David S. Hinley
(Ch 6: with
Remko vd Pols
Machteld Meijer)

Version 4
20 july 2000

PinkRoccade

Barracuda Content Design

A methodology for the provision of Application Services.

Barracuda Content Design

A methodology for the provision of Application Services

1.0 Management summary

Most businesses are now to some extent critically dependent on their application systems to support their business processes and provide pertinent management information in order to facilitate executive decisions.

Organisations continue to make significant investments in application systems both in terms of their initial development and their continued maintenance. Indeed software maintenance is the most important contributory factor to system life-cycle costs. Maintenance and support of an application can amount to between 50% (IBM Research Institute) - 67% (Institute of DP Management) of the overall life-cycle costs. This places a great burden on any organisation trying to find sufficient funds to budget for the initial development of new or replacement systems that may be necessary to exploit new business opportunities.

Continual maintenance and evolution of existing applications is a wise investment and is necessary to ensure that existing systems continue to meet the needs of the business and function dependably. For many years maintenance was perceived as a problem area; however, competent management can now ensure that the maintenance and evolutive processes are efficient, effective and provide value for money.

Executives focusing on their business directorates are skilled in strategic thinking and providing the necessary leadership to accomplish the business objectives. ICT management have traditionally had either an operational or project management background, with perhaps a limited appreciation of aspects of service management. The current emphasis on service issues within ICT requires both strategic and organisational decisions on how to sustain large and complex application systems at an acceptable level of quality for many years to come.

Executives are increasingly aware of the limitations or problems that applications pose to the achievement of their organisational goals. Applications Management is the discipline which aims to provide application servi ces in order to minimise future risk, ensure that applications function dependably and continue to meet the business needs.

Contents Page

2.0 Introduction: The need for a framework

Maintaining software and evolving application systems within the overall context of providing application services is considered to be a growing market, that is becoming increasingly competitive. Companies have experienced the effects of the major problems with software maintenance for a number of years and are keen to:

- reduce the customer risk,
- obtain value for money,
- have a higher degree of freedom in which to exploit business opportunities,
- experience demonstrable improvements in the quality of their applications and in the levels of service provided.

As more ICT service providers recognise this market niche, it is imperative that one can demonstrate that a quality service can be provided at realistic market rates. To communicate this message to all parties, a framework or model is beneficial.

For example, in offering to manage an organisation's application portfolio, a clear value proposition should be put forward. This can be facilitated by reference to a framework which is able to outline or demonstrate how customer and suppliers can work together in partnership and achieve mutual benefits.

Most organisations have become lean and specialised in terms of their competencies or streamlined in terms of their business processes and need to focus on their business direction and competitive edge. The dynamics of business place extreme pressure on application systems that in effect model the business processes, and support management by providing pertinent information on which to base decisions.

An applications service provider might offer propositions regarding the scalability of the applications, rationalisation in terms of achieving more with less, and in providing greater business benefits. Such propositions may be attractive to an organisation, e.g. enabling their Company Executives to focus more on strategic initiatives.

2.1 The importance of applications management

Application management is a serious concern for most organisations for:

- Financial reasons – maintenance and support of an application may account for two-thirds of the overall cost of each application during its service life.
- The adverse effect the existing applications may have on an organisation's ability to exploit new business opportunities and emerging markets.

- The need to address the quality of applications as an integral part of an ICT service that the business is increasingly dependent upon.

The cost as well as the importance of software maintenance is now widely recognised, particularly in the light of global Y2K programmes. On a company basis, it is now recognised that application systems can have a longer life expectancy than first realised and this presents problems in how to manage and exploit ‘legacy’ systems.

Indeed although software maintenance has been viewed traditionally in operational and perhaps tactical management terms, the longer term consequences are now being realised in terms of issues affecting the strategic intent of many organisations. For example, in the U.K. the merger of demutualised financial service organisations has been postponed on more than one occasion primarily because of difficulties in bringing together their application portfolios.

The overall costs to companies for maintaining their existing software is difficult to quantify. There are a number of reasons for this; a simple fact is that although maintenance has been defined and the types of maintenance activity are readily described, few organisations adopt the technical definitions. Companies tend to distinguish between systems development and software maintenance on an arbitrary basis often linked to their budgetary processes and reporting. For example, a study conducted by Olivetti Information Services recognised that from a sample of 35 companies, 21 distinguished between maintenance and Development projects mainly on the basis of a size estimate for the foreseen activities. They also found that a significant number of companies (9) used the term maintenance, to indicate only the addition of end-user functionality to existing systems. Corrections and adaptive work being conducted simply as overhead within the ICT budget!

Few companies have a formal way to separate maintenance and development activities based on the intrinsic nature of the actions required to perform the task. This has been viewed by some as beneficial in that the true cost of maintenance and the effort expended can often be hidden. Indeed managers’ have been wise to be cautious with annual maintenance efforts expending somewhere between 25-75% of their available IS manpower.

An overall figure, estimated some 10 years ago, was that the maintenance spend in Europe was approx. £30 billion p.a. (Source: INPUT 1990). Indeed large organisations, e.g. the U.S. Federal Government were reported (Quality Assurance Institute) to be spending $3.75 billion p.a. on maintenance. This is possibly an underestimate as the figure represents only 23% of the Federal Government IT budget.

Although the sensitivity towards spend in relation to IT budgets remains, Executives are increasingly aware that their businesses are critically dependent on application systems. Consequently, organisations are willing to make a significant investment in application systems both in terms of their initial development and their continued support and maintenance, to ensure that the system meets the evolving business needs and continues to function dependably.

An important factor for Applications Management is whether the investment made can result quickly in the achievement of realisable benefits, as far as the organisation is concerned.

To optimise the benefits profile of existing systems, it is necessary to accomplish changes in a timely and cost-effective manner. Systems reliability should at least be sustained or moreover improved, and in addition the maintainability of a system should not degrade, otherwise future changes will be progressively more expensive to perform.

The need to react quickly to new business opportunities means that the lead-time to provide the necessary systems support for new products and service offerings is on the critical path. Despite the advances in Development methods, including: Rapid Application Development (RAD), Joint Systems Development (JSD), there is often little time, suitable resources, and perhaps a lack of capital, to develop or procure a new system from scratch. Evolution of an existing system could be a viable alternative with additional benefits in that the non-functional requirements, including systems behaviour, performance etc. can be communicated more readily between the customer and the developer.

The customer keen on exploiting business opportunities, perhaps even presented by new technology, needs to be aware of the options, risks, and service quality arguments. Above all, application service providers' need to be able to raise awareness of the possibilities and be able to tackle the business concerns described above.

However, from the perspective of the application service provider, it is also necessary to be aware of ones own business opportunities and undertake sufficient assessments to assure that the supplier risks are manageable. This may involve an assessment of the applications portfolio and gaining a common understanding and commitment to achieve high quality levels of service.

To provide a quality service, application managers need to have good visibility of the business direction and key objectives. This can best be achieved through mutual trust and understanding and through developing a strong relationship with the customer.

Having described the primary reasons why application management is important it is worthwhile reflecting on some of the quality issues that are important from the perspectives of both the customer and service provider.

Maintainers have generally been concerned with the quality of the product and have payed less attention to the overall application service. It has become increasingly evident that the process by which the software is developed and maintained has a major impact on the product. The basic argument is that the processes undertaken govern the product attributes.

This 'focus on the process' is highlighted from the management perspective, when one considers the top six problems facing maintenance managers:

- user demands for enhancements to application systems

- competing demands on maintainers time

- the quality of application system documentation
- unrealistic user expectations
- quality of the original development
- meeting scheduled commitments

Consequently, there has been widespread interest in process management. Indeed, an analytical approach to processes, including for example, maintenance and customer relationship management, should enable them to be understood, measured and improved. The assumption is that this will lead to an increase in the quality of both the product and service offering, and therefore encompasses the client perspective.

2.2 The definition of applications management

Applications management has evolved from two directions, namely: software maintenance management, and service management. By combining the two managerial disciplines and through focussing on the organisation's application portfolio as a whole, the Business Managers may focus on organisational goals and strategic direction, whilst application management consider how best to maximise service quality and cost efficiency with respect to the evolving business needs.

In defining applications management, the underlying process of software maintenance and evolution, and service management need to be considered.

Software maintenance is a commonly 'understood' term for which there is no single definition. The standard definition often cited is that of the IEEE, (Std. 610.12-1990), given below:

> *The modification of a software product after delivery, to correct faults, to improve performance or other attributes or to adapt the product to a changed environment.*

Software maintenance is the set of activities that result in a change to the originally accepted (baseline). These changes modify and often enhance the baseline system and are carried out in order to keep the system functioning in an evolving user and operational environment.

The simple definitions of maintenance do little to emphasise the problems encountered by maintenance managers. Indeed for many years the difficulties were perceived as technical in nature requiring new methods and support technology. It is now realised that the problems are predominantly organisational and managerial and that the process of maintenance has greater significance to the business than perhaps was realised initially.

A more useful definition of software maintenance might recognise that companies are increasingly dependent upon ICT in support of their core business functions, and that the dynamics of business and the cost implications of constant change inevitably lead to a significant transformation from technical detail to a managerial perspective.

Software maintenance could be defined as:

> *A set of managerial and technical activities that ensure that application systems continue to meet both functional and behavioural requirements in order to support business goals in a cost-effective, and reliable manner.*

Empirical surveys have indicated that the majority of maintenance is not corrective, i.e. correcting faults in the software, but is in fact concerned with systems evolution. The important requirement for the client is that changes are accomplished quickly and cost effectively. The reliability of the application should at worst not be degraded by the changes. Additionally, the maintainability of the system should not degrade, otherwise future changes will progressively be more expensive to carry out. This phenomenon was recognised by Lehman, and expressed in terms of his laws of evolution, including:

> The law of continuing change – *a program which is used in a real world environment necessarily must change or become progressively less useful in that environment.*
>
> The law of increasing complexity – *as an evolving program changes, its structure tends to become more complex. Extra resources must be devoted to preserving the semantics and simplifying the structure.*

There are many technical and managerial problems in striving to accomplish the objective of modifying and enhancing applications quickly, reliably and cheaply. The service management perspective aims to ensure that customer satisfaction with the overall service is maintained and that change requests are assessed from the point of view of business impact as well as in terms of a technical appraisal on which to base sound estimates.

The service perspective enables the maintenance programme to be optimised, so that critical applications continue to be aligned with business needs. The service perspective also provides both visibility and accountability to the business in terms of a service offering, thus facilitating in the management of application life-cycle costs.

Application managers aim to ensure that the client's application portfolio satisfies business needs and that the applications continue to function dependably and meet user expectations. Applications managers also ensure that the maintenance and evolutionary activities are carried out in such a manner to minimise risk to the business.

Applications management is defined as:

> *The contracted responsibility for the management and execution of all activities related to the maintenance and evolution of existing applications, within well defined service levels.*

Applications management involves far more than software maintenance management in a number of ways, including being more proactive, outward looking or business facing, i.e. recognising the service requirements and expectations of the client organisation. Applications Managers' need also to be cognisant of the tactical management issues relating to software maintenance and systems evolution.

2.3 Common problems with maintenance / applications management

The first reported investigation of the major problems associated with maintenance management was conducted by Lientz and Swanson in 1977. Among the six problems that they identified as the most severe, three were concerned with users of application systems, two involved managerial constraints with time / resource availability, and one problem was technical in nature, i.e. concerned with the quality of software documentation.

Other surveys, for example, that of Ned Chapin in 1984 surveyed 260 maintenance managers and discovered that 48% of the major problems were related to the characteristics of the software itself, such as bad documentation, complex and not well understood code, old code etc. User relations accounted for only 5% of all the problems.

The Software Maintenance Association (SMA) conducted its own survey in 1987, and based on responses from 81 organisations found that the most frequently mentioned (29%) of all challenges could be placed in the category of 'environmental factors'; this included the lack of resources. 'System characteristics' and 'Management' were the next two largest categories with 21% and 18% of all responses, respectively.

A follow-up survey by the SMA in 1990 revealed that the four most populated categories of problems were 'Management' (28%), 'System characteristics' (21%), 'Personnel issues' (19%), and 'Environmental factors' (19%).

The results of these studies do indicate that the perception of maintenance problems has changed over time. In the late '70s and early '80s maintainers felt that the major cause of software maintenance problems was users. Blame later was attributed to the poor state of software, then towards the environment, and finally towards management.

More recent 'scientific' studies using Delphi techniques found that that major problems are:-

Maintenance management

- changing priorities
- inadequate testing methods
- performance measurement difficulties
- large backlog of change requests
- lack of maintenance methodology, standards, procedures and tools
- lack of managerial understanding and support

Organisational environment

- adopting to rapidly changing business environments

- value contribution measurement difficulties
- strategic plans
- lack of support for reengineering

Personnel factors

- low staff morale
- lack of maintenance personnel
- lack of proper training
- understanding and responding to business needs
- high turnover causing a loss of expertise

System characteristics

- system documentation being incomplete or non-existent
- complex and unstructured code
- integration of systems or subsystems
- legacy systems.

