Engineering Development
From Action Learning to Test & Evaluation

Errol Lawson
Stephen Cook
Systems Engineering & Evaluation Centre
University of South Australia
F2-24 Mawson Lakes Campus,
Mawson Lakes, 5095
(08) 8302 3369
Email: Errol.Lawson@unisa.edu.au

ABSTRACT

Eberhardt Rechtin (1991) cites the following two heuristics "Pause and Reflect" and "Choose-Watch-Choose". In another discipline, that of Education, David Kolb (1984) describes the Experiential Learning Cycle as comprising four stages as "Experience-Reflection-Conceptualisation-Planning". Rechtin's two heuristics can be combined to reproduce the same experiential learning cycle described by Kolb. This models the process by which individuals, teams and organisations learn from and understand their experiences and subsequently modify their behaviour, if they so choose. Cyclic learning portrays a process that is not linear and deterministic but rather is iterative. In engineering, this cycle of learning can be observed in the development of novel and complex products and projects. The paper discusses the relationship between Engineering Development and the Experiential Learning cycle and situates Test and Evaluation in both. It concludes that the place of Test and Evaluation in the engineering development process can be understood in terms of experiential learning.

Introduction

This paper discusses the engineering development cycle as an example of learning from experience and the place of Test and Evaluation in that learning process. An objective is to decouple Test and Evaluation from the legislative process mandated by the US Congress as a mechanism for control over Defence projects and restore it to its integral place in the engineering development cycle.

The paper is prompted by the observation that large-scale acquisition organisations, such as that of the Australian Department of Defence may have overlooked the human activity aspects of engineering development in their doctrinaire application of the Systems Engineering process. Two features of the acquisition process which are driving this oversight are:

- The rush to reduce the time lapse between first recognition of an operational need and the fielding of a system; and
- The false belief that novel and complex projects can be got right first time.

The first is legitimate, but requires care in its application. The second is a hangover from the misguided belief of the seventies that an exhaustive analysis of all the relevant information would yield a solution that would be correct for all time.

The main theme of this paper is that engineering development is a learning process. The learning is most often unnoticed, but a moment's thought will show that when a group brings a new system into existence, there is new knowledge created which has to be given a consistent meaning across the project group. That is learning. Much of knowledge is tacit and test and evaluation is accorded a natural place in the engineering development process not only as the proof-of-knowledge stage in the process but also as the occasion for the exposure of the tacit knowledge.

Engineering Development

The engineering tradition, into which the principal author was inducted in 1960, predated the formality of Systems Engineering and recognised the following phases of development:

1. Experimental Model: a single unit built to ensure that the technologies being considered are under control and that the particular combination of technologies does not generate some unpredicted and unwelcome emergent properties.
2. Prototype Model: a single unit built to demonstrate function and approximate form to the prospective clients to ensure that they know and understand the system being developed on their behalf and believe that it will serve some requirement of theirs.
3. Pre-production: a limited number built to the form, fit and function of the final system using production documentation to ensure that the documentation is correct and suitable for manufacture.
4. Production: numbers as ordered built to production documentation using production tooling.
5. Client Evaluation: to ensure that the clients got what they expected and what they needed and that they can operate and maintain the system.

Experimental Models

An important aspect of this engineering paradigm was that it addressed the critical issues separately. It took one risk at a time. Where there was new technology or a new mix of technologies, experimental models were built to prove the ability of the designers not only to predict and control the desired performance of the system, but also to ensure that any unwanted performance had not emerged from the new technology or combination of technology.

There was no exclusion of representatives of prospective users from the evaluation of the experimental model. They were welcome so that they could begin their own program of learning the potential of the technology to meet some requirement of theirs.

The objectives of an experimental model were to:

- Minimise cost
- Reduce technical risk
- Establish a technical baseline for the next phase
- Assist prospective users begin their learning process.

The elimination of technical risk and establishment of a technical baseline at minimal cost were the primary objectives of constructing this model. But in addition, the prospective users were involved and were given the opportunity to begin to appreciate what the technology could do for them. Both the development engineers and the prospective users learnt from the experimental model.

