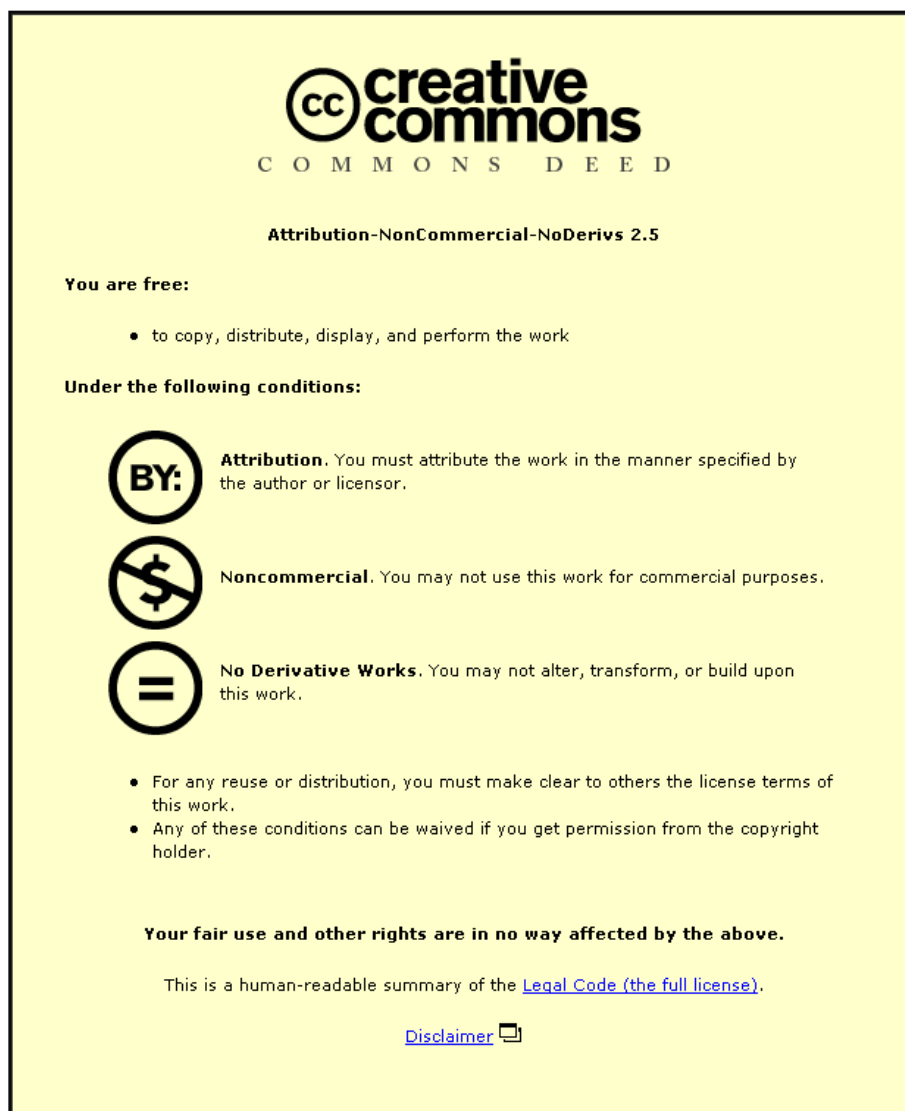




This item was submitted to Loughborough's Institutional Repository (<https://dspace.lboro.ac.uk/>) by the author and is made available under the following Creative Commons Licence conditions.



CC creative commons
COMMONS DEED

Attribution-NonCommercial-NoDerivs 2.5

You are free:

- to copy, distribute, display, and perform the work

Under the following conditions:

BY: **Attribution.** You must attribute the work in the manner specified by the author or licensor.


Noncommercial. You may not use this work for commercial purposes.

No Derivative Works. You may not alter, transform, or build upon this work.

- For any reuse or distribution, you must make clear to others the license terms of this work.
- Any of these conditions can be waived if you get permission from the copyright holder.

Your fair use and other rights are in no way affected by the above.

This is a human-readable summary of the [Legal Code \(the full license\)](#).

[Disclaimer](#) 

For the full text of this licence, please go to:
<http://creativecommons.org/licenses/by-nc-nd/2.5/>

Providing demonstrable return-on-investment for organisational learning and training

Michael Elliott

AWE plc, Aldermaston, Reading, UK

Ray Dawson and Janet Edwards

Department of Computer Science, Loughborough University, Loughborough, UK

Abstract

Purpose – The aim of this paper is to present a holistic approach to training, that clearly demonstrates cost savings with improved effectiveness and efficiencies that are aligned to business objectives.

