LANCASTER UNIVERSITY

Computing Department

Improving the Requirements Process

Pete Sawyer, Ian Sommerville and Stephen Viller

Cooperative Systems Engineering Group

Technical Report Ref: CSEG/30/1997

http://www.comp.lancs.ac.uk/computing/research/cseg/97_rep.html

Submitted to The Fourth International Workshop on Requirements Engineering: Foundation for Software Quality, June 8-9 1998, Pisa, Italy.

ABSTRACT

The state-of-the practice in requirements engineering is currently such that organisations wishing to improve their requirements processes find it hard to discover, evaluate and apply good practice. Good practice certainly exist but dissemination of practical experience is poor. Standards coverage of the requirements process is also patchy. This paper describes the *Requirements Engineering Good Practice Guide* which attempts to fill this gap by disseminating good requirements practice within a process improvement framework. The work is motivated by the authors' judgement that it is timely to exploit industry's interest in software process improvement as a vehicle for raising the profile of good requirements practice.

CSEG, Computing Department, Lancaster University, Lancaster, LA1 4YR, UK

Tel: +44-1524-65201 Ext 93799; Fax: +44-1524-593608; E-Mail: julie@comp.lancs.ac.uk

http://www.comp.lancs.ac.uk/computing/research/cseg/

Improving the Requirements Process

Pete Sawyer, Ian Sommerville and Stephen Viller
Lancaster University
[sawyer, is, viller]@comp.lancs.ac.uk

Abstract

The state-of-the practice in requirements engineering is currently such that organisations wishing to improve their requirements processes find it hard to discover, evaluate and apply good practice. Good practice certainly exist but dissemination of practical experience is poor. Standards coverage of the requirements process is also patchy. This paper describes the *Requirements Engineering Good Practice Guide* which attempts to fill this gap by disseminating good requirements practice within a process improvement framework. The work is motivated by the authors' judgement that it is timely to exploit industry's interest in software process improvement as a vehicle for raising the profile of good requirements practice.

1. Introduction

Many organisations have made a commitment to software process improvement (SPI) based upon, for example, the SEI's Capability Maturity Model (CMM) for Software [Paulk 93]. Many more organisations have taken steps to comply with the ISO9001-3 [Johnson 93] quality standard which shares many of the same aims as SPI. Although SPI programmes are often accreditation-driven, strong evidence that SPI yields real economic benefits [Diaz 97] is increasingly causing organisations to view SPI as a tool for gaining commercial advantage.

SPI emerged during the late 1980s and its influence on the industry’s practice, particularly in North America, has been growing ever since. Requirements engineering (RE) has also received much industry attention and research funding during this period. However, the impact of this on industrial practice compares poorly with that of SPI. This prompts the question: why is there a difference? The need to gain accreditation works as a driver for SPI but we don't think that this completely explains the disparity.

The question interested the REAIMS project (Requirements Engineering Adaptation and IMprovement for Safety and dependability). Our conclusion was that we, as RE researchers, consistently underestimate the effort needed to adopt a new technique and integrate it into an existing process. We seldom recognise the risk involved in selecting and applying even a mature RE technique on real projects. Inevitably, as schedules are squeezed, even long-recognised measures such as requirements tracing become vulnerable to neglect [Davis 95, Ramesh 95]. The result is that, for too many organisations, RE continues to be the most risky and intractable aspect of system development. No software process, whatever its "capability", can keep delivery times, costs and product quality under control if the requirements are poorly formulated or unstable.

Clearly, these were problems which would also face the ultimate acceptance of the novel techniques and tools developed in REAIMS (e.g. [Sommerville 98]). A workpackage of REAIMS was therefore assigned to addressing the basic process problems impeding organisations’ ability to exploit good RE practice. We reasoned that if the principles of SPI could be applied to RE processes, they would offer a practical means to substantially improve the state-of-the-practice.

Many organisations have reached the same conclusion and have attempted to apply SPI to their requirements processes. Unfortunately, while requirements engineering issues are touched upon by the CMM and others, it is generally addressed in insufficient detail to permit systematic detection and rectification of weaknesses. This is borne out by a recent survey [ESPITI 96] of European companies' attitudes to SPI which showed that the participating organisations considered the principal problem areas to be the requirements specification and the management of customer requirements.