Good practice requires minimal documentation, that which is sufficient to enable a recreation of the model, as well as reports describing the testing and the results, including any retesting to solve unanticipated problems. One of the common mistakes was to lose configuration control in the belief that it was not necessary for a design that would never be built again.

Prototype Models

The investment in a prototype model is an indication that a potential user was sufficiently interested to fund the next stage of the engineering development. Usually only one was built, together with enough spares to enable recovery from a serious failure. This is the stage when the users evaluate the operational features that have been made feasible by the technology. For that reason alone, the developers had to resist the temptation to insert a bit more technology.

The prototype needed to be closely representative in form, fit, and function, to a final notional system and robust enough to withstand realistic field use to enable the user to evaluate the operational characteristics in as near as possible to the envisaged situation.

The mechanical components of prototypes may be hand made, but, as with the experimental model, documentation needed to be maintained under configuration control otherwise the value of a prototype could be lost if any aspect of the system which a user has trialled and reported on was not recorded properly. The growing trend for the functionality of systems to be determined by software has exacerbated the need for good configuration management. It is highly desirable for production engineers to begin their learning phase during the prototype phase but this is not always possible because production contracts are frequently awarded after this phase.

The objectives of a prototype model are to:

- Demonstrate form, fit and function to the potential users.
- Maintain the technical baseline of the experimental model.
- Receive feedback from the users.
- Establish an engineering baseline.
- Complete the users' learning phase.
- Commence the production engineers' learning phase.

Feedback from the users comes in a statement of user requirement, and provided the testing had taken the users' learning to completion, the users' requirements should be stable by the end of this phase. At the end of testing and reporting on the prototype model, the technology and engineering baselines and the user requirements baseline should be established, documented, agreed, under configuration control and stable.

The three baselines have evolved sequentially but with iterations throughout the stages of experimental and prototype models. Stable baselines can only be attained when the engineers know what the users need and the users know what the engineers can provide.

Pre-production Models

Design for production has to include design for function in the operational environment, operability and maintainability and requires significant investment. If production numbers warrant, there is an accompanying investment in production tooling and if the system is complex, there is a further investment in test equipment. It, therefore, is very important that the production tooling is proven before production commences.

The objective of the pre-production phase is to prove the tooling. Design engineers and user representatives are no longer central to the action. However, the production engineers should be given the opportunity to learn what it is they have to build and how they are going to build it. They will have suggestions, which may ease their production problems, but could inadvertently change the design or the performance. For this reason, the designers and the users need to be easily accessible in order to answer questions and approve or disapprove change proposals.

At the completion of the pre-production phase, the production engineers will have completed their learning and the production tooling will have been proven. The maintainers should have been allowed to make their contribution during the design. The pre-production model gives them an opportunity to influence the design before it is committed to production.

If the new system is especially novel and complex, the pre-production units may be subject to extended use by the users. This is sometimes necessary but can be dangerous if the users' learning was not completed in the prototype phase. Modifications are very expensive at this stage and will almost certainly disrupt the delivery schedule. However, if the operational environment has changed to the extent that the performance requirements are no longer valid, the program has to be halted and restarted or cancelled. There is no point in producing a system that is of limited use to the users or no use at all.

Production Models

Given that the system has passed all the developmental testing and approval gates, production contracts can be awarded and deliveries commenced. Ideally, the design engineers and the production engineers have learnt all they need to know to deliver a system that meets the users' requirements. The users and the maintainers have had opportunities during the developmental phases to learn the implications of the new system. However, it is not until the system enters service that the final determination of its effectiveness can be made.

Risk Management

The development process outlined above has the disadvantage that its sequential nature can extend the time from technology opportunity to fielded system to the point where the requirements have been superseded or the technology has been surpassed. Nonetheless, the advantage of the sequential approach is that risk management is integral to the process. Each phase concentrates on one risk area only, effectively holding all other variables constant. The final stage of testing discovers if the uncertainties associated with that particular phase have been resolved.

Those responsible for the risk area in the next phase are introduced in order that they have commenced their own process of discovering the system in their idiom and are ready for their turn. But they wait their turn.