Design/methodology/approach – Extending Kirkpatrick's evaluation framework with Phillips's return-on-investment (ROI) concepts, the paper conveys a number of successes; including trainee satisfaction and the capturing of improved knowledge and skills.

Findings – The paper includes case studies of how, and to what extent this knowledge has been applied with examples of resulting efficiency savings. The paper shows that there is growing agreement that one of the primary drivers, if not the key driver of long-term organisational effectiveness, is the ability of an organisation to learn effectively. The methodology requires some additional assessment and course preparation to establish a basis from which to demonstrate learning effectiveness. The financial benefits of the applied learning are far greater than the additional preparatory costs.

Practical implications – Learning organisations that anticipate, react to change and learn, are likely to maintain a competitive advantage. These organisations are constantly looking for more effective and efficient ways of training. Paradoxically, other organisations will often slash training budgets in times of hardship, as training departments are unable to demonstrate the effectiveness of their programmes.

Originality/value – The paper presents a practical example of how training should be applied to truly demonstrate its value in the workplace

Keywords Training evaluation, Learning, Skills, United Kingdom

Introduction

A training programme for teaching the concepts of software engineering and related support practices was introduced into AWE plc in response to a need identified from a comprehensive software quality management system (SQMS) implementation assessment (Elliott et al., 2006). The assessment uncovered a lower level of adoption of the SQMS than would normally be expected for an ISO 9001 certified company. This was due, in part, to a lack of knowledge and understanding of some of the software quality and engineering principles. A characteristic of this organisation is that a significant proportion of software development is performed by scientists and engineers in specialised units who, although they are highly qualified and experienced, have not been trained in software engineering, but have undertaken software development, to support their main role. In this respect, many software activities appear like a cottage industry and the analysis of the training need was relatively clear.

This paper describes the training methodology adopted and how it fitted in to an overall strategy to improve the level of implementation of the SQMS. The research for this paper has provided an opportunity to compare the methodology used against industry best practice evaluation techniques and to consider modern learning concepts to further improve the learning experience provided. This is conveyed in the literature review. A key theme within this paper is the demonstration of how the training and subsequent learning is applied by delegates in their working practices and how the new knowledge and skills improve the efficiency of work activity and how this leads to a cost-benefit. The costs and benefits are described under a number of headings. The difficulties associated with demonstrating a true return-on-investment are conveyed with a few case studies on how this was achieved. The paper concludes with a summary of the total savings achieved with a brief comparison with other methods.

Literature review

Training programmes are an essential feature of organisational life (Sandi and Robertson, 1996). Training initiatives are widely acknowledged to be a salient feature of the competitive organisation's corporate strategy and, in times of great change, learning is the key skill (Tennant et al., 2002). Employees, managers and organisations rely on training as a solution to enable issues to be resolved, yet only 35 per cent of UK companies have measured the effectiveness of their training and development programmes (Hale, 2003). To value and reward learning, an organisation must have a method of determining performance in learning and gaining knowledge. A method to assess the value of learning is to measure the performance of employees in creating items of business value from the learning (Morey and Frangioso, 1998). For training to be effective it must have specific

objectives and outcomes, which directly lead to business benefit (Morey and Frangioso, 1998). It is surprising that despite heavy investment in training, organisations fail to evaluate adequately the value or success of their training programmes and many that do, do so inadequately (Hale, 2003).

A method to assess the value or impact of learning is to compare the activities and outputs against a recognised industry best practice model. One such model is the Kirkpatrick training evaluation model (Turner, 2006). This simple yet effective evaluation system presents four levels of attainment as can be seen in Table 1. At Level 1 success of the training is assessed in terms of satisfaction and planned action. Level 2 requires the measurement of improved skills or knowledge. Level 3 captures the application of skills and knowledge “back-on-the-job”. Level 4 measures the resulting changes or improvements in the business from this applied knowledge.

Table 1, Kirkpatrick’s training evaluation frameworks (Kirkpatrick, 1998)

Level No	Level Title	Description
1	Reaction, satisfaction & planned action	Measures participants’ reaction to and satisfaction with the learning experience and captures planned actions.
2	Learning	Measures the learning and improvement in knowledge and skills.
3	Application & implementation	Measures changes in on-the-job behaviour and progress with planned actions.
4	Business impact	Measures the changes in business impact. Has the training helped towards the operational objectives?