The top six problems uncovered are outlined below:

Changing priorities – This is perhaps more complex than at first sight. The general perception is that due to the length of the change cycle, new requests arise which are more critical or another user has a problem requiring investigation, resulting in a great deal of time being wasted in stopping and starting maintenance tasks. Typically long term activities, e.g. reengineering suffer as the necessary resources are continually being pulled away.

A related issue is that of planning and particularly the difficulties encountered when business plans are not well defined or adhered to, and there is insufficient attention paid to communicating the benefits related to the work in progress.

Inadequate testing methods – The general problem is the lack of understanding and use of testing methods, the lack of time, lack of regression testing, and the lack of rigorous operational acceptance standards.

Performance measurement difficulties – This problem is perhaps related to the fact that there is no common or standard definition of maintenance resulting in difficulties in measuring individual or group performance. The lack of task definition impedes the determination of appropriate measures.

System documentation – The lack of good quality documentation decreases maintenance productivity and increases the learning curve dramatically for new personnel.

Adapting to a rapidly changing business environment – The dynamic nature of business and the impact new technology can have on business processes means that existing applications become less useful rapidly with time. Evolution and adaptation can become increasingly difficult in the case of old systems, because of poor quality code, high complexity, or old technology.

Large backlog of change requests – Users can be dissatisfied and impatient if their requests are not dealt with in a timely and efficient manner. The classic approach to dealing with this problem is through an established system of priorities. Consultancy organisations, in the U.K. were reporting low success in controlling this problem, even in recent years. Other useful strategies include: incrementally improving old systems, improving managerial and service support, requirements management, and charging policy.

A key finding on the common problems is that management problems correlate with personnel problems and are often causing them. System characteristics may also induce personnel problems. It is also noted that the organisational environment has traditionally engendered management problems.

There are many factors which influence the high cost of software maintenance. For example, if change requests are serviced in the order in which they are made then extra costs may be involved. This is because some changes may overlap or affect others, resulting in duplication of work and increased complexity. Costs can be reduced by scheduling change requests and through the creation of work packages.

Many of the surveys undertaken in the past to identify key problems have focused on software maintenance. However, many of these problems are common issues for applications management.

To summarise, Application Managers need to be aware that:

- Organisations can have a number of ‘legacy systems’ which need to be dealt with. These are often large applications which have been and continue to need to be supported for a number of years.
- Over the last 45 years, there has been six orders of magnitude in scale of application size; the development and maintenance processes are barely adequate to implement and sustain applications in a predictable and repeatable manner to meet customer expectations.
- For many applications, the root problem is complexity; sometimes this arises through perfective maintenance in order to gain additional functionality, and the scale of the application means that it is not possible for one person alone to understand the complete software system.

- Verification and validation remain major problems; it is often difficult to find errors simply by testing; yet with an intensively used application, with many concurrent users, the faults may appear relatively frequently.

User expectations are generally high. The important requirement from the client's perspective is that changes are accomplished quickly and cost-effectively. The reliability of the software should be at least sustained or moreover improved and should certainly not be degraded. Efforts need also to be taken in order that the maintainability of the application system does not reduce, otherwise future changes will be progressively more expensive to carry out.

Empirical surveys of maintenance work indicate that the cost of maintenance as a fraction of total systems life-cycle costs is high, although the cost does not appear to be rising partly because of effective management control mechanisms and partly due to the general improvement in maintenance technology. Technical processes may also be optimised to reduce both the costs and the number of introduced defects, through change control and impact analysis.

Impact analysis is a key activity in the maintenance process and can provide both technical and management information early in the process life-cycle. The emphasis being to provide accurate information about the impacts of changes at an early stage and to focus on the management issues.

3.0 Historical perspective

This section provides a chronological sequencing of key contributions that are pertinent towards our current thinking on applications management. It is not intended to be a comprehensive 'Annals of Software Engineering' or 'Introduction to Computer Science'.

The early and, at that time, large development projects of the '60s resulted in the recognition of Software Engineering as a discipline and the necessity to organise development. Key concepts introduced in the '60s and relevant today include: modularity e.g. modular programming, and 'divide and conquer' i.e. forms of partitioning to deal with complexity. Major players included: Brooks, Dijkstra, Jackson, Knuth, Parnas.

The philosophies applied are embedded in Legacy systems being maintained today, e.g. many business systems have Cobol programs of an enterprise origin from this period, and indeed 'high tech.' command & control systems e.g. for NATO Naval forces are being maintained in Algol '68.

In the 1970s, there was a shift of emphasis from programming towards analysis and design. Key concepts introduced in the '70s included: abstraction and data independence. Much effort was put into Data Analysis, and major players, e.g. Chen, Codd, Constantine, DeMarco, Gane, Martin, Orr, Warnier, Weinberg, and Yourdon institutionalised Structured Analysis & Design methods.

The analysis techniques, e.g. abstract modelling were applied to rationalise and build upon experiences gained on large transaction orientated systems. Life-cycle management models, e.g. the Water-fall model for development projects became fashionable.

It was also in the 70's that the first maintenance models were published. These models tried to distinguish maintenance as a different process from development. Key contributions included:

- The **Liu model** (1976) – this model was applicable to perfective maintenance and recognised that gaining an **understanding of the existing system** was an **essential** prerequisite prior to designing new logic for the additional functionality.

- The **Boehm model** (1976) – Barry Boehm is more noted for his work on software economics, his cost model COCOMO, and the Spiral model for software development that explicitly incorporates risk assessment into the development process. However, his contribution towards maintenance modelling was the recognition of the importance to **revalidate systems following modification**.

- The **Sharpley** model (1977) – this model attempted to recognise that maintenance was not solely concerned with enhancements, but that support was also required to deal with problems. **Problem verification and diagnosis** were important activities, and also **configuration management in terms of baseline verification** before and after change.

- **Yau and Collofello** model (1978) – at this point in time some of the practicalities of maintenance were being understood more fully by management. This model recognised that investigation and understanding was necessary prior to the generation of a maintenance proposal, and that **ripple effect analysis** based on the proposal was necessary. The model also recognised that **regression testing** was necessary to ensure that the modifications have not adversely affected other parts of the system.

It was also in the 70's when Lientz and Swanson undertook their survey of software maintenance problems, previously discussed, resulting in their book *Software Maintenance Management* published in 1980.

The late '70s saw the adoption of PROMPT – a project management method by the CCTA which was later to be transformed into PRINCE (Projects in controlled environments) in 1989.

The 1980's saw the establishment of rigour and conceptual integrity; a more disciplined approach towards systems development was advocated and structured methods, e.g. SSADM (1982) were introduced.

Two maintenance models of significance were put forward in the '80s, namely:

- The **Parikh model** (1982) – this model recognised that it was necessary **to identify the business objectives** of what was to be achieved through change, as these were not always explicit in the request for change.

- The **Patkow model** (1983) – recognised that **specification of** the maintenance **requirements**, similar to requirements definition in development, is a key stage, facilitating in the **diagnosis and change localisation**, prior to the modification design.

The period saw the introduction of CASE tools and the widespread adoption of PCs, resulting in significant changes to the traditional DP centre and MIS department. End-user computing became fashionable, the need to reduce development lead-times and to provide much needed support to the PC user and deal with non-routine requirements resulted in the widespread establishment of Information Centres incorporating Help desks, prompting the need to rethink: service support and delivery, as opposed to Operations.

It was in this period that the CCTA's IT Infrastructure Library (ITIL) came into being, i.e. a comprehensive reference to 'best-practice' service management.

The late 80's saw an interest in quality management and the weaknesses of IT project management, and life-cycle models became apparent. It was evident that no single development or maintenance model was appropriate for all situations. The models were often used by managers for communication exercises, and for initial planning purposes, but they did not cope well with change, and they did not represent the day-to-day activities and necessary interactions. Project Management tended to be based on milestone events, and the quality aspects were judged on the deliverable.

A software process movement emerged on the mid-1980s when the shortcomings in managing development and maintenance processes were recognised as prime inhibitors of growth in software productivity and quality. Lessons had been learned from the often failed attempts to introduce CASE technologies, and the failure to realise benefits. It was realised that it was necessary to focus on the processes and then to select methods and tools that support the processes.

Watts Humphrey recognised that if the maturity of an organisation's software process influences its ability to meet cost, quality, and schedule targets, then it is wise to understand what distinguishes mature from immature organisations. He put forward a model of process maturity, namely the Capability Maturity Model, (CMM), in 1989. The staged structure of the CMM was based on product quality principles that have evolved over 60 years, and the maturity framework into which the principles underlying the CMM have been adapted was first elaborated by Crosby (1979) in his book *Quality is Free*. The CMM provides a framework for identifying five levels of maturity that lay successive foundations for continuous process improvement; these were first described in *Managing the Software Process* (1989).

The 80s also saw the establishment of The Centre for Software Maintenance (CSM), and now the Research Institute for Software Evolution (RISE) at the University of Durham, U.K. This was the first centre of excellence to research maintenance problems world-wide and to organise International Conferences on Software Maintenance.

It was through such International forums that in 1988, Mel Colter from the University of Colorado, presented his vision of software maintenance becoming integrated with software development with some form of common management structure to support a product during its lifetime. The concept was to move from 'survival' to 'product management', by means of the following transformations:

→ product responsibility;

→ product improvement;

→ product direction;

→ product management.

This vision led to the coining of 'Applications Management' during the 90s.

A number of process models and frameworks have been developed in the last ten years relevant to software maintenance and applications management and these are discussed in the Application framework library, [4.0].

4.0 Application framework library

This section provides a brief description of specific research project results, models and frameworks that are available within the Public domain. Each contribution to advancing the knowledge of software maintenance and to applications management is highlighted in terms of the lessons learnt.

4.1 SWM process model (CSM, Durham)

The first detailed generic Software Maintenance Management (SWM) process model was published (Hinley & Bennett, Centre for Software Maintenance, 1992) as part of a joint industry and academic research project, funded by the Information Engineering Directorate of the DTI, UK.

The 'Models and Metrics' project was concerned with the question of whether industrial software maintenance management could be improved through the utilisation of a comprehensive process model.

The reference model attempted to graphically represent 'best' software maintenance practice. It was found to provide Managers' with further insights and support of the process, and facilitated in improving real maintenance in line with organisational objectives.

As part of the research, the scope of maintenance management was refined, see **Figure 1** The scope depicts seven high level process elements described below:-

- **the development of a maintenance strategy** following the evaluation of the company objectives, organisational goals, mission statements, policies and directives. The existing applications and current development and support processes are assessed in terms of their value as company assets. The desired strategic direction is determined and the software application requirements are denoted in portfolio terms, so that the future maintenance strategy is both justified and aligned with the business direction.

- **prototyping both process and product technology**. It is well recognised that the dynamic nature of business and the necessary changes required to applications for them to continue to meet the needs of the business, necessitates that not only do applications evolve, but that the maintenance and support processes are improved. The evolution of applications and the software process is a primary consideration for applications management. Decisions need to be taken about, for example, how existing artefacts may be rejuvenated, software quality improved and good software re-used. These are difficult decisions and it is suggested that a prototyping approach may be applicable to: discover requirements, pilot alternative approaches, prototype new deliverables, and generally facilitate the decision process by raising management's awareness of the options and elevating confidence in the feasibility.

- **project planning and control**. Such activities are at the heart of the management process, but they are often inadequately conducted in the maintenance situation. For example, maintenance planning is often undertaken to a poor standard, and work packages are neither controlled or well-defined. This is one of the management domains in which maintenance teams may work closely with application management in order that the technical decisions are well founded and can be related to the broader business perspective. The confidence of the customer can be enhanced if there is some confidence that milestones and targets are achievable, and that they are involved in terms of deliverable acceptance and verification in accordance with defined criteria.