REAIMS' solution to this problem was the *Requirements Engineering Good Practice Guide* (REGPG) [Sommerville 97] which extends the principles of SPI to requirements engineering. The REGPG draws upon existing SPI models to define a framework for 66 requirements practices derided from existing standards, reports of current practice, and the practical experience of the REAIMS partners. However, unlike the CMM or ISO 9001-3, the REGPG is not intended as a standard or for accreditation purposes but as a practical, easily understood and easily applied guide.

2. RE Processes, Practices and Standards

Requirements engineering is concerned with the discovery of required properties (the *requirements*) and their transformation into a form which will serve as the basis for development of a product which will exhibit those properties. However, the determinism implied by this definition is very misleading. Requirements engineering isn't a discrete activity which is enacted at the start, concludes with the production of a specification and is followed by architectural design. For example, in market-driven product development, the life-cycle for successive software releases may be so short that the requirements process is constantly enacted in parallel with down-stream life-cycle activities.

2.1. An RE Process Model

Several studies [Boehm 94, Potts 94, Hutchings 95] strongly suggest that the requirements process is cyclical. Figure 1 illustrates a spiral model which has been abstracted from these studies and from the experience of the REAIMS industrial partners.

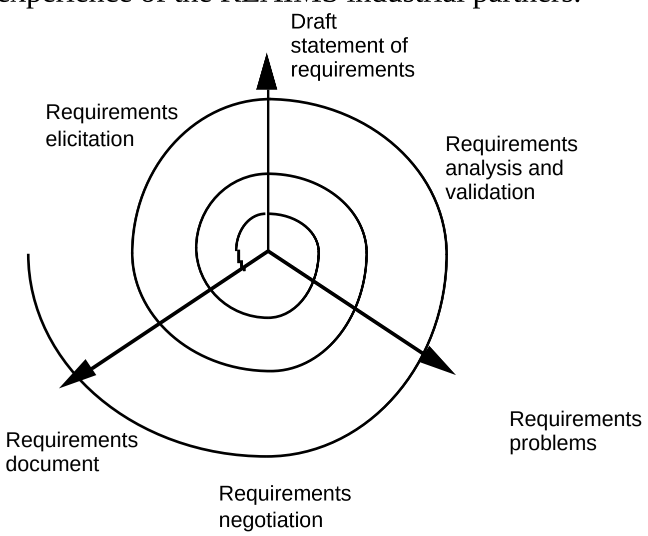

Figure 1. A generic requirements process model

The radial arms of the spiral represent both increasing cost and the generation of information by all three phases. The more iterations of the spiral, the better the quality of the requirements information but the more resources are consumed. This last point is important since, in reality, design activities cannot be deferred indefinitely while the requirements evolve slowly towards an elusive state of perfection. The three activities of each cycle are:

1. *Requirements elicitation* Given a statement of organisational needs and other inputs, different requirements sources (stakeholders, domain experts, operating regulations etc.) are consulted to understand the problem and the application domain. The resulting requirements may be incomplete, vaguely expressed and unstructured.

2. *Requirements analysis and validation* The requirements discovered during the elicitation phase are integrated and analysed. This is designed to identify problems such as missing information, inconsistencies and requirements conflicts.

3. *Requirements negotiation* Problems discovered during analysis need to be resolved. The analysts and stakeholders clarify their understanding and consider possible solutions. This may require negotiation to establish the necessary trade-offs. The elicitation of further requirements information and the initiation of a further cycle may be necessary.

Cutting across these three activities and subsequent development phases is *requirements management*. Requirements management permeates the whole development process and is concerned with coping with the emergence of new requirements information and with the inevitable changes to which requirements are subject. Two of the most important goals of requirements management are ensuring that requirements are traceable and the enforcement of change control.

2.2. RE Process Problems

The RE process is necessarily cyclical in order to cope with the characteristic problems of discovering and managing the requirements. These include:

- Requirements are hard to elicit. The customer may not have a clear view of what they need. The requirements engineer's job is not simply to "capture" the customer's requirements but to help the customer identify and articulate them. This is made difficult by the fact that the customer is seldom a homogeneous entity but is more often a collection of different stakeholders with different concerns and different foci on their task. Many requirements simply cannot be discovered without doing some analysis. Such emergent requirements usually occur as a consequence of the interaction of other requirements. Hence, the cost of addressing them cannot be predicted without an up-front investment.

- Requirements change over time. The customer's understanding of what they need may change, their business may evolve during the course of the development project or the competitive environment may force a radical reassessment of the product.