Phillips (2002) extended the Kirkpatrick model to include the concept of return-on-investment (ROI). Here the business impacts are converted to a monetary value and compared to the cost of investment in the training and all other contributory factors. Phillips (2002) shows that the most common form of training evaluation consists of trainee reaction. He warns that delegates “stating they are satisfied in attending a course” does not necessarily mean that their new knowledge or skills will be applied. A criticism of Kirkpatrick’s model is that there is no initial evaluation to put the training into context, or identify the business need, and although there is the potential to realise that resources have been wasted, this comes later in the process. Another model to consider is context, inputs, reactions, and outcomes (CIRO) (Morey and Frangioso, 1998). This deals, in part, with Kirkpatrick’s weaknesses, but lacks the assessment of business impact. Turner (Turner, 2006) describes five organisational levels where learning outcomes are achieved which introduce personal motivation as a drive for learning. The five levels conveyed are, business, organisation, team, professional and personal. A survey of US companies evaluating training programmes found 13 per cent did not carry-out evaluations, 52 per cent used

satisfaction as the key evaluation technique, 3 per cent used skill acquisition, 17 per cent assess impact on the job, and 13 per cent gathered data on changes in organisational performance (Tennant et al., 2002). Another benefit gained by evaluation training programmes is that they can be used as a diagnostic technique to improve the training or they can help decide on alternative training (Tennant et al., 2002).

The effectiveness of the training itself can be improved by applying modern thinking on learning styles and treating course delegates as customers with individual needs. Dwyer (2001) proposes a new training model that is learner centred. It extends the usual context-centred model where the main focus is on the cognitive environment, to embrace emotional, physical and social environments as well. His research on how the brain works concludes that the ideal learning environment has the qualities of:

- being emotionally safe;
- being free from intimidation and rejection;
- being high in acceptable challenge;
- having active participation; and
- being a place where learners can experience a relaxed alertness.

Food plays a part as a nutritious breakfast containing high levels of protein will provide alertness during mornings and then a balanced diet containing carbohydrates will maintain alertness throughout the day. Caffeine drinks act as diuretics and deplete the body of water the brain needs to learn (Dwyer, 2001). Course design usually considers the need for more active learning in the afternoon than the morning, but consideration should be given on the fluctuations of neurotransmitters, which occur every 90 minutes (Dwyer, 2001). New concepts should be introduced in the first 20 minutes followed by ten minutes of “downtime” where learners process the new information and are able to make meaning of the new information. The next ten minutes should be used to further elaborate, reinforce, and summarise (Dwyer, 2001). Kolb’s ideas (1984) on learning styles have been introducing new ways of teaching which embrace the learner-centred focus. A questionnaire helps to identify whether someone is an Activist, Reflector, Theorist, or Pragmatist. Knowing your learning style can help participation in more suitable learning opportunities. Kolb (1984) also describes a concept, where it is believed that experience plays a central role in the learning process. He proposes a four-stage cycle for learning:

- (1) The concrete experience.
- (2) Observations and reflections.
- (3) Formation of abstract concepts and generalisations.
- (4) Testing implications of concepts in new situations.

The ultimate aim of the AWE software engineering training was to improve the application of the software quality management system. To evaluate the success of this training in terms of value, the financial benefits should be considered. It is therefore useful to review the availability of best practice cost models.

Most financial models for software systems have been produced for estimating the cost of software development, such as COCOMO (constructive cost model) (Boehm et al., 2000). At the “basic” form of this model, an estimate of the number of man-months that a particular development project will take is presented from an assessment of the number of lines of code likely to be produced. At the “intermediate” level, the model uses a range of development processes, system and people attributes to refine cost and schedule estimates.

In considering the cost-benefits and value of software quality or engineering practices the effectiveness of reviews, particularly software inspections (Fagan, 1976) are often cited. An example would be that they cut testing costs by 50-80 per cent or that they remove 80-95 per cent of faults at each development stage (Freeman and Weinberg, 1991). The importance of design information in a maintenance regime, is given by Bennett et al. (2000) in that with a complex system, 50 to 95 per cent of the cost needed to make a change can be taken to understand the program. Although this is not a detailed review of all cost models associated with software development practices, it does leave a feeling that there is “No silver bullet” (Brooks, 1987).

Training methodology

The philosophy for the required training identified at AWE was to provide a good introduction to the principles of software engineering and link this into how to implement the company software procedures. A partnership was established with a trainer provider that not only had a ready-made software engineering course, but also had experience of implementing the methods and techniques from a diverse range of software systems similar to AWE. The first author of this paper contributed to the training by providing the links to the Company Software Procedures. A feature of this part of the training was to not only provide examples of good AWE practice, but also identify some situations where people had misinterpreted the requirements in a manner that was slightly humorous. This helped give the course a lighter feel, as software engineering can otherwise be a dry subject.