However, the phases can be run in parallel, as is implied by the term Concurrent Engineering. In Concurrent Engineering, the various communities, technology developers, design engineers, production engineers, users and maintainers, all have to learn their specific aspects of the system concurrently. There is a considerably elevated risk in concurrency. It requires very alert and experienced management and well practised conflict resolution techniques to sort out the tangle when two or more risks strike simultaneously.

In the Skunk Works (Rich and Janos, 1994), Lockheed managed concurrent engineering over many years by co-locating technology developers, design engineers, draftsmen, and production workers and facilities and maintaining close links to the user community.

More recently, Admiral Arthur Cebrowski (2001), Director, Office of Force Transformation, (US Navy) embraced operational prototypes and spiral development. When the order of the day is to bring on a Revolution in Military Affairs, several cycles of development and testing are required for the operational concepts and the system design to converge.

If we view the delivered operational system as an emergent property of technology, people, the military reward system, concepts of employment and the external environment, it is apparent that Cebrowski is engaged with not one spiral, but at least the five listed above. They are all dependent variables, which makes for a highly complex undertaking. A change to any one will affect the others. It is the nature of complexity that they are interdependent and whatever the process of development that is applied, it needs to accommodate new knowledge in every area and recognise the system consequences.

Experiential-Learning

David Kolb (1994) is one of the thinkers who describe the way people acquire knowledge as Experiential Learning or learning from experience. His idea that knowledge is constructed through a series of Experiential-Learning cycles resonates with the steps outlined in the engineering development stages. That is, there is a sequence of phases, and within each phase there is a sequence of discovery and learning, refer to Figure 1.

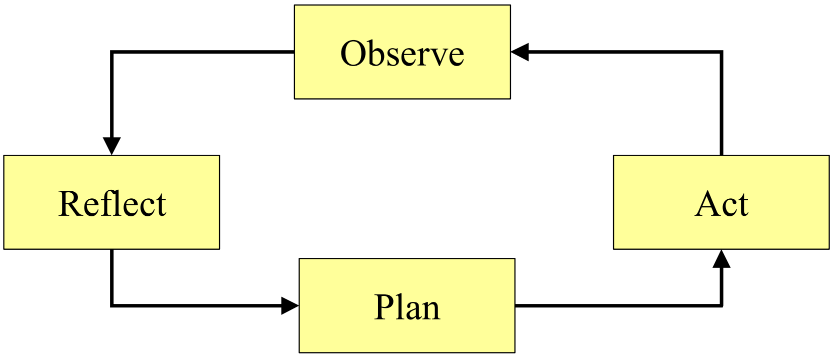

Figure 1 The Experiential-Learning Cycle

There are a number of variations in the actual detail of the experiential cycle. The OODA loop (Observe-Orient-Decide-Act) is one of these credited to a USAF fighter pilot. A "Decide" step in the cycle at figure 1 might add further veracity to the model. But it is a model and there are limits to detail. The four-stage model is useful in that it demonstrates that there is a very large constituent of learning – albeit unremarked – in the engineering development process.

The four-stage model is also useful in illustrating the different learning styles that people exhibit. Kolb's argument is, for example, that a person whose strengths are in reflection and planning would have a learning style that is different from someone whose strength is in acting.

It is important for team leaders to remember that people have different learning styles, and different teaching styles, as well. Much of the data transferred during an engineering project is in a standard format that almost certainly suits one learning style more than others.

Test & Evaluation

To this point the paper has traced the engineering development process and related that process to a series of learning cycles. Testing and evaluating is an integral part of the engineering process, not some separate stand-alone process. This integral characteristic of testing and evaluating is reinforced by reference to the experiential-learning cycle. Rechtin's heuristics "*Choose-watch-choose*" and "*Pause and reflect,*" are very appropriate. They can be applied within each of the five phases of the engineering development process, as well as to the entire process.

The heuristics apply to each phase and can be restated as: *Do what ever it is you have planned, take the time to observe the results and reflect on those results until you understand the implications before you decide on your next course of action.*

This is a practical definition of the engineering development process with test and evaluation as an integral component. The major objective of each phase is a baseline in technology, engineering, user requirements, or production. The particular details of testing and evaluating at each phase are determined by the objectives of that phase.