- The requirements process is constrained by time and cost. The requirements phase has a finite budget which is usually a small proportion of overall development costs. In large systems engineering projects, partitioning may take place very early with different subsystems' development proceeding before the requirements are fully understood. In such cases, system requirements errors are hard to rectify because they affect the system architecture. The consequence is often radical change to the software requirements late in the project life-cycle.

In a large project, these problems inevitably result in requirements being overlooked or poorly understood. One estimate is that 40% of requirements require rework during the course of a development project [Hutchings 95]. When problems emerge late in the development process, they may necessitate a radical reassessment of priorities as the need for redesign impacts on already agreed requirements. This is expensive and inevitably has knock on-effects as, for example, test plans have to be revised and value judgements have to be made about where resources should be reallocated.

Requirements problems are present to a greater or lesser degree on almost all projects despite the existence of techniques which help address them. In many cases these problems are due to requirements processes failing to manage the requirements information and failing to deploy the most appropriate techniques. Consequently, many systems default to solutions which reflect the developers' view of the problem rather than that of the customer [Potts 95]. Inadequate understanding of how to identify, resolve and manage requirements problems has led to requirements processes which are at best poorly able to cope with the problems and at worst, completely *ad hoc*. Consensus about what constitutes best practice in RE is relatively hard to arrive at (one reason why we prefer to talk in terms of *good* practice).

2.3. RE and the Role of Standards

As noted above, good practices exist which help address most requirements problems, yet the problems are still prevalent. The reasons for this seem to be a combination of:

- Poor awareness of what techniques exist.
- Lack of information about how to evaluate the suitability of new techniques.
- Lack of guidance on how to effectively adopt a new technique and integrate it within an existing requirements process.

However, pockets of good practice do exist where organisations have successfully adopted techniques and tailored them to their process. While dissemination of this experience is patchy, the experience nevertheless exists. In these circumstances is it normal for an industry to develop standards to help disseminate good practice. In RE, the industry is still in the early stages of doing this.

There are no widely-known generic RE standards, but there are several systems and software engineering life-cycle standards which cover requirements issues. One of the best known is the European Space Agency's PSS-05 [Mazza 94] which defines a six phase development life-cycle of which the first two phases, User Requirements Definition (UR) and Software Requirements Definition (SD), comprise the requirements process.

Standards such as PSS-05 contain much wisdom. However, they are principally concerned with defining good practice and process activities rather than helping organisations in their adoption. There is strong evidence [Lubars 93, El Eman 95a] that relatively few organisations meet the standards. One reason for this is that the adoption of new practices incurs changes to organisational processes. This is difficult because it often incurs up-front expense (on training, tool investment, etc.) and may have knock-on effects which are hard to predict.

SPI addresses this problem by providing guidance on the implementation of different practices within a framework (or *improvement model*) designed to allow them to be implemented incrementally. Incremental implementation allows the effect of practices' implementation to be assessed in advance and evaluated after the event. A further benefit is that incremental implementation of practices allows the improvement model to be structured as a series of steps which recognise the value of implementing basic practices but also provide a route to greater improvements by the adoption of increasingly targeted practices. One way of characterising this is: if software/systems engineering standards distil industry experience of how to perform a process, SPI standards distil industry experience of how to change a process.

Humphrey's pioneering work on SPI in the 1980's [Humphrey 89] resulted in the SEI's CMM for software. The CMM is structured around 5 *maturity* levels. Process maturity represents the degree to which a process is defined, managed, measured, controlled and effective [Paulk 93]. The more mature a process, the more it is possible to accurately forecast and meet targets for cost, time of delivery and product quality. In the CMM, achieving this predictability is the goal for organisations with immature processes. As processes become more mature, the range within which results fall is narrowed. Eventually, at high maturity levels, it becomes possible to set and achieve more ambitious targets. The emphasis moves from understanding the software process, through exerting control over the process, to achieving on-going improvement.

The CMM maturity levels range from level 1 (Initial) which is an ad hoc, risky, process to level 5 (Optimising) where a process is robust and subject to systematic tuning using data from completed projects. A level 1 organisation will find that costs and timescales vary widely; a level 3 organisation will be able to make predictions with a high degree of confidence for similar kinds of projects; and a level 5 organisations will be able to tolerate novel projects gracefully.