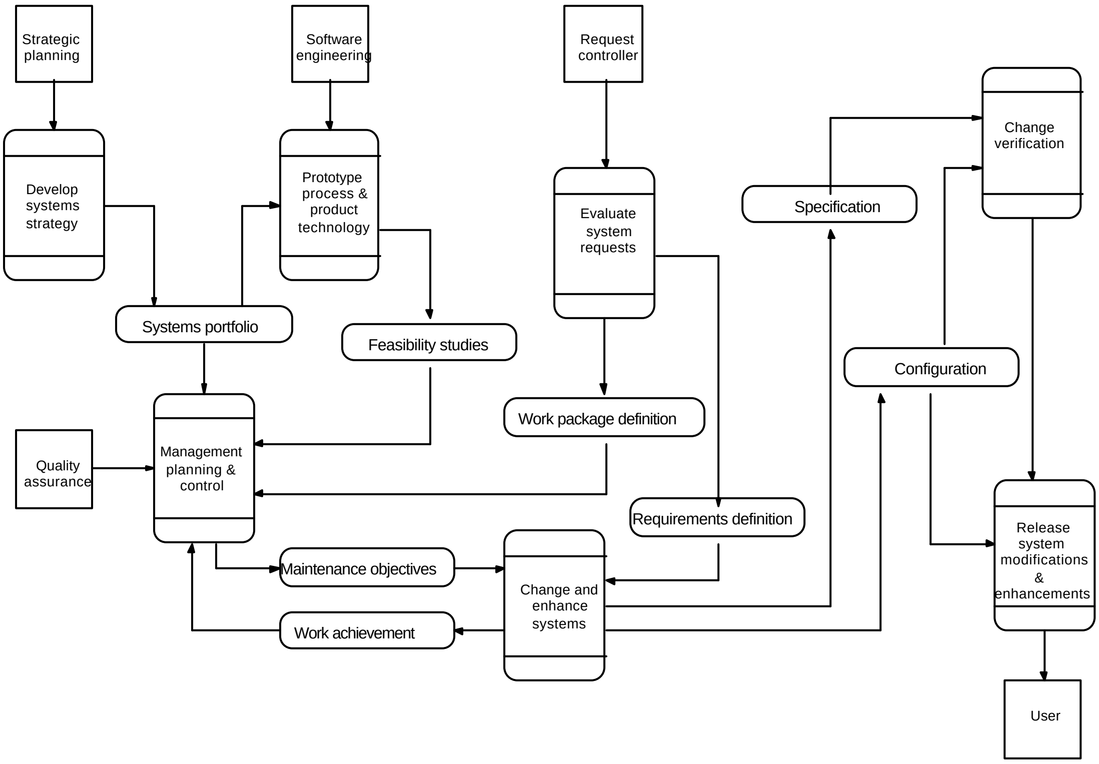
- **request control**. To improve the overall management, there is a need to establish a well-defined interface between the customer and the applications support team. A number of activities associated with this 'change management' domain have been identified, including: the collection of background information pertinent to requests, maintaining a request register, problem definition, risk assessment, high level impact analysis, assigning suitable task priorities, formulating work packages. The role of change manager has overall responsibility to ensure that the quality objectives associated with customer service are achieved.

- **software change process.** Clearly, this is central to the technical activities associated with the maintenance teams. Management control is emphasised, e.g. planning, monitoring and reviewing. Better practice involves both explicit quality assurance and configuration management, with actual changes to source code occurring within the latter part of the process, and only after appropriate reviews, e.g. of the change definition and design. The effects of any software change should be analysed and further change may be instructed in order to stabilise the application.

- **product evaluation.** This provides an independent control mechanism for the change process and provides the necessary management information for release planning. The activities should not simply be concerned with software testing, but have wider scope encompassing product assurance functions that are independent of the technical verification activities undertaken as part of the change process. Better practices embrace all aspects of the application, including: documentation control, standards, configuration management, audit etc.

- **release control.** This involves a number of planning, review and project close-down activities which are beneficial to process improvement. It is necessary to establish a well- defined user relationship with quality objectives targeted at customer acceptance. Evaluation activities and configuration management within the change process provide the necessary information for the contents of a release to be determined and built. Good communications are necessary with the customer to ensure that the necessary planning for delivery and acceptance is performed, so that the application can be released to the customer, accepted and supported in an efficient manner.

Figure 1 – Context model for the management of software maintenance



The framework challenged the current activities of maintenance managers at the time and made them aware of better practice recommendations. For example, the model recognised the need to consider software application systems in 'asset' terms and to consider the maintenance of the product portfolio, a recommendation made some 10 years previously by Lientz and Swanson, but rarely enacted upon by maintenance managers.

An interesting spin-off from the modelling work was in the development of goal-orientated approaches to process improvement. The approach facilitated in the identification of management metrics for both software maintenance and in measuring the efficacy of process change. The outcome was a positive effect on managerial ability to meet both product quality and service expectations.

This latter work moved forward from simply managing technical processes to, with due regard for economics, to ensure that the useful life of a software product is maximised. At this time it was noted, that software maintenance is not only governed by the type of request, the urgency of the requirement, but also by the justification of the effort required and the benefits that can be achieved. Moreover, software maintenance is concerned with quality improvement in that it provides an opportunity to correct obvious faults uncovered during use, and enables the existing applications to be enhanced or perfected to meet the customer's operational needs.

One of the conclusions drawn from the use of the framework, pertinent to applications management is the recognition of a key business in ensuring that software assets are effectively managed, and that the quality of the applications is at least sustained or improved as a result of the maintenance process.

In considering existing systems as an asset and not just as a ‘legacy’, it is necessary to identify and assess the risks of maintenance and assure that the value of the application is maximised from the customer perspective. The need to identify and manage risks as an essential element of better practice for software maintenance management was recognised by Hinley & Bennett. In their paper, *Reducing the Risks in Software Improvement through Process-orientated Management*, published in 1993 by the IEEE Computer Society Press, they noted the failure to take into account the very different choices faced by applications managers as compared to development project managers, and the need to widen the risk assessment to concern three broad issues, namely:

- Procurement - issues arise at a number of levels, e.g. procurement of packaged software to replace existing bespoke applications, the use of contractor services and supplier policies. It is often necessary to address cost reduction, improved quality, including reliability.

- Existing systems – few maintenance managers consider the application portfolio in asset terms and there a few strategies adopted by which software products and processes are considered for re-use. Of particular importance to application managers are the issues of: increased variety of system use, e.g. multi-function, and shorter lead-times for incorporating enhancements, and manpower justification.

- Organisation – it has been noted that maintenance work is often only weakly related to business goals and critical success factors. Quality goals are rarely stated in measurable terms and any improvements are difficult to demonstrate. There is an urgent need to consider quality measures in product and service terms and provide value for money.

Preliminary studies at the time indicated that where an organisation had established a risk management procedure it was usually geared towards contract review for new development work or initial tender, and was rarely geared towards a consideration of the complete life-cycle.

The management framework was also applied in the development of management control mechanisms for software maintenance, namely: organisational, request, change, and release. These control mechanisms were analysed in terms of generic processes and a number of process elements and logical data-stores relevant to risk assessment were identified.

For example, the organisational model included:

- Organising resources for maintenance;

- assessing the skills and competencies of available resources;

- establishing maintenance projects, defining roles and a suitable quality management framework;

- managing the software asset strategy;
- creating and maintaining an environment for software maintenance, including configuration management;
- reviewing maintenance plans, the environment, and software strategy.

It was established that risk assessment was an integral part of the scheduling activity and the review of maintenance plans. In practice, maintenance generally lacks an explicit risk management process, although impact analysis is often undertaken to scope the size of the change and consider alternative solutions.

It was concluded that risk management needed to be considered at a number of levels for the identification, analysis, avoidance, and economic control of those risks which threaten the software assets or the operational capability of the customer's business. At the maintenance level, the process should be an integral part of maintenance work planning and thereafter monitored and updated as appropriate.

Field trials with application managers indicated that because they were often unfamiliar with the application portfolio, it was beneficial to involve team members at the outset, i.e. at the definition stage of work packages. Even a high level consideration of software size, structure and technology was valuable in obtaining a general risk rating. A work sheet for risk factor assessment were proposed for each stage of the maintenance / development stage; for example, the following aspects were considered at the analysis stage:

- the number of tasks in the defined work package;
- the number of discrete user groups of the application;
- the level of decision making;
- the quality of existing requirements documentation;
- availability of customer representatives;
- work package priority;
- the analysts domain knowledge and experience of the software functionality.

Independent assessment of the affects of risk management as an integral part of applications management showed that there had been a positive improvement in the ability of the maintenance team to meet twin management goals of at least sustaining the existing high level of software quality and carrying out enhancements to cost budget. Indeed risk management was found to have had a positive effect on the following critical success factors:

- determination of customer requirements and expectations;
- establishment of the change requirements and deliverables;

- recognition of, and adherence to appropriate standards and procedures for the level of assurance necessary;
- ensuring the appropriateness of management controls;
- recognising the need to update or re-appraise estimates;
- establishing resource requirements;
- recognising the need for new tools and facilities;
- valuing domain knowledge, acquisition and sharing.

The maintenance management model, addresses some of the concerns of applications management with respect to management control and the creation of an applications portfolio. This has an affect on the overall organisational control and a direct bearing on the applications management strategy.

At the tactical level, the distinction between projects and work packages is made, and the issues of promoting the maintenance service, prioritising requests, planning change, and release management are considered. Operational management is demonstrated to have effective control mechanisms that ensure the continued provision of an appropriate application management service.

However, whilst the model considers quality assurance, it does not adequately describe the service management aspects, e.g. contract negotiation, and business benefit realisation.

4.2 The Capability Maturity Model (CMM) applied to Maintenance (SEI-Carnegie Mellon)

The Software Engineering Institute (SEI) based at Carnegie Mellon University has been instrumental in the development of a scale by which process maturity may be measured, and thus also a metric for process improvement. EU-funded initiatives such as the Esprit Project: Bootstrap, have resulted in alternative assessment criteria, but the five-levels of process maturity have been generally adopted.

However, the ideal level of maturity that an organisation should seek to maintain is dependent on the organisations goals and business needs. For example, even within a large Company, different software divisions may need to attain different levels of process maturity in order to remain competitive within the market.

Maturity models facilitate in the assessment of the current process maturity, and may also be instrumental in any improvement programme in order to achieve a higher level of maturity. However, the early models provided little guidance on implementing better practices, and were not geared towards organisational needs (business drivers) and priorities. These criticisms are being addressed to some extent by efforts to standardise process maturity and improvements, e.g. the SPICE initiative.

It should be noted that the maturity models mentioned, namely: SEI-Capability Maturity Model (CMM), Bootstrap, and SPICE focus primarily on the process of systems development and do not necessarily give an accurate assessment of software maintenance activities or the provision of services.

However, the structure of a maturity level consisting of key process areas and for which key or 'best' practices have been identified, is a useful framework which can be applied to applications management.

The Capability Maturity Model, recognises numerous goals and key practices associated with software development and acquisition, but it is not explicit to applications management and omits a number of key processes and integration features necessary to describe applications management 'best' practice. For example, the CMM does not address the marketing issue, cost reduction, and tactical maintenance decisions, necessary for dealing with legacy systems.

Furthermore, the CMM does not address the aspects of service planning and service provision which are an essential part of applications management.

An applications management framework needs to be cognisant of the development process 'best' practice. It is also necessary to consider the maturity of existing and proposed maintenance processes within any client organisation, to determine a transition plan towards achieving process improvement.

4.3 Maintenance Assistant Capability for Software (MACS)

MACS was an EU-funded Esprit II Project (No. 2570) which started in 1989 and was completed in 1993. Cap Gemini Innovation were the prime contractor, with organisations from the UK, France, and Italy participating. Academic input was provided from the Universities of Bremen, and Limburg. The total effort expended was 71 man-years with a budget of 11m. ECU.

It was recognised that applications must adapt to change in the rapidly evolving market. For business evolution, two options are open: build new applications or maintain existing ones. MACS was founded on two premises: (i) that in building any new applications, care should be taken to prepare for future maintenance, and (ii) whilst maintaining existing applications, techniques such as reverse engineering and knowledge representation are necessary to facilitate software component re-use and software development in general.

The MACS project regarded reverse engineering as a central activity in software maintenance, in terms of design recovery and re-documentation to facilitate system comprehension. The reverse engineering of an existing application enabled it to be restructured, for components of the application to be reused, and for existing applications to be re-engineered to produce new applications.

During the MACS project, it was found that in reverse engineering, gaining a suitable abstract model for comprehension and re-documentation was a significant problem. The existing requirements documentation and functional specifications produced during development were a contributory factor towards the maintenance problem.

It was also noted that assistance is required to produce abstract models, primarily because of the lack of information on the design decisions taken during the development process. Assistance is both helpful and feasible if this knowledge can be captured.

MACS also discovered that re-use and re-engineering based on design recovery and re-documentation was a laborious process and at that time (c.1991) difficult to automate.

The project concentrated on three areas of assistance linked to change management, reverse engineering, and impact analysis. The project deliverables included a knowledge base containing MACS data that contained answers to the 'what', 'why' and 'how' questions posed in maintenance.

Reverse engineering leading to design recovery and some re-documentation can be beneficial in reasoning about the effects of changes to the application and in the formulation of maintenance strategy and detailed change proposals. Finally, the MACS approach is beneficial in improving the maintainability of applications by facilitating traceability and evaluation of the completeness of any abstract logical models.

The MACS method is centred on maintenance level activities and the provision of suitable tools to facilitate maintenance, including: change management, problem reasoning, reverse engineering, and document browsers. The MACS method does not really address any of the strategic objectives of applications management, (it may help improve productivity through the use of maintenance tools, and hence have an affect on cost).

MACS may support some tactical management decisions, e.g. by means of effective project control. At the operational level, the MACS method deals effectively with the technical issues associated with maintenance and also supports reverse engineering.

4.4 European Platform for Software Maintenance (EPSOM)

EPSOM was a sub-project of the Eureka Software Factory, and was primarily concerned with investigating the notion of an integrated software maintenance environment. The project started in 1991 and lasted four years; it was led by Matra Espace, and involved Cap Gemini, Hoskyns, and the CSM at the University of Durham.

EPSOM studied real world maintenance teams to try and understand their problems and derived a number of factors which affected the cost-effectiveness of maintenance. These could be broadly categorised as: People, Methods, Product, and Tools issues.

Under the People banner, the key factors were found to be their management, satisfaction, and training.