Prior to attending the course, delegates were encouraged to undertake a self-assessment. This consisted of answering questions contained in a spreadsheet, on what actions required by the software quality management system had and had not been implemented. Self-assessment was particularly useful in support of the training as not only did it baseline the current level of implementation from which to track subsequent improvement and value, but it also helped gain an insight as to the level of understanding of software engineering or quality processes. This helped tailor the

training to specific individual training needs and present examples that were more relevant to attendees.

From the onset it was felt important to ensure attendees felt the course was of value. This was achieved in part through the monitoring of course evaluations but also by informal comments requested at the end of each course tutorial. The evaluations consisted of a comprehensive set of questions on course objectives, value, applicability, joining instructions, food, etc., to be answered on a six point scale ranging from 1 for excellent to 6 for very poor. This proved a good system and fully met the “reaction” element of Kirkpatrick’s evaluation framework at level 1, described in Table 2.

Table 2, The achievements of the AWE plc SQMS training against the evaluation frameworks of Kirkpatrick (1998) and Phillips (2002)

Evaluation Level	Training Objective	Achievement Action
1	Reaction/Satisfaction	Very good course evaluation results and encouraging comments gathered after each tutorial or exercise
	Planned Action	Increased percentage implementation of the SQMS monitored from follow-up self-assessments. However, not achieved or considered by delegates not participating in self-assessment
2	Learning	Monitoring of correct actions from exercise and case study. Course test results.
3	Behaviour/Application	Increased percentage implementation monitored from follow-up self-assessments and discovered on internal and external audits.
4	Results/Business impact	Measuring the efficiency of applied learning. Gathering data on business benefits.
5	Return-on-Investment ROI	Calculation of cost-benefit from conducting training internally. Development of cost models to facilitate gathering of data on systems demonstrating ROI.

After a number of courses it became clear that the AWE specific elements were considered the most valuable, so the course was altered to become a completely bespoke course, tailor-made to the company software system. Nearly all tutorials were geared to further implementing the SQMS requirements. Attendees would document their own processes, collate inventories and assign risk categories and software product baselines. They also learned to understand system measures and

metrics and apply their own. The courses were not made compulsory but the skills provided by the course were documented in a competency framework within the company software procedures.

Measuring the costs and benefit of training

Changes to an internal course

The change to the bespoke internal course provided an immediate cost-benefit. The cost incurred from the partnership with the training provider, which amounted to £600 per delegate was no longer needed as the course was delivered entirely by the first author of this paper. The course provided an introduction to software engineering principles which, to obtain this from an external company would cost in the region of £1,500 per person for a four day course. The same amount and type of information and some exercises were put into three days of the AWE course, as there was an element of “padding” in the external training. Part of the feasibility to change to a bespoke course included a review of course content from a range of training providers. The cost-benefit of providing an internal course is presented in Table 3.

Table 3, Training Course Cost-Benefit for an Internal Course

Year	2000	2001	2002	2003	2004	2005
Number of attendees	14*	23*	n/a	30	30	33
Total cost savings	£12600	£20700	n/a	£45000	£45000	£49500

* £900 saving per delegate, due to cost incurred from having the training partnership for the years 2000 and 2001. The savings for 2003 onwards are calculated as £1500.

Table 3 provides a powerful example of the benefits in providing internally developed and managed training courses in comparison with costs associated with attending external courses. The nine purely internal courses were run in the three years, 2003 to 2005, so this took 27 trainers’ days plus a total of 18 days for administration and course preparation. Much of the preparation was associated with case study material and the self-assessment that was sent out prior to the course. The results demonstrate very effective resourcing as the cost of the time to provide the training was approximately £9,000 and therefore the total cost saving of £130,500.

Time reduction to produce a management plan

Follow-up assessments were conducted to monitor how much of the SQMS was being applied. This would appear as an increase in percentage in the self-assessment and would be in line with Kirkpatrick’s model at level 3. As any improvement in the implementation of the SQMS can be directly attributed to the

training provided, the benefits of the improved implementation of the SQMS can also be considered to be the benefits of the training when calculating the return-on-investment.

The first step to calculate the monetary value or cost-benefit associated with the training was to develop an efficiency model. The main beneficial impact of the training would be a significant reduction in the time some of the actions required by the SQMS would take to be completed. Taking the production of a software management and control plan as an example, delegates on the training case study syndicate would develop the main elements of a plan and have it peer reviewed in one and a half hours.