The CMM has a *staged architecture*. Each maturity level has a focus which is supported by a number of key *process areas*. For example, level 2 (repeatable) focuses on project management. Accordingly, the key process areas are: Requirements management, Software project planning, Software project tracking and oversight, Software subcontract management, Software quality assurance, and Software configuration management. For each of these, the CMM describes a set of key practices. The process areas effectively set capability goals which should be met if the supporting practices are adopted and standardised.

Level 2 is interesting because its focus on project management means that it is concerned with controlling a process. As such, it represents the minimal standard which an organisation needs to control risk. Beyond level 2, the practices should help, not only to narrow the range of predicted costs, but also to make the range begin at a lower value. In other words, the efficiency of the process should increase along with the ability to plan accurately.

Of course, the CMM is not the only SPI model. Others exist or are under development (e.g. the ISO/IEC 15504 draft standard - formerly known as SPICE - [Rout 95]). In addition, ISO 9000 (in its software quality form: ISO9001-3) is commonly regarded as an SPI model, despite being a basic standard for quality rather than a strategy for process improvement (see [Paulk 94] for a comparison of the CMM and ISO 9000). The CMM has been influential on our work because it is the most widely used SPI model and has been shown to lead to real economic benefits [Herbsleb 97].

However, the experience of the REAIMS industrial partners and others [Hutchings 95] is that few process attributes effectively address the requirements process. For example, requirements management is a process attribute (at level 2) but other requirements activities are not, nor are they supported by key practices.

During REAIMS it became apparent that there was a substantial amount of knowledge about good requirements practice in the form of software/systems engineering standards and partners' experience, but little advice on how best to use it to improve RE processes. The REGPG was developed to fill this gap by applying the principles of SPI to the RE process to help improve organisations' leverage over the requirements process.

3. The REAIMS RE Process Maturity Model

The ain of the REGPG was to promote the improvement of organisations' requirements processes and to do so within a framework compatible with existing SPI models.

Like the CMM, the REGPG adopts a framework of multiple process maturity levels. The CMM has demonstrated that maturity levels provide a reference framework which sets out a clear strategy for process improvement. However, a 5-layer framework was not appropriate for the REGPG. This is because the current state of the practice of RE [Lubars 93, El Eman 95b] makes it doubtful whether any RE processes exist which, in CMM terms, could be characterised beyond *Defined* (level 3). Certainly, we know of none from which we could extract generic principles. This situation is improving but, because of the problems outlined above, still lags behind the general levels of maturity achieved elsewhere in the development process.

For these reasons, the REGPG defines 3 maturity levels (Figure 2):

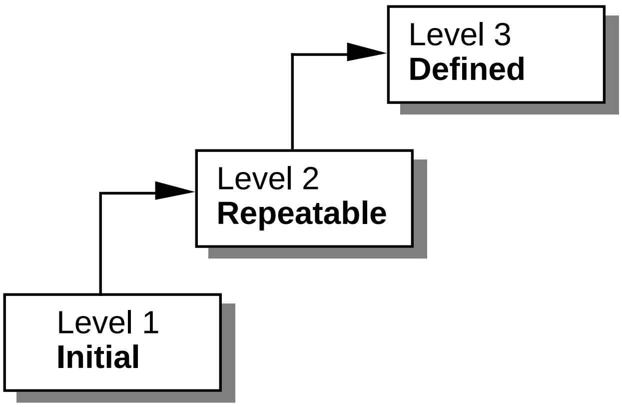

Figure 2 The 3-level REAIMS process maturity model

- *Level 1 - Initial level* organisations have an *ad hoc* requirements process. They find it hard to estimate and control costs as requirements have to be reworked and customers report poor satisfaction. The processes are not supported by planning and review procedures or documentation standards. They are dependent on the skills and experience of the individuals who enact the process.

- *Level 2 - Repeatable level* organisations have defined standards for requirements documents and have introduced policies and procedures for requirements management. They may use tools and methods. Their documents are more likely to be of a consistent high quality and to be produced on schedule.

- *Level 3 - Defined level* organisations have a defined process model based on good practices and defined methods. They have an active process improvement programme in place and can make objective assessments of the value of new methods and techniques.

These have an analogy to the CMM. Levels 1 and 2 correspond approximately to those of the CMM. Level 3 corresponds approximately to level 3 and above of the CMM. That is to say, if requirements processes exist which are *Managed* (CMM level 4) or *Improving* (CMM level 5), we have not seen them and would not know how to recognise them.

3.1 Good Practice Guidelines

A process's maturity level is determined by the practices employed in the process.