Methods were influenced by process models, testing, and quality assurance. The product was affected by design, re-use, and objectivity. Tools were thought necessary for configuration management, traceability, and reverse engineering.

Based on their findings the EPSOM team focused on the development of a traceability platform aimed at the need for capitalisation of knowledge. They attempted to model the dependency links that exist between the various objects / documents used or affected by maintenance. This model was based on a 'V' model for maintenance, comprising of 11 main activities ranging from problem understanding, localisation, solution analysis, impact analysis, through to regression testing, acceptance testing, and modification reporting.

The EPSOM team had a strong interest in traceability in the following areas: system understanding, problem understanding and localisation of the need for change, impact analysis, test selection and specification, documentation updates.

The traceability toolset that resulted from the project consisted of a program loader for structural analysis and indexation, and a repository of objects and links. Software was produced to manage the repository, browse the objects to facilitate problem understanding, analyse the objects, e.g. semantic analysis, propagation graphs, and to produce impact reports.

The traceability toolset facilitated in the selection of tests and documentation updates. However, the EPSOM model is mainly concerned with the implementation of maintenance rather than application management and the provision of quality services. Disciplines covering configuration and project management are prominent in EPSOM, but there is little evidence that service management and customer relationship management are considered within the 'V' model. At the operational level, EPSOM provides an effective maintenance procedure, albeit less focused on preventive maintenance and systems evolution.

4.5 Reverse Engineering into CASE Technology (RECAST)

This project was led by the CCTA and involved both the CSM and LBMS Ltd. The academic input was focused primarily on the transformation method, whilst LBMS were concentrating on tool support.

The goal of the project was to extract a non-loss representation of current Cobol software to be stored in a form amenable to analysis, and to be able to represent the results of the analysis in SSADM diagrams and notations.

The motive behind the project was largely pragmatic in that there exists numerous legacy systems in government departments that are poorly documented. Furthermore, the CCTA had a vested interest in the promotion of SSADM as a development method.

The RECAST structural model comprised of four stages, namely the identification of user views, the identification of the logical data model, the identification of menus and dialogues, and finally, the identification of the system processing.

The four stages resulted in a number of SSADM deliverables which are input into the SSADM Physical Design module.

The project recognised that dependency existed between the different products derived during the reverse engineering and that consistency need to be verified.

It was concluded by the CCTA that RECAST was a practical and useful method of reverse engineering, but that it was not the answer to all maintenance problems; they were particularly concerned about tool support and also the skills needed by the RECAST practitioners, particularly in terms of domain knowledge of both the business application and the implementation environment.

4.6 Software Life-cycle Support (ITIL Publication)

In 1993, the Software Lifecycle Support module within ITIL was published. This advocates closer co-ordination between customers of IT services, IT infrastructure managers and software developers with the ultimate goal of ensuring the provision of quality IT services which meet the demands of the business. The module recognises that customer involvement, software development and maintenance can be co-ordinated into a single approach aimed at reducing costs and managing change throughout the software lifecycle; from its design to retirement and/or replacement.

The CCTA's approach to the software lifecycle:

- incorporates IT infrastructure management requirements at the design and development stages;
- ensures that legacy systems are evaluated and where necessary their decommissioning is planned;
- helps to assure that service delivery meets the service level requirements;
- facilitates opportunities to consider new technologies which may be introduced by means of some re-engineering;
- helps to deliver and maintain systems which are more maintainable to help meet changing business demands over an extended economic life time;
- reinforces availability and quality requirements at an early stage;
- assures that systems are not only designed for maintainability, but can be readily supported throughout their life time.

The approach emphasises planning at a number of levels, including the strategic view of demand for IT services. The business strategy initiates work programmes in order to realise the strategic benefits, and programme plans are formulated in order to meet the strategic intent of the organisation.

The ITIL publication recognises the need for planning at various levels, including strategic, programme, and project. The issue of maintainability and the necessary relationships between IT infrastructure management functions is also recognised. However, the organisation issues, e.g. within the service organisation are not resolved, and there is too much emphasis on development models and methods, which are not wholly appropriate to the tactical and operational management of an applications service.

Software Life-cycle support is cognisant of the levels and management roles and relationships for applications management, but does not furnish the necessary framework for the design and delivery of an applications service.

4.7 Application Management Environments and Support (AMES)

The EU-funded Esprit Project: 8156 – AMES sought to undertake a detailed study of the domain of applications management and bring together practical experience gained in a number of EU countries and market sectors to form a methodological basis and framework.

The AMES project team included: Cap Gemini, Intecs Sistemi Spa, Matra Marconi Space, Valation Teknillinen Tutkimuskeskus (VTT), and the CSM. The project was thought to be beneficial by the partners concerned for three reasons:

- software maintenance and evolution was taking up valuable human resources;
- there was a strong case for support, either because the activities involve complex cognitive tasks that were not well supported by tools, or they involve a large element of time-consuming clerical work;
- major productivity gains could be obtained by better supporting these activities.

Hence AMES objectives were to provide support for applications management by means of:

- a methodical framework
- a set of tools
- a flexible integration infrastructure, and
- integrated application management environments.

The project was founded on previous research including: MACS, EPSOM, and also the classic studies of maintenance management, e.g. Leintz and Swanson ('78-'81).

However, applications management involves a number of activities and is much broader in scope than simply software maintenance. The AMES project recognised that in order to provide high levels of service, an applications manager must have effective strategies to deal with four classes of maintenance problem, namely:

- **Lack of organisational understanding**. It is well recognised that organisational issues have a large impact on the efficiency of maintenance activities.

- **Poor knowledge of the application system**. Knowledge of the application and its environment are important. It is necessary to understand the application domain, including the vocabulary. It is also necessary to understand the nature and purpose of the 'objects' handled by the system and how the system is used within the operational environment.

- **Low or weak process capability**. Formalisation of the software maintenance process is generally poorer than that of the development process and this is both detrimental to the application and also presents a poor image to the client, inevitably affecting the service adversely.

- **Lack of supporting tools**. Many of the maintenance activities, e.g. diagnosis and analysis of the software, impact assessment, configuration management, re-documentation etc. can be facilitated by software engineering tools. These are often neglected in the maintenance arena, due to budget constraints or the lack of business case for support tools.

AMES attempted to address these general problem areas by the development of a methodological framework supported by an integrated application management environment.

4.8 Foster's 7-model (British Telecom)

In 1993, Foster noted that the cost of maintenance was receiving some attention from the business perspective, but had received little academic attention, i.e. in terms of validated cost models, unlike development, during the '80's.

As a result of his research Foster proposed an investment cost model, that regards software as a corporate asset which can justify financial support in order to sustain its value. He proposed a business model allowing an organisation the ability to calculate return on investment in software by methods comparable with investment in other types of asset.

Foster's approach is to derive a model for assessing the financial implications of a proposed change in an activity, thereby providing a means to calculate both cost and benefit. By expressing the result in terms of ROI, the change can be ranked against competing demands for funding.

The model comprises of seven levels that can be broadly related to various management or decision making levels within any organisation. Within the model, the highest level perceives the software as an asset to the organisation , whereas at the lowest level there are technical details about the maintenance process, e.g. the allocation of time, reverse engineering techniques, methods and tools to support the maintenance task.

The structure of the model is given in the following table:

LEVEL	MEANING
Asset	Software as a company asset
Portfolio	The set of software items owned and used by the company
Network	A subset of the portfolio
Channel	The support chain for one software product
Team	A maintenance team
Function	A function performed within the team (e.g. request evaluation)
Topic	A component of a function (e.g. benefit assessment)

The model was developed for use by the internal maintenance and support management at British Telecom. It deals with the cost factors that affect maintenance, and the model represents a spectrum of activities from low level maintenance up to strategic decisions. The top three levels (network, portfolio, and asset) attempt to consider how applications are an asset to an organisation, and how the asset should be managed.

The model does relate to strategic objectives, e.g. identifying the need for maintenance, quality improvement, cost control, and resource management. Unfortunately, the model does not provide sufficient detail to populate an applications management framework.

The tactical level is represented by 'team' and 'channel', but the model does not address the objectives of contract negotiation and the promotion of the maintenance service. At the technical, operational level, represented by 'function' and 'topic', issues such as resource allocation are dealt with, indeed the model recognises groupings, similar to the better practice guidance on work packages, in order to exercise greater control of the application management service.

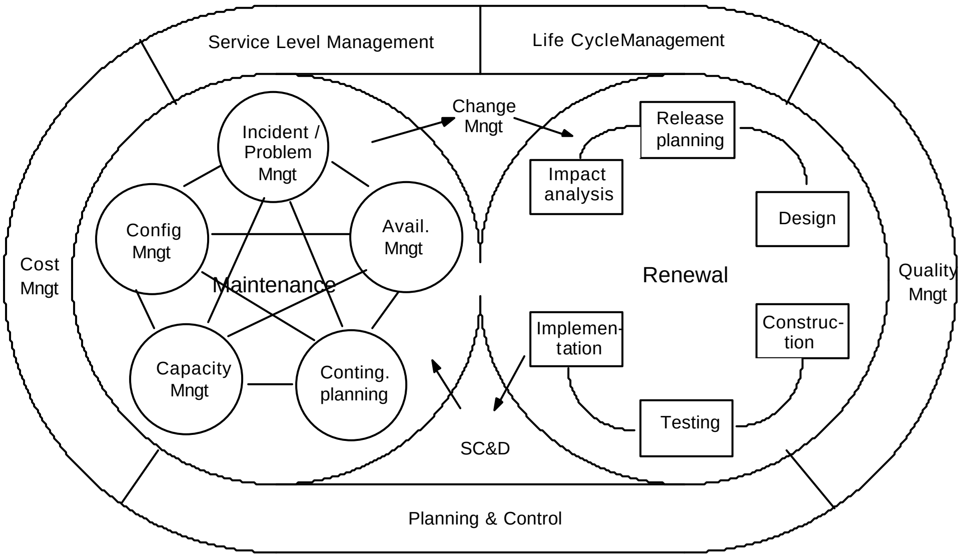
4.9 R2C (PinkRoccade)

The R2C model, see **Figure 2**, represents a management method for establishing and managing an application service oriented towards evolutive maintenance and service support. The model, is a business model that addresses also the customer interface, the quality expectations and the budgetary limitations imposed on the maintenance team.

R2C was developed to address the known problems with software maintenance management, and provide a standard framework that would promote this important business service. It can also be used to raise awareness of the potential renewal strategy available, particularly to those customers who may be familiar with ITIL concepts and terminology applied to service management.

R2C provides management direction, control and continuity with respect to application services.

Figure 2 – The R2C process model



The R2C model presents a process orientated approach to the organisation of ICT, thus creating a highly flexible ICT support organisation. The model depicts a situation in which there are service level agreements between the client and service provider on the required level of service provision. The service team functions as a single port of call for the client organisation. In order to deliver the ICT services, the service team steers the service provider and suppliers in accordance with the principles of cost and quality management. In addition, the service team is continuously involved in life-cycle management: on the basis of this, the service team generate a future-oriented ICT strategy in which continuous, gradual and controlled development forms the key element.

The processes concerned with making and monitoring the agreements on the quality and quantity of ICT service provision are depicted within the model as ‘service level management’. This process focuses on optimising service provision and assures that guarantees on service provision are honoured.

Life-cycle management covers the activities which underpin the support of the client’s business processes, through management, steering and operational support throughout there existence.

Cost management entails the processes concerned with managing and charging of costs incurred in ICT service provision. Financial management ensures that optimal consideration is given to price and performance. Good cost management from a relationship perspective can make the financial consequences clear and facilitate the decision on options with the client.

Quality management assures the quality of ICT service provision and establishes the quality of co-operation between client and supplier. The quality of the products is evaluated against norms and standards, assessments, and through extensive testing. Process quality focuses on the methods and procedures followed, and also on the level of personnel training and tool support.

Planning and control is envisaged as an extension of service level management, in which planning and control ensures that services are provided as agreed. A basic assumption made is that there is an explicit agreement with the client on what services are to be delivered and when the services are to be delivered.

The R2C model has a central hub which depicts the main technical process elements of ICT support, namely: exploitation, renewal, maintenance and development. A distinction is made between continuous and project-based processes. Renewal and development both require a project-based approach: they are aimed at achieving a particular target by means of a finite operation. Both exploitation and maintenance mainly consist of continuous processes aimed at maintaining the existing service provision.

Applications management provides an integral approach to ensure a problem free transition between these cycles. Two specific connecting processes are recognised within R2C, namely: change management and software control & distribution. Change management provides a mechanism for managing the initiation, registration, implementation and evaluation of changes to ICT. Implementing right first time (RFT) changes is necessary for continued service provision.

Software control & distribution ensures that there are safe working methods for the control and distribution of operational software. Such methods limit the risks of unauthorised access and can be seen as an 'air-lock' for service product offerings: adapted ICT components being transferred to the exploitation cycle.