The improved skills and knowledge acquired from this experience was found to significantly reduce the amount of time delegates took to produce a management plan in the future, giving a reduction of the order of 30 per cent. This concept is modelled in Figure 1.

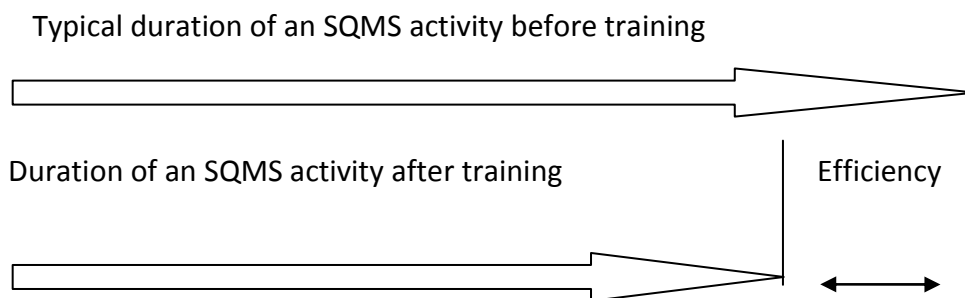


Figure 1, AWE plc training efficiency model

Cost reduction originated by times savings

This increased efficiency could be modelled with some degree of accuracy as a reasonable amount of data was available from corrective action from audits to calculate an average value (Elliott et al., 2005). This concept was to develop a complete model for all the other learning activities that could take place from the training course to produce a total efficiency value. Further examples of efficiency savings include the reduced time for developing baselines, capturing requirements in the company template, assessing risks, deciding an appropriate life cycle etc. It is estimated that a total efficiency value of £15,000 could be realised if all SQMS activities were improved. For each reassessed system, the percentage improvement from before and after training was assigned against that percentage of the total efficiency figure to document the cost benefit from the training as can be seen in Table 4.

Adoption is the number of attendees completing the self-assessment divided by the total number of attendees. Example efficiency saving for 2001, 19 per cent of total

possible saving of £1,500, multiplied by the number of systems assessed, 16, =£45,600.

Table 4, Training Course Efficiency Saving

Year	2000	2001	2002	2003	2004	2005
Number of attendees	14	23	n/a	30	30	33
Attendees completing self-assessment	n/a	16	n/a	20	17	16
Adoption %	n/a	70%	n/a	67%	56%	48%
Average % improvement	n/a	19%	n/a	23%	15%	16%
Applied equivalent efficiency savings	n/a	£2850	n/a	£3450	£2250	£2400
Total efficiency saving	n/a	£45600	n/a	£69000	£38250	£38400

Cost of overrun for an average IT project

The next step for calculating the total return-on-investment (ROI) was to calculate the costs for the time on the training course for both the delegates and tutor, and the costs for other resources used in the training. Two further issues were brought into the cost-benefit and ROI equations. First, the cost of applying or implementing the SQMS itself and the benefit the introduction that the new software system would have on, say, resource utilisation. A literature review on cost models was conducted to see if there were models available. The conclusion was there is really no generic model that can be used, although a number of cost statements exist on the various software engineering processes. These can be divided into two main areas, costing associated with new projects or costs associated with maintaining existing systems. It was decided to develop generic models that had some credibility for these two areas to facilitate discussions on cost-benefit. The first model was based on the Standish factors for successful software or information technology project (Standish Group International, 2001). The factors were translated in the various software project management and software engineering processes. These were calculated using the cost of project overrun for an average IT project of 178 per cent (Standish Group International, 2001).

The concept for the model is that if none of the processes are implemented the total overrun of 178 per cent would be applied. If all processes were implemented the project would be brought in on time to the allocated budget without any overrun. This is presented in Figure 2. A similar concept was applied for maintenance activities, in that each change is considered a mini project and the same philosophy applied, that if all best practice software maintenance processes were applied no overrun would be incurred. If none of the practices were applied, the 178 per cent overrun cost would occur, as depicted by Figure 3. These concepts were applied as a multiplication factor for each fulltime equivalent on the project. The idea was that

these costs would help engage practitioners in discussions on efficiency and cost-benefit for their working practices (see Table 5).