The REGPG describes 66 *good practices*. These have been derived from existing standards, reports of requirements practices (e.g. [Forsgrem 95, El Eman 95a]) and the experience of REAIMS partners and others. Within these, we have recognised that while consensus exists on the genericity and utility of many practices, the value of others are more project, organisation or application domain-dependent. Similarly, while some practices are within the scope of immature organisations, some practices must be underpinned by other measures or require specialist expertise. To reflect this, the practices are classified according to whether they are *Basic, Intermediate* or *Advanced*:

- Basic practices represent fundamental measures which underpin a repeatable process by, for example, defining documentation and basic management standards. Basic practices should almost always be the first to be adopted.

- Intermediate practices are typically more complex but help make the process more systematic by, for example, using methods for conceptual modelling. Intermediate practices usually have to be underpinned by basic practices in order to be effective.

- Advanced practices are practices which require substantial specialist expertise or which support continuous improvement. Advanced practices include practices which are of most benefit in specialist domains. They usually have to be underpinned by basic, and occasionally, by intermediate practices.

Clearly, the classification we have chosen for some of the practices will be controversial. However, we have tried to counter what we believe is the natural tendency to underestimate the difficulty in applying many seemingly straightforward practices.

To help with evaluating the good practices, they are presented as guidelines. Each guideline provides a qualitative assessment of:

- The key benefits of the practice. This indicates how the requirements process should be improved by adopting the practice.

- The cost of introducing the practice. This provides a qualitative indication of the level of effort and investment needed to integrate the practice in an existing process in terms of, for example, staff training, support systems, etc. It is important to distinguish implementation costs from application costs. For example, *in the short term*, it may not be practical to introduce a practice which addresses a key problem area and has low application costs if it requires extensive training of staff already working to tight deadlines on a project.

- The cost of applying the practice. This provides a qualitative indication of the level of effort in using the practice effectively once it has been introduced. Clearly, some practices consume additional resources so this is intended to help an organisation do a cost benefit analysis on their process improvement measures.

Good practices are analogous to the CMM key practices with the important difference that they are not rigidly associated with maturity levels. Good practices are associated with process areas (products or activities) to which they contribute. These are:

- The requirements document.
- Requirements elicitation.
- Requirements analysis and negotiation.
- Describing requirements.
- System modelling.
- Requirements validation.
- Requirements management.
- Requirements engineering for critical systems.

REGPG maturity levels do not focus on selected process areas to the exclusion of others. We don't mandate, as the CMM does, which process areas should be addressed in order to achieve (e.g.) a repeatable process. Sometimes it makes sense to concentrate resources on a particularly weak process area but if there are weaknesses across the RE process, prioritising of improvements needs to be more flexible. For example, the following practices listed in table 1 are recommended to support requirements management:

Good Practice	*Cost of introduction*	*Cost of application*	*Guideline classification*	*Key benefit*
Uniquely identify each requirement	very low	very low	Basic	Unambiguous references to specific requirements are possible
Define policies for requirements management	Moderate	Low	Basic	Provide guidance for all involved in requirements management
Define traceability policies	Moderate	Moderate-high	Basic-intermediate	Leads to consistent traceability information being maintained for all systems
Maintain a traceability manual	Low	Moderate-high	Basic	Acts as a central record of all project-specific traceability information
Use a database to manage requirements	Moderate-high	Moderate	Intermediate	Makes it easier to manage large numbers of requirements
Define change management policies	Moderate-high	Low-moderate	Intermediate	Provides a framework for systematically assessing change proposals
Identify global system requirements	Low	Low	Intermediate	Finds the requirements which are likely to be most expensive to change
Identify volatile requirements	Low	Low	Advanced	Simplifies requirements change management
Record rejected requirements	Low	Low	Advanced	Saves re-analysis when rejected requirements are proposed again

Table 1 Requirements management good practices

As part of an improvement plan, an organisation with a level 1 process should consider implementing the first 4 practices. However, implementing any of the other 5 practices would not normally be practicable. Starting from a low maturity base, basic practices from other areas will usually offer a better return.

In many cases, the costs of implementing and applying basic practices are relatively low. However, as table 1 illustrates, this is not always the case (requirements management is slightly unusual in this respect) and substantial investment may be necessary to attain a repeatable process. It is not necessary to implement *all* the basic practices before intermediate or advanced practices. We recognise that the utility of practices will differ according to factors such as the application domain or customers' working practices.