Several generic continuous processes are necessary to maintain the running of the existing ICT infrastructure and applications. These cover a number of ITIL disciplines including:

- configuration management - concerned with registering and storing information on configuration components or versions;
- service desk - handling the contact with users with enquiries, complaints or incidents. First line support is provided to users and information is given on the consequences of change;
- availability management - concerned with monitoring and ensuring the availability of services and ICT components;
- capacity management - ensuring the optimal deployment of means to balance supply and demand;
- problem management - aimed at preventing or minimising disturbances in ICT service provision;

- contingency planning - deals with a wide range of measures which must be taken in case of disaster, such as fall-back facilities and back-ups.

The R2C model depicts renewal as a finite, project based cycle, encompassing the following activities:

- business impact analysis to determine the impact of the desired changes to the service provided;
- planning of the changes to be made and agreeing the costs and delivery dates. Before any changes are undertaken, a great deal of care is taken over defining the project, process and product requirements, planning, budgeting and project design.
- design, including the analysis phase;
- construction: realising and/or assembling the systems;
- testing the altered service components with final acceptance testing by the customer prior to sign-off;
- implementation with attention been given to conversion, training, instruction and migration.

Depending upon the client's business process affected, a method for renewal is agreed. There are a number of different scenarios recognised within R2C, namely: a phased approach, a staged approach (prototyping), incremental, and evolutionary approaches.

PinkRoccade have acquired considerable experience in the practice of R2C and lessons have been learnt. These include:

- the method can be too rigidly applied and should be considered as a framework which needs to be tailored to specific client engagements. For example, the model doesn't differentiate between the management of a single or many applications within a portfolio.
- the scope of the model covers both maintenance and renewal, but there is no strategic element, e.g. in terms of the renewal strategy, aligned to the business needs of the client organisation.
- the combination of the development and maintenance processes with the traditional ITIL service management processes provides a strong synergy on which knowledge may grow and be shared across teams.

R2C can be readily compared with the basic 3-level applications management model developed during the AMES project.

At the strategic level, R2C appears to address most of the objectives of applications management, and furnishes links to the tactical and operational levels. Possible weaknesses have been recognised in respect of relating business strategies and work programmes to maintenance needs, and perhaps in the area of marketing in regard to value propositions.

At the tactical level, the activities performed in R2C during the start-up process are concerned with the initiation of maintenance and support activities, e.g. service quality planning, preparing staff, ascertaining customer needs. These activities relate strongly to the tactical objectives of: effective negotiations, and project control. In addition, the review and renewal activities within R2C relate to service evaluation and perhaps smooth operational transition.

The technical level, is also well represented in R2C with reference to the tools and techniques to support both management and maintainers. Some aspects could be improved, e.g. impact analysis, risk management, and perhaps in objective setting and planning & control.

5.0 Challenges for the future

In section 2.0 we provided an introduction in terms of the importance of applications management and the problems reported which are common to both applications management and software maintenance.

In section 3.0 we time-lined the development of models that have been put forward to help managers understand and reason about the software maintenance process, over the last 30 years . Section 4.0 documents some of the more recent published frameworks and models applicable to applications management. The description of specific research projects has been given to base-line contribution in terms of advancing knowledge and the lessons learnt.

Based on our evaluation of existing frameworks and an understanding of the problem domain, a new framework for Applications Management is proposed, see Section 6.0. This section highlights the necessity for a new framework, based on the existing and future challenges.

The demands for a new framework are outlined below.

We have described how in 1982, the Parikh model recognised that it was necessary to identify the business objectives of what was to be achieved through change, as these were not always explicit. Furthermore, we discussed the Software Maintenance Model, produced by Hinley & Bennett, in 1992 in which the strategic direction as well as business objectives are denoted in terms of application requirements, so that the future maintenance strategy can be both justified and aligned with the business direction.

Challenge 1: Evolution of the applications portfolio is facilitated when the applications managers' have visibility of the business; this can best be achieved through mutual trust and co-operation, and hence a new framework must support the development of a mature relationship between the client and the service provider in terms of a strong partnership.

Challenge 2: In establishing, such partnerships applications managers need to transcend standard IS service management metrics and track performance and client satisfaction in such a way that there is visibility and accountability in the management of software assets. Clearly applications managers need to work with their clients to ensure that there are demonstrable benefits and realisable business value in the applications investment. (*Foster's model was an early attempt to look at ROI of maintenance activities*).

Many of the early models were focussing on a specific issue, perspective or aspect of software maintenance.

Challenge 3: Few models, have been demonstrated to be sufficiently flexible to be used in all application management scenarios and provide a suitable platform for the consolidation and integration of existing tools applicable to the various market sectors of interest.

The framework library has demonstrated that there are a number of de facto standards, e.g. ITIL for Service Management, SEI-CMM for software process assessment etc. Applications management can benefit from a more holistic approach which not only considers the process, people, and technology, but also orthogonal views of strategic, tactical and operational issues.

Challenge 4: Few models, have been demonstrated to be sufficiently flexible to be used in all application management scenarios, and to deal with the diverse and complex problems in providing an application service.

The research projects conducted in the 90's which looked at the problems of maintenance and reverse engineering, e.g. EPSOM, recognised that traceability was a key issue for the capitalisation of knowledge, leading to improvements in maintenance productivity, quality, and cost effectiveness.

Challenge 5: The proposed framework must provide a traceable pathway in which changes to individual applications and the evolution of the application portfolio may be managed successfully.

In functional terms, the maintainer needs to be supported in relating the request to the software components; this can be facilitated by providing 'views' of the code and through impact analysis tools. Possible views include that of: global structure, module structure, and code views, e.g. variable tracing. Tools may facilitate in terms of code visualisation, e.g. graphical representation of software.

The maintainer must also be able to reason about possible solutions (scenarios) to resolve problems and be able to justify a course of action. Furthermore, for maintenance and evolutive type work, it is also necessary to be able to consider the connectivity or interconnections between software entities. This aids both traceability and facilitates in the evaluation of both the software and the abstract model representing the application within its environment.

Challenge 6: The structure of many organisations has changed radically in recent years for a variety of reasons, including:

- The need to react quickly to changing circumstances and external market pressures;
- The need to recognise, develop and sustain core skills and competencies;
- The need to protect the 'know-how' of the organisation in terms of its products and services;
- The constant striving for improvements in the quality of the service product offerings and the ability of the organisation to innovate and maintain competitive advantage.

New organisational designs have emerged, e.g. the 'Clover-Leaf' model that depicts an organisation with a smaller, central organisational core in which core skills and competencies are sustained. The central core is supported by (i) part-time and temporary workers to meet the flexible and dynamic nature of the market, and (ii) contracted specialist and support services.

For many organisations, ICT is not considered a core skill or competency, although it may in fact be critically dependent upon IS for a number of its business processes. This presents one of a number of opportunities for ICT service organisations to meet the changes in their client's organisational structure, by recognising the potential demands placed upon them to not only maintain the applications systems which are critical to the client's business, but to work more closely in partnership with the clients organisation to provide a higher level of quality service in relation to applications support, evolutionary change and re-development.

There is a recognised need and hence a growing market for application services to:

- extend the effective life of the systems within the applications portfolio of the client.
- Provide a business perspective to software maintenance, such that it is no longer perceived as a technical activity, simply concerned with corrective or perfective work to fix faults or add functionality.
- Recognise the external demands placed on the clients business are changing (environmental change and market demands), and it is necessary to ensure that the critical application systems remain fit for purpose and that the client can react swiftly to changes in the business environment.

Challenge 7: To 'win' business in the application services market, it is necessary to have a clear and consistent strategy which minimises both customer and supplier risk.

An applications management strategy is required which can be presented to the client and which demonstrates that the applications which are critical to the business are being managed like any other business asset. The applications management strategy should include the following elements:

- Business and tactical reviews of the planned maintenance and evolution of the applications portfolio in the light of the emerging business needs;
- formal definition of the maintenance and evolutive processes and process management with suitable metrics to demonstrate process efficacy;
- development and installation of suitable tools to support both the managerial and technical activities in relation to the entire service product offering;
- planning and control of the service product offerings;
- regular assessments of the modified applications to ensure that service quality is not compromised (product assurance);
- release management and validation that customer expectations are satisfied.

To underpin the applications management strategy, the service provider needs to recognise and encourage a multi-displinary team, e.g. strategic planning, quality assurance, change management etc.

An important challenge for the future is in the management of legacy systems. Attempts have been made to deal with these systems in terms of renewal strategies involving reverse engineering and re-use, e.g. RECAST.

Challenge 8: A framework needs to be developed that is sufficiently robust to deal with legacy applications in all environments in terms of renewal, replacement, and retirement. Previous attempts have focused on either particular software environments, or have been hindered by the lack of tool support.

To 'win' business in what is likely to become an increasingly competitive market, the proposed framework must demonstrate that it underpins a commercial and measurable service.

Challenge 9: The application management framework must clearly focus on the provision of an application service as a value-added activity. Service provider and customer risks need to be recognised and are effectively managed through timely assessment and mitigation. The reliance and co-operation necessary between the various roles should be made explicit, in order that the relationships between the various disciplines and applications management can be defined.

5.1 Evaluation criteria for future frameworks

The surveys of the common problems associated with maintenance and applications management, taken together with the conclusions of academic and industrial research into these problems, has facilitated in the identification of a number of activities which are deemed necessary in order to achieve the goals of applications management.

These activities are primarily the concern or responsibility of different levels or roles within a service organisation and have been denoted as: the strategic level, the tactical level, and the operational level.

The strategic level bridges roles and responsibilities from the application owner to the service provider, and the following objectives, see **Table 1**, can be recognised:

- Identify the need for maintenance in relation to the existing business strategy and work programmes;
- Continuously strive for quality improvements in the service provision and deliverables;
- Manage the organisations resources efficiently;
- Identify and make cost reductions whilst maintaining quality standards;
- Assure the marketing strategy for application services.

For each objective, a list of applications management considerations has been compiled , and possible activities in order to accomplish these objectives.

Table 1: Strategic level objectives for applications management

Strategic level objectives	Considerations	Possible activities
Identify the maintenance need in relation to business strategy & work programmes	· assessment of current business processes · identification of problem areas · cost/benefits analysis · potential problems / risks · proposed actions / maintenance appraisals	Present viable business case Undertake risk analysis Manage the application portfolio
Continuously strive for quality improvements	· have a strategy for quality improvement · have defined work packages or project plans, where appropriate · monitor conformance to quality standards	Quality assurance Service reviews Ascertain levels of customer satisfaction
Reduce costs whilst maintaining quality	· increase efficiency	Monitor resource utilisation Mandate improvements
Manage organisations resources efficiently	· resource utilisation · operational efficiency · maintain and improve skills base	Defined procedures Training
Define a market strategy for the provision of application services	· identify appropriate markets	Marketing

The objectives and requirements clarified at the strategic level become prime drivers for tactical management, i.e. for the implementation of the strategic intent. In the realisation of the strategic vision application managers must meet their own objectives of:

- ensuring effective project control;
- negotiating and securing contracts with customers which minimise customer and supplier risks;
- promoting the use of their maintenance services;
- assuring smooth project handover and the effective management of systems release to customers;
- compliance with the organisational strategy.

Of particular concern for tactical management is the adoption of appropriate policies for dealing with the repair, replacement and retirement of legacy applications. The control of resources is also an important consideration at this level. It is necessary for tactical management to ensure that there is sufficient flexibility in the applications management process, to ensure that the desired goals are achieved to budget and other constraints, and that the desired quality levels are maintained. **Table 2** shows the considerations necessary by application managers and possible activities related to accomplishing their tactical objectives,

Table 2: Tactical level objectives for applications management

Tactical level objectives	Considerations	Possible activities
Effective project control	· defined project plans · achievement of project objectives	Priority setting Monitoring achievements against plan Risk and resource management
Negotiate and agree AM contracts with customer	· have available and adequately skilled staff, including negotiators · convey image and capability as service provider to the customer · ensure 'equitable' benefits are perceived by both parties	Produce an applications management plan for the customer Produce a cost breakdown statement for service provision Ascertain legal requirements of service provision Project a good corporate / professional image
Promote use of applications	· promote organisation /	Marketing of services, including technical demonstrations,

management services	service profile · demonstrate how the AM service will benefit the customer · show how the maintenance activities are attuned to the business needs of the customer	seminars etc.
Assure smooth handovers and release	· clearly defined acceptance criteria	Configuration management Document control Release control Acceptance and sign-off.

The operational level provide the maintenance and evolutive services to satisfy the objectives of the previous levels. Operational management, e.g. maintenance and support managers, dictate the services which are performed by the maintainers and support staff. It is necessary for operational management to liase at the service level with tactical managers to ensure that they are provided with the necessary resources, including tool support. Operational management also ensures that appropriate process and products metrics are defined so that sufficient data can be collected to inform applications management on performance, and for service improvements to be proposed.

Maintenance planning and other management activities undertaken at the operational level, ensure that maintenance processes are defined and that sufficient preparation is undertaken and competencies sustained for maintenance and evolution of the applications portfolio.

Preparatory activities undertaken by the maintenance teams, include: application understanding, design recovery, application analysis, problem analysis, problem localisation, impact analysis, and solution specification.

Such specifications are later used during maintenance to update the application and its supporting documentation. Applications are modified to comply with the solution specification and regression testing is undertaken to verify that modifications have not caused unintended side effects.