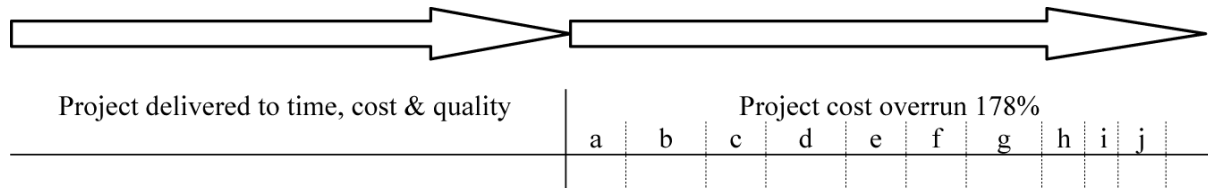


Figure 2, Assignment of percentage cost overrun to Standish Success Factors (Standish, 2001). (see Table 5 for key)

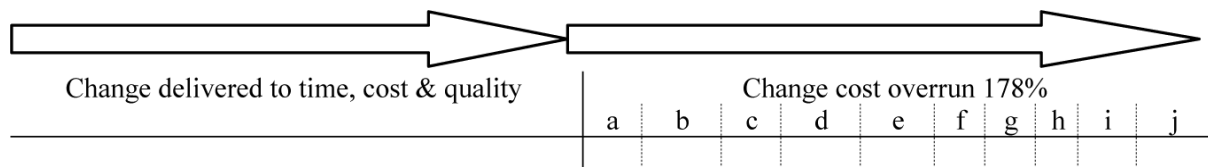


Figure 3, Assignment of percentage cost overrun to typical change activity (see Table 5 for key)

Table 5. Key to Figures 2 and 3, Standish Project Delivery Success Factors and best practice maintenance processes

Initial Project Delivery			Maintenance (a change project)		
Item	Standish Success Factors	%	Item	Typical SQMS Process	%
a	User Involvement	15.9%	a	General Management	5%
b	Executive Support	13.9%	b	Staff Effectiveness	15%
c	Clear Statement of requirements	13.0%	c	Requirements Management	5%
d	Proper Planning	9.6%	d	Design Techniques	20%
e	Realistic Expectations	8.2%	e	Programming Best Practice	10%
f	Smaller project milestones	7.7%	f	Testing Techniques	10%
g	Competent Staff	7.2%	g	Conducting Reviews	5%
h	Ownership	5.3%	h	Process Measurement	5%
i	Clear vision and objectives	2.9%	i	Configuration Management	15%
j	Hard-working, focused staff	2.4%	j	Effective Use & Efficiency	10%
k	All Other Practices	13.9%			

A validation exercise was conducted to see how accurate the efficiency model was against actual examples. Interviews took place which highlighted the problem that people were either not aware how long activities took or they were reticent to convey that information. A typical scenario would be that a person would say that to produce

a Software Management and Control Plan would take a week in total duration, yet the actual duration could be many months. Clearly other tasks would be attended to, however in some cases this would be the highest priority task. Further probing and detailed investigations found that the average or typical values in the model were reasonably accurate.

Project benefits generally consist of the quantitative value, payback or interest that is returned for an investment. The costs for a project refer to the expenses, expenditures, and capital outlay necessary to apply the project, which will hopefully result in some benefit. The equations used to calculate the return-on-investment start with a benefit to cost ratio.

$$\text{Benefit to cost ratio} = \text{project benefits} / \text{project cost}$$

The return-on-investment percentage is similar to the cost-benefit ratio except that the costs of the project are subtracted from the benefits before dividing by the costs.

$$\text{Return-on-investment} = (\text{project benefits} - \text{project cost}) / \text{project cost}$$

Another way the implementation of a SQMS can provide a benefit is through staff turnover. Documents are often written and only used at that time of production or review. However, additional value can be realised in the event of staff recruitment, as the length of time needed to learn, understand and become proficient with a system would be reduced if comprehensive documentation were available. Costs could be compared in the following situations:

- How long would it take to learn what the software did without a requirements specification?
- How long would it take to understand the software structure without design information?
- How long would it take to replicate testing arrangements without a test plan, procedure, and specification, and revalidate test data?
- How long would it take to understand what the software system consisted of without a baseline listing?

Product management system

A case study to be considered for a return-on-investment was a product management system. There were two aspects that were considered for calculating value and return-on-investment. First, from the efficiency of having the software system developed and then the benefit of producing the required documentation in the situation of staff-turnover when a new recruit had to learn what the system consisted of, and to become proficient in its management and use.

