It may be cheaper to use a DIY approach, but it does not obviate the need for adequate preparation, sufficient resources, and a critical assessment of progress and results. Although we have a few criticisms of PA, from our limited use of the assessment stage, our conclusion is that more thought needs to be given to the pitfalls of using a self-directed approach to something as important as process assessment. PA does not come with any 'health warnings', and it would be naive to expect it do so. On the other hand, many of the points we make below apply to any process improvement project, whether done with PA, CMM or similar.

The importance of management commitment to the exercise itself cannot be over-emphasized. PA does not really stress this, perhaps assuming that purchase of the product is a useful indication of commitment; although this is unlikely since it is not expensive. The case study exemplifies the problems of failing to elicit and *sustain* commitment. The project was initiated by the senior development manager, with encouragement from above. But this may have been because there was the expectation of a high rating. Even before the findings were produced, there was a lack of visibility by management as the project progressed; and this was seen by some as evidence of a lack of interest. The delay in producing the report, and the

absence of senior management from the de-briefing session for participants were also unfortunate aspects. Now that the assessment stage has been completed, but with no obvious progression, there is a danger of a loss of credibility, with participants wondering what all the fuss was about.

PA cannot be held to blame for this: On the other hand we do not believe that our experience is unique in this sort of project. Perhaps PA should devote more attention to the pitfalls of a self-directed program, with guidance on how best to avoid some of the more likely problems. It is ironic that PA stresses the importance of management commitment to the software engineering process, but does not stress that this is equally essential to the success of using PA itself. Moreover this cannot be an initial enthusiasm, but must be sustained throughout the project. The problem the case study organization will have is in re-gaining the enthusiasm of those involved if and when it is decided to continue with the process. Rather than having shown continued commitment to improving the situation, which impressed everybody at the beginning, management have prevaricated and seemed to indicate that they do not wish to improve, or feel incapable of improving, the software development process. Perhaps a cautionary note to the effect that management might not come out of the exercise as well as they expected might have eased the way for reception of the actual findings and so prevented the loss of momentum.

Another aspect not emphasized by PA is the importance of selecting the personnel to take part in the assessment exercise. PA suggests that a team of managers and technical staff complete all the stages together. This does not help identify the specific staff to involve, nor the qualities they should have. The project team for the case study decided that staff at all levels in the software development process should be involved, and this was reflected in the roles of the ten people who completed the questionnaires and were interviewed. It was also decided that people with both a positive and a negative attitude to the process should be included. With no prior knowledge of the individuals involved, it is difficult for the authors to say whether they were representative in this respect; but it is true to say that their contributions were not always what the organization may have expected.

Another decision we took was to ensure that users (customers) were represented. This was done by involving senior managers from external departments, since it is they who initiate most development projects. At first they were a little unsure of their role, but it was made clear that it was their responses as users of the software, and their involvement in the development process itself that were required. Both those selected contributed greatly to the findings and certainly widened the view obtained of the current state of software development within the organization. PA does not stress this user involvement, possibly because users are thought of as external to the organization; although some of the attributes assessed clearly involve users - e.g. customer communication.

One area where the project team failed was in not involving all senior managers involved in software development. It was suggested at the beginning that all relevant senior managers should be briefed, and should take part in the questionnaires and interviews. However, only one 'friendly' senior manager, apart from the two user senior managers, did actually take part. While nothing may have been lost in terms of their contribution, they may well have provided more impetus to the whole project and prevented the subsequent loss of momentum.

PA fails to highlight that the interview findings will be imprecise, and are likely to derive from the feelings of the respondents involved. This means that parts of the assessment report are not backed up by evidence, but may reflect general sensitivities, and will need further investigation in some areas. Although we were prepared for this, we were surprised at the opinions that were expressed. There was some discussion of the validity of these results, and how best they could be included in the report. The project team felt it was important to note and respond to such findings, since process improvement is as much a cultural change as it is a technical one.

One of the great strengths of PA is the advice given on what to do once the results are known. The descriptions of each grade provide a useful starting point for identifying the findings, and the associated questions are useful in obtaining supporting information. The comparison with common and best practice, whilst not being all-important, is also useful. Another point in the assessment stage is the identification of comparators for later use. These include the percentage of people working on maintenance, the turnover rate for software personnel, and also various annual budgets. These are useful figures in a general sense, but are often not used at all despite being easily available, as was found in the case study organization.

8 COMPARISON OF PA WITH CMM

In comparison with the CMM from the SEI (CMM, 1993), PA is a low-cost alternative. It does not give the specific measure that the CMM gives for process maturity; the figure it gives is a grade rather than the score. PA also goes further in its investigations than the CMM through use of the qualitative and quantitative questionnaires. This does widen the view of the process and so aid a better understanding of the current situation. (It is the use of the qualitative questionnaire that brought out many of the feelings from the respondents, as described earlier.)