Maintenance managers also ensure that:

- comparisons of planned and actual maintenance work are conducted;

- appropriate acceptance testing is undertaken;
- modified software is (re-)installed into the operational environment successfully;
- documentation is changed to ensure that it remains consistent with the application;
- quality assurance reviews are undertaken;
- all changes to the applications are approved.

The proposed framework may be evaluated against the criteria presented above. The issues and limitations with the existing contributions described in the Application Framework Library have been discussed already.

In summary, weaknesses have been identified in the existing frameworks and it is intended to address these limitations in the design and development of a new applications management framework.

6.0 ASL: A new framework for Applications Management

This section describes the proposed framework.

6.1 Introduction

The proposed framework for Applications Management is based upon better practice for maintenance management, as described in the literature. We have discussed previously how maintenance has evolved from simply a 'black' art in the modification of existing systems, with little management insight, to a consideration of how the quality attributes are influenced by the actual maintenance process.

Empirical surveys conducted over a period of twenty years have shown repeatedly that the problems associated with maintenance are essentially managerial and not simply technical issues, concerned with tools and techniques.

Management have therefore sought frameworks, and in some instances process models, by which they might understand and reason about software maintenance practice.

We have discussed the ways in which Applications Management is much broader in concept than software maintenance management. The advent of better practice guidance for service support and service delivery in the 1980s-'90s, and in particular the CCTA's IT Infrastructure Library (ITIL) has provided some insights into aspects of service management that augment the management of maintenance.

However, this does not give a clear or comprehensive picture of Applications Management. For example, the service issues in respect of applications management requires more of a strategic perspective, coupled with tactical management decisions on how to sustain large and complex application systems to meet the dynamic and growing demands of the business.

Furthermore, if we look at ITIL and Applications Management from simply two aspects, namely in terms of support and delivery, we can observe that:

- ITIL contains a number of support functions and processes, e.g. Service Desk, configuration management, problem management etc. that are aimed at ensuring that the ICT infrastructure is supported and remains stable; the objective being to ensure that changes are always managed and do not compromise operational services.

- ITIL service delivery contains a number of management processes, e.g. availability management, business continuity management, capacity management, service level management etc. that are aimed at continuity of service to agreed service levels and quality expectations.

- Applications Management necessarily has a support function that may be driven by Change Management (an ITIL service support function). It may also be reactive to the Service Desk or Incident Management team in relation to application incidents requiring investigation or problem analysis.

- Applications Management in terms of delivery is focused on change with an emphasis on enhancement, renewal, renovation etc. to extend the life of systems within the application portfolio, and hence improving the business benefits profile.

We have discussed in detail the need for a framework in Section 2. For example, in providing application services, it is imperative that the provider can demonstrate how a quality service is to be provided that will meet the needs of the potential client. An applications management framework is extremely beneficial in such discussions.

We also discussed how cost considerations have been raising the profile, perhaps adversely, from the aspect of software maintenance management. However, considering cost from the service perspective provides an opportunity for Applications Managers. For example, the client can be actively engaged in budgetary decisions regarding the levels of service required in relation to fixed and variable cost elements.

The fixed costs are associated generally with the operational overhead and the support level. The variable cost element is associated with changing or developing application systems, e.g. perfective change requests can be scrutinised by a Change Advisory Board in view of the cost of change and the relative business priorities and business benefits. Similarly the business case for a systems enhancement project should be subject to both business and applications management review.

Applications management must also ensure that they have the necessary business and managerial reviews focused on improvements and minimising fixed costs associated with a level of service.

By simply facing the 'spot-light' and providing information on which the business can influence expenditure, e.g. to address business priorities, Application Managers can offer a professional service and alleviate some of the concerns and issues raised by clients over ICT development and maintenance costs.

Improvement in the quality of information between Applications Management and the client also leads to enhanced communication channels generally and provides the opportunity for relationship development and mutual benefit. The business benefits from greater influence / control of service costs and the ability to realise the benefits attributable to business support from application systems. The application service provider benefits from greater knowledge of business direction, objectives, and priorities; this information can then influence the tactical management decisions and support processes.

A fundamental aspect of the proposed framework is the recognition of levels or tiers, namely: strategic, tactical, and operational. This relates to one of the challenges presented in Section 5, namely that it is simply not sufficient to define processes, describe roles, and consider tool support for the process, but that it is necessary to address the perceived issues from a number of organisational perspectives. The orthogonal views can be linked in terms of cost and quality considerations – two of the most important drivers for Senior Management, that are also influenced by the process, personnel, and tool deployment.

The strategic level is content driven, i.e. in the consideration of the applications portfolio. The aim is to derive the strategic intent, i.e. the determination of the futures of the various applications within the portfolio. This involves a periodic review of the assets in order to ascertain that the client's organisation continues to have the right portfolio of applications in view of the changing business circumstances.

An important factor for Applications Management is whether the investments made by the client can result quickly in the achievement of realisable benefits, as far as the business is concerned. Evolution of existing systems can be a viable alternative to replacement, e.g. there are mutual benefits in the expression of both functional and behavioural characteristics, and through familiarity with the service domain.

This relates directly to the better-practices identified through the validation of the software maintenance model, see Section 4.1. *In developing a maintenance strategy, the software application requirements should be denoted in portfolio terms, so that the future maintenance strategy is both justified and aligned with the business direction.*

The strategic assessment of the applications portfolio is also important in assuring that the service provider risks are manageable, as well as in the development of appropriate maintenance strategies. Strategic investigations provide Application Managers with the necessary visibility if business direction and key objectives, and can be instrumental in customer relationship management.

The tactical level is directed by the strategic intent, e.g. in relation to policy – this might include: repair, retirement, replacement of specific legacy systems. Tactical management addresses the fact that longer application life expectancy presents problems in managing and exploiting legacy systems. These problems are usually associated with poor quality documentation, high complexity, and old technology.

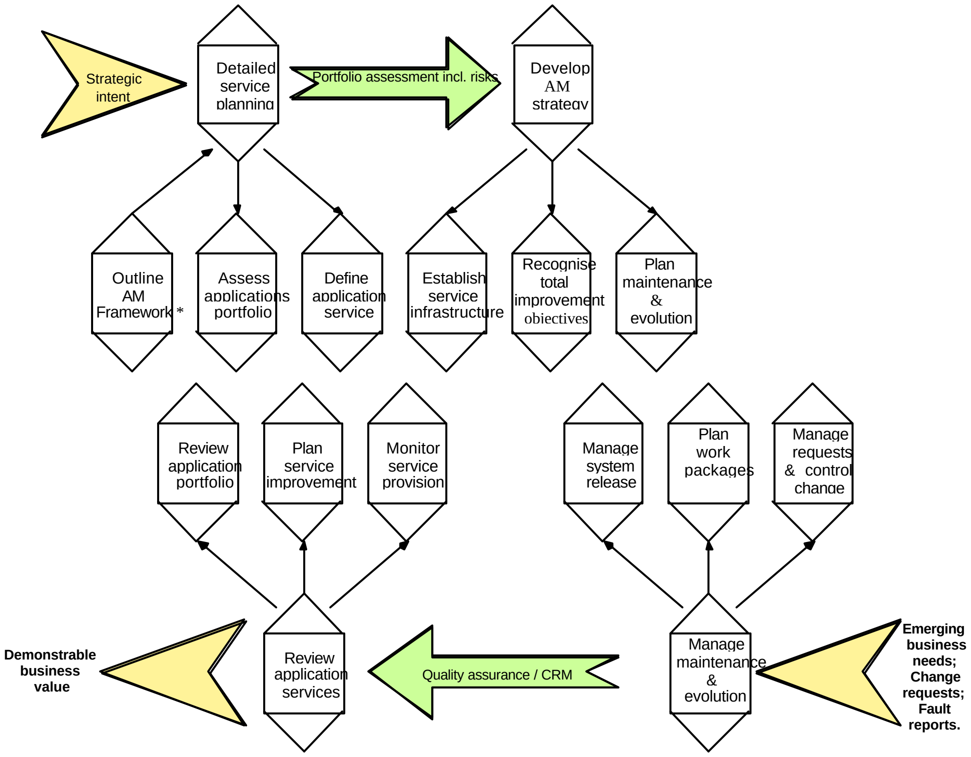
Tactical management is also necessary to address variable cost elements associated with application changes, and should be influenced by the customer directly.

We have noted Lehman's Laws of evolution previously, and clearly changes to applications are a necessary way of life. However, in managing the necessary resources and ensuring that applications are modified and enhanced quickly, reliably, and cheaply, it is necessary for Applications Managers to consider work packages and project planning and control aspects. The maintainability and other quality attributes need also to be addressed and that the customer is generally satisfied with the overall service offering, including support.

The operational support level represents not only the fixed support costs associated with service support, but also includes the maintenance and evolutive activities. Maintainers are responsible for identifying and implementing solutions to defined problems, and for ensuring that necessary changes are under control and the resultant application meets business requirements.

To summarise, applications management involves not only the steady state, service management aspects, and more reactive elements based on change requests or project proposals; it is also concerned at the strategic level with the business life-cycle, and is becoming a recognisable partner in demonstrating business value, see **Figure Q**.

Figure Q – High level activity diagram for applications management


Strategic intent
Detailed service planning
Portfolio assessment incl. risks
Develop AM strategy
Outline AM Framework *
Assess applications portfolio
Define application service
Establish service infrastructure
Recognise total improvement objectives
Plan maintenance & evolution
Review application portfolio
Plan service improvement
Monitor service provision
Manage system release
Plan work packages
Manage requests & control change
Demonstrable business value
Review application services
Quality assurance / CRM
Manage maintenance & evolution
Emerging business needs; Change requests; Fault reports.

The tactical and operational activities include aspects of:

- service management, i.e. service planning, service control, quality assurance, and service improvement;
- software maintenance: corrective, and preventive maintenance;
- systems enhancement, including: adaptive, perfective, and evolutive maintenance;
- user support, primarily aspects of service support including: a service desk function, problem management, and training.

The diagram represents four key activity domains for applications management. It is assumed that an Applications Management Framework is being used in order to undertake the detailed application service planning in order to achieve the strategic intent.

The strategic activities are associated with the recognition of an applications portfolio and an assessment in terms of service provider risks and the definition of appropriate application services to meet the business needs.

An appropriate applications management strategy can be derived following the portfolio evaluation. To realise the strategy, maintenance plans will be formulated alongside high level improvement objectives. An applications service infrastructure will also need to be established based on the budget and needs of the client.

Day-to-day operational and technical management activities are associated with the planning and control mechanisms, including request control, change management, and release management. These control mechanisms are facilitated by workload planning, including project management, and the definition of work packages.

The cycle is completed by the review process, which examines not only the products, i.e. applications, but also is concerned with service improvement generally. The review is at three levels, namely the client's business, applications management, and technical levels; the results being fed back into the strategic intent and planning processes.

Parallels may be drawn to existing management disciplines, e.g. service management, software maintenance management etc., however, the activity domains are noticeably different within the sphere of applications management, i.e. there are different task co-ordination mechanisms, dependencies, and role interactions.

For example, the review of application services might appear to simply involve service management. But within this context, application managers are concerned with demonstrating 'value' to the business in line with the priorities and budget set by account management. Additional activities are undertaken to assure customer satisfaction in the longer term and in developing the relationship between customer and service provider. Strategic relationships are developed to ensure that the service needs of the customer are recognised within the planning functions and that the maintainers are clear about how the functionality needs to be delivered, i.e. addressing the customers fundamental operational needs.

The latter point assures that even systems enhancement isn't simply a technical activity, it also has a service perspective embodied within an evolutive process in terms of addressing the customer needs to respond to changing business demands and to continuously improve the applications service.

One final and important aspect of the activity diagram is the service infrastructure. This not only covers the resource distribution for maintenance teams, but user support aimed at ensuring the users make effective use of the applications, and also ensuring that any problems are managed effectively to resolution. In this area, better practices may encompass those of ITIL service support to some extent, but these are augmented with the need to consider the longer term training, skills and competencies of both users and service providers.

6.2 Proposed framework

The proposed framework for applications management, is derived from the principles and ideas presented earlier.

The purpose of the management framework is to be able to describe applications management in its depth as well as its breadth, i.e. so that the reader can appreciate the activity domains and the aspects which are of interest to the reader or require management attention. It may also be used to facilitate awareness and understanding.

It is intended that the framework will support three perspectives (levels of management), i.e. strategic, tactical, and technical (or operational). It can therefore be readily evaluated against the criteria presented in Section 5.1. A second criterion for the several clusters in the framework implies the distinction whether a process is supporting a service function or an application function.

Application Management was defined as: the contracted responsibility for the management and execution of all activities related to the maintenance and evolution of existing applications, within well defined service levels. This means: the support of the business processes by information systems, for the full life span of these business processes.

Thus Application Management contains two essential elements:

- 'The support of the business processes by information systems': making sure that the applications really work and taking care of the support of the everyday activities of the organisation. This means a continuous service delivery by:
    - making good agreements on the service level;
    - quick recovery of the agreed service levels when they are not met;
    - an application management team that is accessible to questions and remarks of customers about the services;
    - prevent incidents by having a pro active attitude.