3.2. Assessing Processes and Implementing Improvements

Effective process improvement requires that the baseline from which improvement is to commence is known. This necessitates some means for assessing the process. One approach is to try to build a detailed model of the process. However, this is often hard to acquire since people often have different perspectives on the process and process documentation is often a poor reflection of actual practice [Rodden 94]. A more pragmatic approach is often taken, therefore, by evaluating the process against a checklist of process areas and practices.

This kind of process assessment can be used in two ways:

1. Assessment is used to characterise the process maturity level within the overall improvement framework. The process areas which must be addressed in order to move to the next maturity level are then identified and practices which address these process areas are introduced. This approach works best in a staged architecture where process areas are associated rigidly with maturity levels. Within process areas, particular weaknesses may not be distinguished from other, less pressing problems and improvements may only address them indirectly. However, it has the merit of imposing conformance across processes and organisations.

2. The results of the assessment are used to identify the specific weaknesses in the processes. Improvement efforts can then be focused upon these by selecting practices which directly address these weaknesses. These can result in improvements more closely tailored to the organisation rather than being oriented to accreditation. However, it is harder to apply this kind of assessment in the context of an improvement framework because processes maturity levels are harder to evaluate.

In practice, these two uses of process assessment are complementary. The first has been proven to be workable by the CMM where comparison of many disparate organisations has, for the first time, been possible. The merits of the second approach have, however, been recognised by ISO/IEC 15504 which is attempting to develop a less rigid association between process areas and maturity levels. We have tried to accommodate the merits of both in the REGPG. We have had the luxury of being able to do this because the REGPG is not intended for accreditation.

The REGPG assessment approach is to rapidly gain an overall view of a process and the extent to which it repeatable or defined. The requirements practices used in the process are checked against a checklist of the REGPG good practices. This does not build a detailed model of a requirements process but does reveal what practices are in use and the extent to which they are used. This allows us to position the process within the improvement framework and identifies areas where the use of good practice is weak.

In a large organisation, the extent of good practice use will vary according to project, engineer, customer, etc. To accommodate this inevitable variation, each good practice is assessed as being:

1. *Standardised* The practice has a documented standard in the organisation and is followed and checked as part of a quality management process.
2. *Normal use* The practice is widely followed in the organisation but is not mandatory.
3. *Used at discretion of project manager* Some project managers may have introduced the practice but it is not universally used.

4. *Never* The practice is never or very rarely applied.

To help perform this analysis in a large organisation, the REGPG recommends the following assessment process:

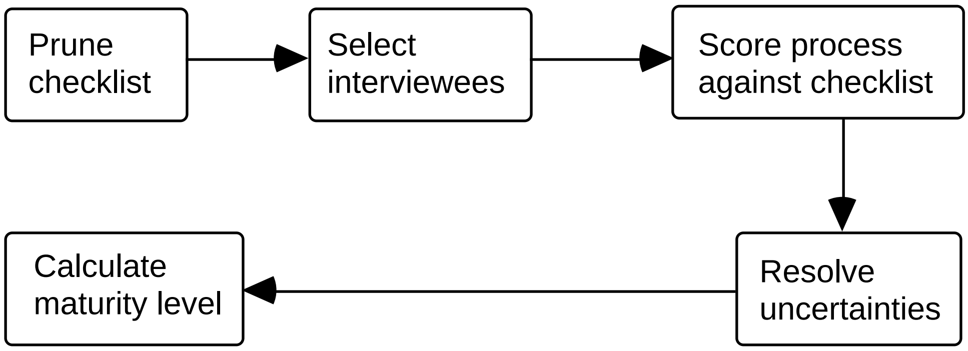

Figure 3. Process maturity assessment

The activities involved in the maturity assessment process are:

1. *Prune guideline checklist* Identify and eliminate practices which are obviously never used. In most cases this will substantially reduce the number of questions to be asked.
2. *Select people to interview* An accurate assessment of the extent to which practices are used will depend on who is asked. This activity is designed to identify the people who are best placed to know.
3. *Score practices against checklist* This initial scoring should be "quick and dirty" to identify the process areas which are uncontroversial and those where there is uncertainty about the practices used.
4. *Resolve areas of uncertainty* This activity is designed to resolve uncertainty about the practices used. This may entail reconsultation of the people who enact the process with the aim of clarifying the ambiguity.
5. *Compute process maturity* A score is compiled based on the above. 3 points are scored for a standardised practice, 2 for normal use, 1 for discretionary use and 0 for practices which are never used. The higher the score, the fewer weaknesses there are likely to be in the process.