This system started life as a complex Excel spreadsheet that contained the entries of the components of a system. It was used to calculate the production and maintenance schedule for in-service items. The spreadsheet was used to manually

calculate a range of scenarios for optimum component production and/or maintenance. These scenarios could take many weeks to complete. It was decided to automate this process with a software system as part of a development contract. The developers of this Excel spreadsheet system had not applied any formal SQMS requirements and as a consequence a cost of £45,000 was incurred to produce a requirements specification in a total project cost of £215k. To manage a new contract, the system owners attended the in-house software course. On completion of this contract, the new software system provided scenario calculations that could take less than one hour to compute. Calculating the number of scenarios normally undertaken each year and the time saving gain from the new system it still took three years to provide a break-even point and a return-on-investment for its procurement. However, the responsiveness of the calculations meant that optimum maintenance and production schedules could be developed in the same responsive manner. The benefit associated with the new faster and more accurate scheduling of maintenance and production activities is very significant as this is a high resource area.

This case study also provided a means to test the use of staff turnover for calculating the benefit from training. It was calculated that the time taken to learn a system with good documentation was estimated to be approximately two-thirds of the time needed to learn about a system without any documentation.

Use of scientific analysis software

Another case study for the benefits of applied learning was in the use of some scientific analysis software. The development of a local SQMS in response to the knowledge and skills learned on the in-house course meant that the creation and review of important quality documents, such as the project management plan, took four weeks. In deciding the development approach or life cycle as part of the project management plan, it was noted that the software evolution was responsive to each customer request in turn. Each request could take four weeks to develop that customer solution. As a direct result of the training on appropriate life-cycle selection and implementation of quality review procedures to create a project management plan, the project managers sought resource to facilitate a demonstration of current software capability to all prospective customers. The conclusion of this exercise was that requirements were requested in one well-managed meeting that reduced the development burden considerably. It was estimated that 12 weeks were saved in development time. So the four-week documentation schedule and decision making captured in the management plan saved eight weeks from the schedule.

In both the case studies described, the benefits of using best-practice software engineering techniques and in implementing the SQMS can be directly attributed to the training provided. Without this training there is no reason to suggest that the project managers would have changed their approach and so they would continue as they had before. The self-assessment findings that the SQMS was poorly

implemented before the training, indicating that the improved methods used after training were as a result of what had been learned on the course.

Training course results

The training course evaluations shown in Table 6 have produced excellent results. The core issues considered more important, such as meeting course objectives and trainer knowledge, scored particularly well, though other aspects, such as joining instructions, food and building environment issues scored less well. The initial course of 2000 was marketed as a pilot course requiring endorsement to continue by the delegates. The novelty of this pilot may account for the very good initial score of an average 1.78. The course tests were introduced because some of the case study work was not being performed accurately every time. The tests were made challenging to ascertain the true level of the learning taking place. It also gave feedback that allowed the trainer to modify or increase emphasis on learning points. Predictably the results improved each year as training course content was modified to take-on the extra emphasis.

Table 6, Training Course Results

Year	2000	2001	2002	2003	2004	2005
Average course evaluation	1.78	2.13	n/a	2.01	1.9	1.77
Average test result	n/a	n/a	n/a	64.3%	71.4%	77.7%

Key: A course evaluation score of 1 is excellent, 2 is very good, 3 is good, 4 is average, 5 is poor, 6 is very poor

The use of self-assessment techniques to assess applied learning and evaluate associated efficiencies is also very powerful and cost effective. It does emphasise the need to have self-assessment as a pre-requisite for the training in order to provide a baseline from which to measure any improvement resulting from the training. This then enables the measurement of the applied learning back on the job, which is the true objective of any course.

Conclusions

The methodology presented and associated results show that the training provided and learning experience created has been successful in demonstrating the benefits that training can have in the workplace. This success can be quantified by the cost-benefits and the financial return-on-investment results. The total cost savings from Table 5 for running an internal course totalled £172,800. The total efficiency savings from the applied learning calculated from Table 6 were £191,250.

The difficulties associated with demonstrating cost-benefits have also been conveyed, particularly the accuracy of recording the duration of activities. The human perception factors and the reticence to convey efficiencies can be significant hurdles to overcome. The cost overrun model can be utilised to convince project and system managers of the need to apply best practice. The benefits of applying the correct development life cycle have been highlighted by the reduction of eight weeks for a small scientific analysis software project.

Demonstrating a cost-benefit is a powerful means by which any quality manager can influence working practices. Overcoming the perception that quality assurance can be a bureaucratic overhead is a significant challenge, so any methodology to demonstrate its value can be highly influential. The reduction of a third in learning time for the product software system provides justification for documentation as a knowledge capture benefit.

This research, in particular the use of reasonably accurate task durations, compares favourably with the work of Phillips (2002) and Rico (2002). Phillips utilises surveys to gather results for return-on-investment, which are prone to personal bias, and the models developed by Rico, which although excellent as a reference model, may not be accurate in all cases.