Thus, the focus is on services, the actual service that is provided and makes it possible (together with infrastructure management) that applications are being used. Usuallly this accounts for 10-20% of the total Application Management.

- 'For the full life span of the business processes': organisations evolve, environment and market change. Therefore, the supporting information systems have to change to keep on functioning optimally. These processes are concerned with enhancement of applications to future wishes, demands and needs, both technical and functional. These application related processes account for the major part of the application services.

On these grounds operational processes, tactical controlling processes and strategic policy processes can be distinguished. This leads to the framework depicted in Figure X.

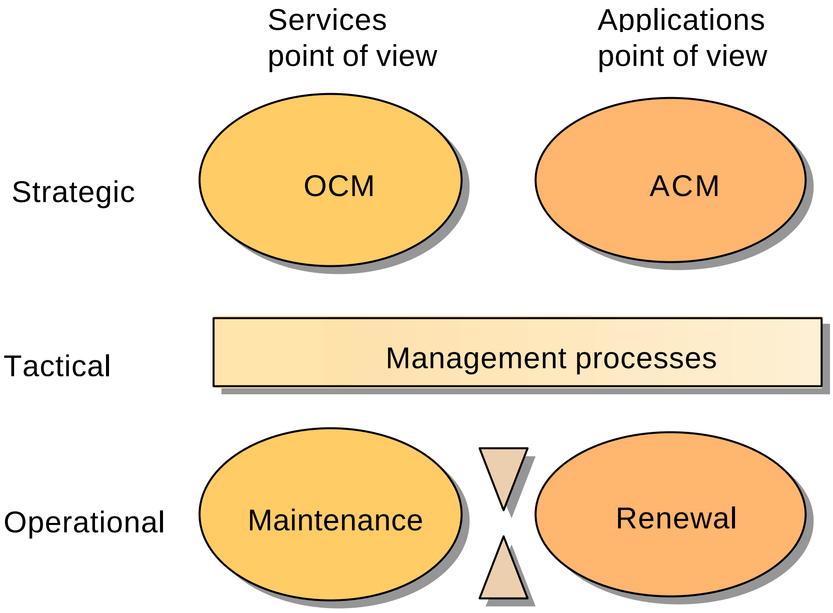


Figure X The ASL Framework

The ellipses and the rectangle in the middle represent a cluster of processes.

The *operational level* recognises two process clusters:

- ‘maintenance’ of applications: the processes in this cluster make sure that the existing applications are optimally used for the support of the business process, with a minimum of resources and disturbances in the operation.
- ‘renewal’ of applications: these processes make sure that applications are adapted to new wishes and demands as a result of changes in the organisation and her environment. In these processes the needed changes are actually made in the software, the data models and the documentation.

The tactical level accounts for the control on the operation and contains the overall management processes. These processes provide a mutual management and control of both the continuous service processes and the project-based application processes. Input for the management processes is provided by the strategic and the operational level. Therefore both orientation to the future and daily reality are strongly and explicitly anchored in these processes.

The strategic level again recognises two process clusters, based on the differentiation between the ‘service’ and the ‘application’ point of view. The service provider of today (for both operational services and systems enhancement) is not necessarily forever the service provider. Many arguments can be found why it is better for the future of an application that the services are provided by another supplier. There will be more competition between the suppliers on the services that are to be provided. The distinction between services and applications allow for an optimal policy.

The process clusters on the strategic level are:

- Organisation Cycle Management: the processes in this cluster are directed towards developing a vision for the future of the ICT-service organisation and translating this vision into a policy for the renewal of the ICT service organisation.
- Applications Cycle Management: the processes in this cluster lead to a long term strategy for the individual applications and for the total information provision of an organisation, with respect to the long term policy of the organisation.

6.3 Process descriptions

6.3.1 Maintenance processes on operational level

Maintenance of information systems on the operational level distinguishes the following aspects:

- The *identification* and *maintenance* of several service objects (an object can be: an application, an interface between teo applications, a component, a database, etc.);
- the *availability* and the *quality* of these objects;
- the deployment of the right *capacities* and *assets*, the right resources in the right amounts, necessary for the service provision;
- the *questions, wishes* and *defects* about or in the objects that are part of the service provision agreed on.

These aspects can be recognised in the definition of the maintenance processes:

- *incident control:* a service desk takes care of the contact with end users and/or functional application managers, handles service calls (complaints, questions, incidents reported via the *help desk)* and provides support on the uses of ICT means. In addition, the help desk provides users with information on the consequences of changes to the ICT service provision. By performing structural analyses on the registered service calls insight is achieved in desired improvements.
- *configuration management* is concerned with registering and storing information on (versions of) configuration components like software and documentation;
- *availability management* is concerned with the processes ensuring, monitoring and guaranteeing the availability of services and ICT components;
- *capacity management* is the process concerned with ensuring the optimal deployment of means, i.e. in the right place, at the right time, in the right amount and at justifiable costs;
- *contingency planning* deals with the wide range of measures which must be taken in the event of a disaster, such as fall back facilities and backups.

6.3.2 Renewal processes on operational level

The development, enhancement and/or renewal of ICT components will occur from the perspective of the renewal scenario and takes place in a project-based manner. In general the following activities are always carried out:

- *impact analysis*: object analysis of the client's primary processes with which the system is concerned in order to determine the impact of the desired changes to the service provided;
- *design*: design of an information system in addition to a more detailed information analysis;
- *construction*: realising and/or assembling altered components;
- *testing*: testing the altered service components before delivery to the client;
- *implementation*: the actual implementation of the altered components. Attention is given, among other things, to conversion, training, instruction, migration and the final acceptance test followed by discharge by the client.

Before the actual realisation is started much attention is given to the project definition and initiation: agree upon the project, process and specifications, planning and budgeting.

6.3.3 Connecting processes between maintenance and renewal

On operational level the following processes, that connect the maintenance process cluster and the renewal process cluster are defined:

- *Change management* is the process in which is determined which requests for change are implemented in a 'release'. This process, in which the client is strongly involved, and that is validated by the impact analysis, results in an agreement on the alterations that will be made, on the scheduling, costs and delivery dates. Change management is therefore the actual entering 'airlock' to adaptive maintenance and renewal.
- *Software control & distribution* is concerned with the processes involved with the control and distribution of operational software. That is to say, a safe working method that limits the risks of unauthorised use and unauthorised alteration or deletion. This process can be seen as an 'airlock' for products: adapted ICT components are transferred to the exploitation cycle.

6.3.4 Management processes on tactical level

The management control processes have the following areas of interest:

- *time:* delivery time, needed capacity and effort;
- *money:* the amount of finances involved in the total service provision;
- the *quality* of the services provided and the way in which this is guarded;
- the *agreements* with clients and suppliers.

In the ASL vision the control processes are integral: the interests and choices of the other process clusters are input for the management processes and these interests are continuously balanced. Exactly all these factors allow for an integral planning and management, as well regarding to the planning and control of releases as of for instance a team. This leads to a situation in which strategy is anchored on the working-floor and in which practice in maintenance

and renewal is translated to higher management levels.
These processes do not only control the on going business but look at the future as well. Of course the recognition of possible risks and the associated measures (parts of risk management) are an integral part of the control processes.

The areas of interest can be found in the tactical management processes:

- *Planning and control:* is envisaged as the management of time and capacity concerning all the processes that are involved in maintenance, enhancement and renewal of applications. The simultaneous control of project-based renewal activities and continuous maintenance activities, often being carried out by the same department and the same people, is one of the big challenges of application management.
- *Cost management:* entails the processes concerned with managing and charging of costs incurred in ICT service provision. Financial management ensures that optimal consideration is given to price and performance. Good management of costs (and proceeds) from a relationship perspective can make the financial consequences clear and facilitate the decision on options with the client. The choices are made together with the client.
- *Quality management* assures the quality of ICT service provision and establishes the quality of co-operation between client and supplier. The quality of the products is evaluated against norms and standards, by means of assessments and through extensive testing. Process quality focuses on the methods and procedures followed, the tools used, and also on the level of personnel training. Therefore, competence management is part of Quality management, as is problem management.
- *Service level management* is the process concerned with making and monitoring agreements on the quality and quantity of ICT service provision. It is oriented to optimising service provision and guarantees the forming of integral agreements on ICT service provision and has the following objectives: agreement upon an optimal level for service provision, controlling this service level and reporting about it.

6.3.5 Processes on strategic level : Organisation Cycle Management

These processes are about the life cycle of the services of the ICT service provider and the adjustment of the ICT service organisation as a consequence of it. The relationship between the ICT service provider and the user organisation becomes less and less self-evident; think of outsourcing, privatising and Application Service Providing (ASP). These developments have a strong influence on the user organisation, but even more on the application services provider. Therefore, the ICT organisation needs to be working more consciously on the tuning of the application portfolio with the needs of the market.

In this process cluster the strategy is defined:

- What does the ICT service provider have to do to keep on providing the desired service level on longer terms?
- What does the ICT service provider have to do to keep or replace the market?

The following processes are recognised:

- *account definition management:* in this process is defined which image, strategy, organisation and approach are necessary to realise the new servi ces and access desired markets;
- *market definition management:* based on the analysis of developments in the market and in client organisations and the trends that are observed from this analysis it is decided which services are desirable for which parts of the market and which requirements upon the services that are to be provided are essential;
- *service delivery management* is the process in which is mapped which services and service levels are desired on the market and to what extend these can be provided by the ICT service organisation based on the present skills; these are translated to a policy and strategy for the services that are to be rendered in the future;
- *skills management* is the process in which is defined which skills and knowledge are necessary, have to be acquired or extended, to be able to provide the services that are desired in the future and is defined how this expertise is supported by and spread within the organisation;
- *technology definition management:* in this process the (develop)tools, technology, methodologies, et cetera that are necessary to realise the services that are desired in the future, are reviewed and chosen.

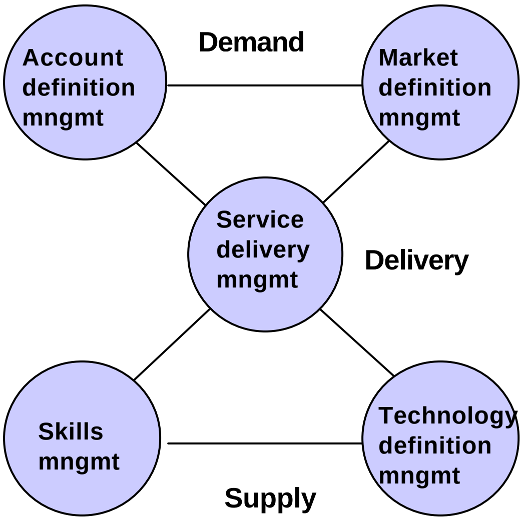


Figure Y: Strategic processes within Organisation Cycle Management

6.3.6 Processes on strategic level : Applications Cycle Management

Applications Cycle Management (ACM) is aimed at the future of service provision and the life cycle of the existing objects in the information provision. This takes place on two levels: on the level of the application and on the level of the level of the complete application portfolio that supports a certain business process.
Trend watching has to be done in the fields: technology, the business processes within the user organisation and the environment / chain of the user organisation.

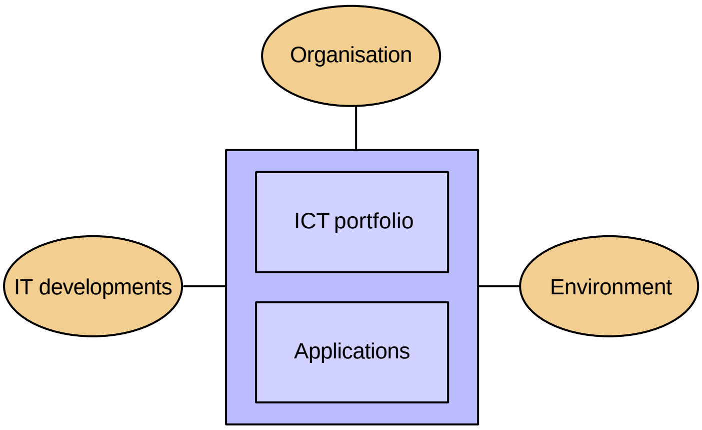


Figure Z: ICT portfolio management

- *ICT-portfolio management* (see also fig.) maps out the meaning and the performance of the existing applications for the client organisations, translates the company policy to the different objects, and defines a strategy for the future for the objects in the ICT-portfolio. This process has much in common with activities like information planning and information management.
- *Life cycle management* defines a strategy for an application. In a fundamental way the match is made between the present possibilities and the future requirements to one or more applications that support a business process. Next a strategy is defined to be able to meet the future requirements. This process is more thorough, has more content and is more strongly aimed at a specific business process than the former process.

The input for these two processes is partly provided by the processes that take care of the external side of the ICT-portfolio and applications. The ICT market place is a dynamic one and developments in the universe of discourse of the user organisations are getting more and more important.