This process will deliver both an indication of poorly supported process areas upon which corrective attention should be concentrated, and a numeric value which provides an indication of the process maturity level. Table 2 illustrates the relationship between the score and maturity level.

Maturity level	Assessment score
Initial	Less than 55 in the basic guidelines.
Repeatable	Above 55 in the basic guidelines but less than 40 in the intermediate and advanced guidelines.
Defined	More than 85 in the basic guidelines and more than 40 in the intermediate and advanced guidelines.

Table 2 Assessment Scores and Maturity

The rationale for this classification is that a repeatable process should have implemented a good proportion of the basic practices. This is because these are generally concerned with standardisation, management and ease of use.

A defined process, by contrast, requires more systematic support provided by intermediate practices. A typical organisation with a defined process should have built up good expertise with using appropriate requirements methods (for example) and will be capable of selecting and applying new intermediate practices and even some advanced practices on particular projects.

The use of advanced practices is not sufficient to make a process a defined one. This is because, in some specialist domains, an organisation may use advanced practices while still having basic weaknesses. An example might be a company specialising in formal specification which didn't trace their requirements or standardise their requirement document structure. If the organisation generally works on small projects for familiar customers, this might not matter. If, however, the scale of their business changed they would find their process insufficiently robust.

Unfortunately, some organisations have such chronic RE problems that a process assessment would reveal nothing but weaknesses and present an unhelpfully large number of potential improvement measures. For these organisations, the REGPG recommends a "top ten" good practices which we think represent the fundamental foundations of a repeatable process. These are predominantly concerned with documenting and managing the requirements and are relatively inexpensive. Unsurprisingly, they correspond closely to long-established good practice. For example, eight of the practices are very similar to recommendations which appear in ESA PSS-05.

- Define a standard document structure
- Make the document easy to change
- Uniquely identify each requirement
- Define policies for requirements management
- Define standard templates for requirements description
- Use language simply, consistently and concisely
- Organise formal requirements reviews
- Define validation checklists
- Use checklists for requirements analysis
- Plan for conflicts and conflict resolution

4. Conclusions

The REGPG is not intended to provide revolutionary solutions to RE. However, we think that there are many RE practitioners for whom a distillation of industry knowledge and experience would be helpful Background work on REAIMS suggests that the RE process maturity of almost all organisations is still at the Initial level. Most organisations have pockets of good practice but their benefits are often diluted by weaknesses elsewhere.

For this reason, the REGPG is principally focused on the achievement of a repeatable process. This addresses what we believe are the most pressing issues of the current state-of-the practice. A speaker at a recent seminar on industrial experience in RE[†] summed up the problem in RE as being that while for most researchers RE is a problem of product complexity, for industrial practitioners the most pressing problems arise from organisational complexity. Organisational complexity is fundamentally a process issue. If the RE process cannot cope with changing requirements or large volumes of documentation, then projects will fail. We believe that the greatest leverage on organisational complexity can be gained from the basic practices which form the main-stay of progression from initial to repeatable RE processes.

Although the REGPG includes a CMM-like improvement framework, it is not intended for accreditation. Industry's enthusiasm for SPI, however, suggests a growing trend for internally-driven SPI programmes where accreditation is not the prime aim. We have tried to exploit this and have adopted an improvement framework which helps orient RE process improvement with other SPI initiatives. Within this, we have had to recognise the current paucity of accepted standards in RE and design our improvement model to accommodate a wide spectrum of views and practices. Hence, the REGPG seeks to initiate a process of consolidation in industrial RE practice rather than to prescribe a standard.

Few organisations can afford to radically change their existing RE processes. The advantage of the REGPG is that it helps incremental improvement by matching the most effective measures with the most pressing problems.

5. Acknowledgements

We would like to thank all of our partners in the REAIMS project for their support in developing the Good Practice Guide. These are GEC-Alsthom Transport, Adelard, Aerospaciale Avions, Aerospaciale Protection Systems (APSYS), Digilog, TÜVit and the University of Manchester. Particular thanks are due to Stephen Shirlaw (GEC-Alsthom), Herbert Schippers (TÜVit), Robin Bloomfield (Adelard) and Jean-Pierre Heckman (Aerospaciale).