CMM does, however, have advantages, the main ones being that it is a de facto standard within the industry, and also that it mandates use of independent personnel. Many of those involved in the case study, from the organization itself, felt that the objectivity and weight added to the project by the presence of the authors contributed greatly to the assessment exercise. It was felt that the presence of these independent personnel helped to create a true picture of the current software development process and how to improve it. But PA itself does not insist on this.

CMM is very expensive because of the use of SEI or SEI-accredited assessors. The benefit of this is that a rating can be obtained that, if favourable enough, can be made use of in marketing. PA on the other hand is purely for internal purposes. Another issue to go with the cost of CMM is that the investment prompts the organization to do something about the situation. Investing a lot of money and then not using the results would not be seen to be a good decision. In the case study example, perhaps the fact that the exercise involved costs in terms only of time and resources coupled with the poor showing of management in the results, made it easy to drop the project. The lack of investment in the exercise may well have contributed to its loss of momentum.

9 CRITIQUE OF PROCESS ADVISOR - AND OF PROCESS ASSESSMENT IN GENERAL

There are many available approaches to process assessment. Most of them involve great expense and high degrees of risk. PA, as a 'self-directed system' appeals to those organizations who want to, or are impelled to, embark on the route of process improvement; but who are uncertain what the progression will entail, and where they are starting from. PA offers a low cost, low risk starting point. On the other hand a PA project must be undertaken with careful thought and planning. The process of gathering the information must be well planned, and although it can be done by internal personnel, there are clear advantages in using disinterested, external people. What must also be realized is that there should be some thought given to the processing of the results, based perhaps on a small number of scenarios - e.g. the expected results, and a range of 'worst cases'.

This study only undertook the first stage of PA, Assessment. It proved very successful in providing a picture of the current software development process and helped to identify recommendations on how to improve it. Nevertheless, PA does have its pitfalls, notably that it is meant to be self-directed and that this may cause problems in terms of the integrity of the results and how they are handled. Independent personnel would not only be useful but may be necessary to ensure project success. It does however, provide a good starting point for any organization wishing to gain experience in process improvement at low cost.

Despite the problems encountered by our target organization, we hope to continue the project to the later stages of PA. Furthermore we hope to extend our work to investigate the practical issues surrounding the application of other forms of process assessment and improvement, including CMM, Trillium, SPICE and Bootstrap.

REFERENCES

- Bootstrap(1994), Bootstrap: Fine-tuning process assessment, Haase, V., Messnart, R., Koch, G., Kugler, J., Decrinis, P., *IEEE Software*, volume 11, no. 14, July 1994
- Bryant, A. (1995), *Standardizing SSADM: Methods, Standards & Maturity*, McGraw Hill
- Bryant, A. and Chan, D.(1992) 'Error Source Accounting - An Approach to Measuring Information Systems Quality', *Software Management*, February 1992
- CMM (1993), *Key Practices of the Capability Maturity Model*, Version 1.1, Paulk, M., Weber, C., Garcia, S., Chrissis, S., Bush, M., CMU/SEI-93-TR-25, Software Engineering Institute
- Humphrey, W. S. (1989) *Managing the Software Process*, Addison-Wesley
- Humphrey, W. S. (1987) A Method for Assessing the Software Engineering Capability of Contractors, *Software Engineering Institute Technical Report*, CMU/SEI-87-TR-23
- ISO (1991), *ISO 9000-3: Guidelines for applying ISO 9001 to Development, Supply, and Maintenance of Software*, ISO

Pressman, R. S. (1992), *Process Advisor: A Self Directed System for Improving Software Engineering Practice*, R.S. Pressman & Associates, Inc

Pressman, R.S. (1993), *A Manager's Guide to Software*, McGraw-Hill, 1993

TRILLIUM (1994), *TRILLIUM A Model for Telecom Product Development & Support Process Capability*, Release 3.0, Internet edition, Bell Canada

Tony Bryant is currently Professor of Informatics within the Faculty of Information & Engineering Systems at Leeds Metropolitan University. Since 1989 he has initiated a series of projects aimed at incorporating the strengths of different methods and specification techniques under the general heading of 'methods integration'. The work of his Methods Research Group now extends into formal specification, process maturity, standardization, and IS methodologies. Originally a social scientist, Professor Bryant obtained his first degree from Cambridge University in 1975, and his PhD from the London School of Economics in 1980. In 1982 he obtained his MSc in Computing from Bradford University and then worked for a number of years in commercial software development before taking a post at Leeds Polytechnic (now Leeds Metropolitan University).

Following a BSc in Computing, John Grogan became a Research Assistant at Leeds Metropolitan University working on "Total Quality and Information Systems Development and Operations". Initially from an engineering background he later moved into system development and operations. His experience in project/quality management, and information systems development has been utilized by commercial organizations in both the engineering and IT industries. He is a Licentiate of the Institute of Quality Assurance, a Graduate Member of the British Computer Society, a member of the UK Academy of Information Systems, and also a Member of the British Computer Society Quality Special Interest Group.