Recommendations

There are a number of improvements that have been identified to further enhance the training and learning experience currently provided. To further assess the learning actually gained from the training and, perhaps more importantly, identify delegates with less knowledge, a test could be conducted at the start of the course. This could be a simple questionnaire to ascertain delegate's confidence on relevant topics. Also, some of the effective learning methods presented in the literature review could be utilised for course improvement.

If the use of self-assessment as a pre-course requisite proves troublesome, the development of action plans, in-line with Kirkpatrick's level 1, "planned action", could be incorporated into the course.

More return-on-investment case studies could be used to build-up expertise to improve the training courses, and to act as a powerful and influential marketing tool to promote the training courses and the software quality management system by demonstrating their value and benefits.

Providing demonstrable return-on-investment for the training has increased the understanding of the benefits of training for management which has enabled better planning and budgeting for the courses. The methods used have also enabled the courses to be continually improved and the benefits of the software quality management system to be demonstrated. The authors recommend, therefore, that

other companies employ the methods described in this paper to obtain similar benefits to those experienced by AWE.

References

- Bennett, K.H., Munro, M., Knight, C. and Xu, J. (2000), "Informatics centres of excellence; research institute for software evolution", *IEE Computing and Control Engineering Journal*, Vol. 11 No. 4, pp. 179-86.
- Boehm, B., Abts, C., Brown, A.W., Chulani, S., Clark, R.K., Horowitz, E., Madachy, R., Reifer, D.J. and Steece, B. (2000), *Software Cost Estimation with COCOMO II*, Prentice-Hall, Englewood Cliffs, NJ.
- Brooks, F.P. (1987), "No silver bullet: essence and accidents of software engineering", *Computer*, Vol. 20 4, April, pp. 10-19.
- Dwyer, B. (2001), "Successful training strategies for the twenty-first century", *The International Journal of Educational Management*, Vol. 15 No. 9, pp. 312-8.
- Elliott, M., Dawson, R.J. and Edwards, J. (2005), "Approaches to internal auditing – a software quality assurance case study", in Bennets, P., Ross, M. and Staples, G. (Eds), *Proceedings of Software Quality Management XIII: Current Issues in Software Quality*, BCS, SQM, Cheltenham, pp. 43-55, March.
- Elliott, M., Dawson, R.J. and Edwards, J. (2006), "An analysis of software quality management at AWE plc", in Bennets, P., Ross, M. and Staples, G. (Eds), *Proceedings of Software Quality Management XIV*, BCS, SQM, Cheltenham, pp. 17-30, April.
- Fagan, M.E. (1976), "Design and code inspections to reduce errors in program development", *IBM Systems Journal*, Vol. 12, pp. 741-4.
- Freeman, D.P. and Weinberg, G.M. (1991), *Handbook of Walkthroughs, Inspection and Technical Reviews*, Dorset House Publishing, New York, NY.
- Hale, R. (2003), "How training can add real value to business: part 2", *Industrial and Commercial Training*, Vol. 35 No. 2, pp. 49-52.
- Kirkpatrick, D.L. (1998), *Evaluating Training Programs*, Berrett-Koehler Publishers, San Francisco, CA.
- Kolb, D.A. (1984), *Experimental Learning: Experience as the Source of Learning and Development*, Prentice-Hall, Englewood Cliffs, NJ.
- Morey, D. and Frangioso, T. (1998), "Aligning an organization for learning: the six principles of effective learning", *Journal of Knowledge Management*, Vol. 1 No. 4, pp. 308-14.

- Phillips, J.J. (2002), *Return on Investment in Training and Performance Improvement Programs*, 2nd ed., Butterworth-Heinemann, Woburn, MA.
- Rico, D.F. (2002), "Software process improvement: modelling return-on-investment (ROI)", paper presented at Software Engineering Process Group Conference (SEPG), Software Engineering Institute (SEI), Phoenix, AZ.
- Sandi, M. and Robertson, I.T. (1996), "What should training evaluations evaluate?", *Journal of European Industrial Training*, Vol. 20 No. 9, pp. 14-20.
- Standish Group International (2001), *Extreme Chaos*, Standish Group web site, www.standishgroup.com (accessed November 2005)
- Tennant, C., Boonkrong, M. and Roberts, P.A.B. (2002), "The design of a training programme measurement model", *Journal of European Industrial Training*, Vol. 26 No. 5, pp. 230-40.
- Turner, E. (2006), "Learning that lasts", *Industrial and Commercial Training*, Vol. 38 No. 3, pp. 137-42.