- *ICT-development management:* this process defines which new ICT developments may be interesting for the user organisation and its information provision. Most important is the technology used for application development, but also new infra structures (like networks, audio, video, etc.) can create possibilities that have impact on the applications.
- *Organisation environment management:* this strategic process makes the requirements and opportunities visible for the applications and information provision with respect to the user

organisation in the light of chain developments. There is a tendency that organisations operate more and more as a part of a chain of organisations. This results in strong dependencies between the information provision of the organisations involved. The potentialities of the own in formation provision determine the position of the organisation in this chain of processes.

- *User organisation strategy:* this process maps out the developments in the user organisation itself, the impact of that for the applications, which hindrances (yes or no technical) are to be expected from the applications and how the organisation and applications can be prepared for this; all of it pro-actively.

7.0 Summary & conclusion

This section describes how the Barracuda Content Design meets the stated quality expectations and the desired outcomes as specified for the project.

The Barracuda Content Design provides a textual and graphical description of applications management and its relationship to service provision. Much of the terminology has been defined in Appendix A: Glossary of terms.

Available Roccade documentation and other public domain frameworks, (many of which have been produced by collaborative efforts of both industry and academia), have been used in the development of the content. The author also acknowledges the contributions of:-

- Marco Pastors
- Tom van Sante
- Remko van der Pols
- Machteld Meijer
- Ben Stoltenborg

The content has been focused primarily on applications management, and relationships to existing better practice guidance have been highlighted.

It is anticipated that the Content Design will provide a suitable foundation for the development of internal Handbooks for Roccade personnel with an active involvement in Application Services.

The applications management framework highlights a number of requirements in order to provide a professional applications service. It also provides three perspectives which are aimed broadly at strategic, tactical, and technical management.

To summarise the framework, it is proposed that to provide a high quality professional applications service it is necessary to:

- manage applications in portfolio terms, i.e. recognise that applications are a business asset that can justify financial support in order to sustain their value.
- identify and assess the risks of maintenance and assure that customer benefits are realised.
- have a valid maintenance strategy for the maintenance and renewal of legacy systems, with the purpose of extending the useful life of existing applications, in a cost effective and timely manner.

- ensure that IT infrastructure management requirements are incorporated during the design and development stages of applications and that maintainability and other quality attributes are considered throughout the applications life.

The proposed framework has addressed also the challenges for applications management, including:-

- the development of mature relationships between client and service provider;
- visibility and accountability for the management of software assets;
- meeting business needs through the timely recognition of potential demands to be placed on applications in support of business processes.

The proposed framework also overcomes some of the limitations recognised for R2C, including: scalability, the lack of strategic direction and visibility of business plans, and benefits realisation.

Appendices

A: Glossary of terms

Acceptance
A maintenance phase performed after implementation during which the service product is checked by the customer in order to verify that it is either: an acceptable solution to a reported problem, or that it fulfils the contractual requirements and meets the customer specification.

Acceptance criteria
Key statements of user requirements or project objectives which define the acceptable level (quality characteristics) that must be demonstrated before the customer will sign-off and accept the product or service.

Adaptive modification or enhancement
Changes made to an existing application so that it will continue to be usable in a new or modified environment, including a changed database or processing environment.

Application
Totality of system features covering a defined area (process) of interest to the business, Such features include software functions, user procedures and service procedures.

Application management
…..

Audit
An independent examination of work products to assess compliance with specifications, standards, contractual agreements, or other criteria.

Baseline
A specification or other product that has been formally reviewed and agreed upon, that thereafter serves as the basis for further development, and that can be changed only through formal change control procedures. Baselines provide control and stability when interacting with customers.

Capability Maturity Model (CMM)
A description of the stages through which software organisations evolve as they define, implement, measure, control, and improve their software processes. The model provides a guide for selecting process improvement strategies by facilitating the determination of current process capabilities and the identification of the issues most critical to software quality and process improvement.

Change
Enhancement (extension or improvement) or maintenance (upgrade or correction) applied to a system. During initial systems development, changes might be corrections due to non-conformity with agreed specifications.

Change management
A defined management process for dealing with requests for change to any aspect of the ICT services provided and/or application systems supported, evaluating the potential and actual impact of the change, arranging for authorisation and implementation of the change.

COCOMO
The Constructive Cost Model for the estimation of software development costs based on the structure of the software system and a variety of cost drivers. COCOMO was developed and introduced by Barry W. Boehm.

Configuration management
The process of identifying and defining the configuration items in a system, controlling the release and change of these items throughout the system life cycle, recording and reporting the status of configuration items and change requests, and verifying the completeness and correctness of configuration items.
Configuration management provides technical and administrative direction and surveillance to: identify and document the functional and physical characteristics of a configuration item; control changes to those characteristics; record and report change control and implementation status.

Consistency
The degree of uniformity, standardisation, and freedom from contradiction among the documents or parts of a system or component.

Contract
Document which specifies the product or service to be supplied. It lays out the legally binding agreements, warranties and responsibilities established between the service provider and the client.

Corrective maintenance
Maintenance performed specifically to overcome reported faults that impair the availability or use of an application. Such work is often prioritised in terms of its criticality to the business: critical, important, minor, cosmetic.

Defect
A flaw in a system or system component that causes the system or component to fail to perform its required function. A defect may cause an application failure during execution.

Design recovery

A particular form of reverse engineering in which the focus is on obtaining some design for the system (not necessarily the original design, but a design which can aid understanding of the system, and further maintenance or re-engineering activities).

Enhancement
The result of a request for change in functional or non-functional requirements that were not specified originally for the existing application system.

Evolutionary enhancement
The provision of additional functionality to the application, as a result of evolving business requirements.

Formal review
A formal meeting at which a product is presented to the end user, customer, or other interested parties for comment and approval. IT can also be a review of the management and technical activities and of the progress of the project.

Framework
………

Function
A set of related actions, undertaken by individuals or tools that are specifically assigned or fitted for their roles, to accomplish a set purpose.

Impact analysis
A phase of maintenance in which the potential impacts of modifications to the system are assessed. The process determines the consequences of a change to an application, in terms of other parts of the application or its environment which need to be modified as a result of the change. Techniques such as slicing, tracing, and data-flow analysis are typically used.

ISO9000
A series of international standards concerning quality management and produced by the International Standards Organisation (ISO). They have been adopted by the European Standardisation Committee under the reference EN29000.

Key practices
The infrastructure and activities that contribute most to the effective implementation and institutionalisation of a key process area.

Key process area
A cluster of related activities that, when performed collectively, achieve a set of goals considered important for establishing process capability.

Key process areas have been grouped by the SEI within particular maturity levels to help determine the software capability of an organisation and understand the improvements needed to advance to higher maturity levels. For example, Level 2 key process areas are: requirements management, project planning and monitoring, subcontract management, quality assurance, and configuration management.

Life-cycle ????
A structured approach to planning software projects by defining phases, activities, and tasks, indicating responsibilities, describing corresponding deliverables, and stating verification and tuning guidelines.

Maintainability
The ease with which an application system can be modified to correct deficiencies or enhanced to satisfy new requirements. There are a number of criteria which can be measured to establish whether a system is easily maintainable, including: the standard of documentation, the quality of the software, the state of the maintenance environment, and the standard of existing procedures.

Maintenance
The co-ordinated activities that enable an existing system to be sustained, according to agreed rules and procedures, by means of receiving requests for change, fault reports, and applying modifications to the system.

Maintenance environment
The environment in which changes to existing systems are developed and tested.

Managed and controlled
The process of identifying and defining software work products that are not part of a baseline and therefore not under configuration management, but must be controlled for the project to proceed in a disciplined manner.

Maturity level
A well-defined evolutionary plateau toward achieving a mature software process. The maturity levels in the SEI - CMM are: initial, repeatable, defined, managed, optimising.

Methodology
A collection of methods, procedures, and standards that defines an integrated set of engineering approaches to the development and/or maintenance of a service product offering.

Milestone
A scheduled event for which some individual is accountable and that is used to measure progress.

Model
A representation of something actual or contemplated with relevant characteristics, the same as those of the actual thing modelled. The features modelled are dependent upon the circumstances and intended use of the model. It is usually developed from a particular perspective and for a particular purpose.

Policy
A guiding principle, typically established by senior management, which is adopted by an organisation or project to influence and make decisions.

Preventive maintenance
Maintenance carried out to prevent malfunctions or improve maintainability. Unlike other categories of maintenance, preventive maintenance is not usually triggered by an external event.

Problem management
The administration of problem progression and analysis of the trends in problem reports and resolution.

Process capability
The range of expected results that can be achieved by following a process. It is often compared with a baseline or documented characterisation of the typical outcome following a specific process under typical circumstances.

Project
An undertaking with specified criteria that requires concerted effort and is focused on developing and/or maintaining a specific product. Typically projects have their own budget and specific time-scales and constrained resources.

Project manager
The person with business responsibility and who is accountable for the project's success in full compliance with contractual commitments / requirements. The project manager directs, controls, administers, and regulates the project.

Project plan
A document that describes the technical and management approach to be followed for a project. The plan typically describes the work to be done, resources required, schedules to be met, and the way the project will be organised.

Prototyping
A technique in which a preliminary version of the system is developed to permit user feedback, determine feasibility, or investigate timing or other issues in support of the development process.

Quality
The totality of features and characteristics of a product or service that bear on its ability to satisfy specified requirements. In a contractual environment needs are specified, whereas in other situations it is often necessary to identify and defined implied needs. Quality may be assessed in terms of the degree to which the application or service component meets customer and user needs or expectations.

Quality assurance
Comprises of all those planned and systematic actions necessary to provide adequate confidence that the given requirements of a process definition are being satisfied.

Quality management
An aspect of the overall management function that determines and implements the quality policy. Quality management can include strategic planning, allocation of resources and other systematic activities for quality such as quality planning, operations, and evaluations.

Re-documentation
The creation or revision of a semantically equivalent representation within the same relative abstraction level. The aim is to create alternative views of a system to aid understanding, and to re-create documentation which once existed or should have existed, or to regenerate such documentation after a change to a system.

Re-engineering
This refers to the process of reverse engineering followed by forward engineering to obtain a new version of the system which (for example) adheres to new quality standards.

Regression testing
Selective re-testing of unmodified functionalities, to detect faults introduced during the modification of a system component, and to verify that modifications made have not caused unintended adverse effects, and that the system still meets the specified requirements.

Release
Formal version of a system, grouping together various elements which may be implemented at the same time, i.e. as a coherent whole.

Reliability
The ability to perform a required function under stated conditions for a stated period of time.

Renewal
…….

Requirement
A documented representation of a condition or capability that (i) must be met by a system or system component to satisfy a contract, standard, specification, or (ii) is needed to solve a problem or achieve and objective.

Reverse engineering
The process of analysing a system to identify its components and their interrelationships, and to create representations of the system in another form or at higher levels of abstraction.
Reverse engineering is an analytical process performed on existing software with the primary purpose of creating objects in a meaningful and convenient format to assist future maintenance work including systems enhancement.

Ripple effect
The phenomenon by which changes to one program area have tendencies to be felt in other program areas.

Risk management
The technique employed to find and rectify the main factors likely to jeopardise the successful delivery of the service product offering.

Role
A unit of defined responsibilities that may be assumed by one or more individuals.

Service
A general term used to refer to work performed for a client as part of a continuous process, such as applications management, rather than discrete pieces of work such as a project.

Service contract
A document that explains the responsibilities of the supplier and those of the client; the detail specification of the service may be specified in a Service Level Agreement (SLA).

Service level
An agreed standard of measurable service which may be included in a contractual agreement. For maintenance, service levels may include agreed response times to client requests or resolution times for problems.

Service level agreement
A document that identifies what is to be delivered and the quality level agreed in objective and measurable terms. It is a joint client-supplier document and acts as a service management tool.

Service management
Service management covers a range of activities which ensures that the application services supplied remains relevant to the client's need and continuously delivers client satisfaction. Service management should address the client's need for both quality and flexibility from the service supplier and endeavour to ensure that costs are minimised through effective resource management and the implementation of better practices and tools for increased service efficiency and effectiveness.

Software life-cycle
The period of time that begins when a software product is conceived and ends when the software is no longer available for use. The life-cycle typically includes a concept phase, requirements phase, design phase, development and test phase, implementation, operation, and maintenance phases.

Software maintenance

Modification of a software product after delivery to correct faults, improve performance or other attributes, or to adapt the product to a changed environment.

Specification
Document providing a definitive description of a system for the purpose of developing or validating the system. Usually, system specifications emerge from user requirements and are refined into functional specifications and technical specifications.

Stage
A partition of effort that is of manageable size and that represents a meaningful and measurable set of related tasks which are performed by the project. A stage is usually considered a subdivision of the software life-cycle and is often ended with a formal review prior to t he start of subsequent stages.

System
A collection of components or elements, that are interrelated and organised to accomplish a specific function or set of functions. Systems possess properties that are different from the properties exhibited by the individual parts.

Testing
The process of executing a program, or part of a program, with the intention of finding errors.

Traceability
The ability to identify the relations between the various software items produced at various stages of development, and changed or added during the operation and maintenance phases.

Verification
Activities designed to evaluate whether the products of a given development phase satisfy the conditions imposed at the start of the phase.