[†] Andy Vickers at the British Computer Society Requirements Engineering Special Interest Group (RESG) meeting at the University of York, 4th February 1998.

6. References

[Boehm 94] Boehm, B., Bose, P., Horowitz, E., Lee, M-J.: "Software Requirements as Negotiated Win Conditions", Proc. 1st International Conference on Requirements Engineering (ICRE), Colorado Springs, Co, USA, 1994.

[Davis 95] Davis, A.: "Tracing: A Simple Necessity Neglected", IEEE Software, 12 (5), 1995.

[Diaz 97] Diaz, M., Sligo, J.: "How Software Process Improvement Helped Motorola", IEEE Software, 14 (5), 1997.

[El Eman 95a] El Eman, K., Madhavji, N. 1995. Measuring the Success of Requirements Engineering Processes, Proc. 2nd IEEE International Symposium on Requirements Engineering (RE95), York, UK, 1995.

[El Eman 95b] El Eman, K., Madhavji, N.: "A Field Study of Requirements Engineering Practices in Information Systems Development", Proc. 2nd IEEE International Symposium on Requirements Engineering (RE95), York, UK, 1995.

[ESPITI 96] ESPITI newsletter issue 2. 1996. Software Process Improvement on the Right Road with ESPITI - the ESPITI European Survey Results. http://www.iai.fzk.de/espiti.

[Forsgren 95] Forsgren, P., Rahkonen, T.: "Specification of Customer and User Requirements in Industrial Control System Procurement Projects", Proc. 2nd IEEE International Symposium on Requirements Engineering (RE95), York, UK, 1995.

[Hersleb 97] Herbsleb, J., Zubrow, D., Goldenson, D., Hayes, W., Paulk, M.: "Software Quality and the Capability Maturity Model", Comm. ACM, 40 (6), 1997.

[Humphrey 89] Humphrey, W.: Managing the Software Process, Addison-Wesley, 1989.

[Hutchings 95] Hutchings, A., Knox, S.: "Creating Products Customers Demand", Communications of the ACM. 38 (5), 1995.

[Johnson 93] Johnson, P. 1993. ISO 9000 Meeting the New International Standards, McGraw-Hill, 1993.

[Lubars 93] Lubars, M., Potts. C., Richter, C.: "A Review of the State of the Practice in Requirements Modelling", in Proc. IEEE International Symposium on Requirements Engineering (RE93), San Diego, Ca., USA, 1993.

[Mazza 94] Mazza, C., Fairclough, J., Melton, B., De Pablo, D., Scheffer, A., Stevens, R. Software Engineering Standards, Prentice Hall, 1994.

[Paulk 93] Paulk, M., Curtis, W., Chrissis, M., Weber, C.: Capability Maturity Model for Software, Version 1.1, CMU/SEI-93-TR-24, Software Engineering Institute, USA, 1993.

[Paulk 94] Paulk, M.: A Comparison of ISO 9001 and the Capability Maturity Model for Software, CMU/SEI-94-TR-12, Software Engineering Institute, USA, 1994.

[Potts 94] Potts, C., Takahashi, K., Anton, A.: "Inquiry-Based Scenario Analysis of System Requirements", IEEE Software, 11 (2), 1994.

[Potts 95] Potts, C.: "Invented Requirements and Imagined Customers: Requirements Engineering for Off-the-Shelf Software", Proc. 2nd IEEE International Symposium on Requirements Engineering (RE95), York, UK, 1995.

[Ramesh 95] Ramesh, B., Powers, T., Stubbs, C.: "Implementing Requirements Traceability: A Case Study", Proc. 2nd IEEE International Symposium on Requirements Engineering (RE95), York, UK, 1995.

[Rodden 94] Rodden, T. King, V., Hughes, J., Sommerville, I.: "Process Modelling and Development in Practice", Proc. EWSPT'94, Villard de Lans, France, 1994.

[Rout 95] Rout, T.: "SPICE: A Framework for Software Process Assessment", Software Process-Improvement and Practice, Pilot issue, 57-66, 1995.

[Sommerville 97] Sommerville, I., Sawyer, P.: Requirements Engineering - A Good Practice Guide, John Wiley, 1997.

[Sommerville 98] Sommerville, I., Sawyer, P., Viller, S.: "Viewpoints for Requirements Elicitation: a Practical Approach",.To appear in Proc. Third IEEE International Conference on Requirements Engineering (ICRE 98), April 1998.