Communities of Practice

Within each phase during the several phases of development of a system, there are a number of identifiable groups who experience learning cycles concurrently but with strong interdependency. Wegner (1998) calls such groups communities of practice and it is possible to describe the engineering development process in terms the activities of and interactions between, communities of practice. A Community of Practice is characterised by Wenger as having the following features:

- Mutual engagement
- Joint enterprise
- A shared repertoire.

Several groups of people involved in the engineering development process described previously can be identified. They include the technologists, the design engineers, the test engineers, production designers, production engineers, the users, the maintainers, and the contract managers.

Each one of these groups shows the characteristic features given by Wenger. Mutual engagement means that the members work together cooperatively. Joint enterprise means that they share common goals, both for their community as well as for any tasking they may undertake. Shared repertoire means that they have an agreed way of describing and addressing problems and most probably a particular language and format for the communication of information and the normalisation of meaning.

Irrespective of the size of a project, or whether the organisation subscribes to a functional or a matrix structure, projects are conducted and progressed when communities of practice contribute their speciality to a larger heterogeneous project group, a meta-community of practice. The Lockheed Works was such a meta-community given permanent status by the company management because it delivered products that the USAF wanted and profit for the company. The support organisation for the Woomera Range from the Weapons Research Establishment (WRE) Salisbury was a meta-community that, because it was initially the sole task of the WRE, became the formal structure as well. The Collins Class Submarine contractor, the Australian Submarine Corporation (ASC), was a meta-community of a Swedish submarine builder, a US project management firm, an Australian electronics firm and a Government owned investment corporation, which had formed for the sole purpose of winning the submarine contract and which had no joint enterprise in view beyond that.

Whatever the project organisation, functional or matrix, and whether the joint enterprise is a one off or one of a series, the members of each specialist community have a dual role. One is to remain competent in their home community while the other is to participate in the meta-community of the project.

While it may be less deterministic than a model of engineering development comprising defined phases that are either sequential or concurrent, a model of a development project as a set of continuously interacting, cooperating communities of practice may be closer to what actually has to happen for a project to succeed.

Tacit Knowledge

For communities of practice to continuously interact and cooperate effectively there must be an ongoing interchange of information. This is well recognised and readily accomplished in the age of Information Technology. Or is it? There is no guarantee that the communities will have a shared repertoire. In fact most will have community specific ways of translating the circulated information into meaning and ranking its significance. That is a serious problem that can be overcome by any or all of leadership, project-wide colloquia, and proximity.

Polyani (1958) described an even more difficult problem, that of tacit knowledge. Tacit knowledge is knowledge that we have which we don’t know we have. Nevertheless, it plays a crucial role in determining how we interpret information, and how we decide what response to make to stimulus from the environment.

Nonaka and Takeuchi (1998) suggest that 80% of an organisation's knowledge of how to conduct its operations is tacit. We may wonder how they arrived at such a precise figure for a parameter that by definition is not measurable. However, we can accept that in the communities and meta-communities that comprise a project organisation, a large proportion of the knowledge that drives the development process is tacit. The implication is that traceability of decisions is an uncertain process.

It may be that the users can only be explicit on a small proportion of their requirement, or the engineers and technologists can only find partial explanations for their decisions. Perhaps herein lies the reason why Reviews and Audits to MIL-STD-1521B so often produced unsatisfactory results.

Discussion

So far we have described the engineering development process in terms of a structure of staged activities in which there is separate involvement of specialists and a progressive build up of the information needed to produce, maintain and operate a new system.

As an alternative, we describe the development process as the continuous interaction and cooperation of communities of practice. The common factor is that the specialists groups of the former description correspond to the communities of the latter.

The description in terms of a structure of activities does not adequately reveal the actual human activities and makes unrealistic assumptions such as the independence of variables, and that the variable can be dealt with separately, and once dealt with, can be frozen. The inflexibility of the structured approach is also mirrored in the fixed-price, milestone payments contractual approach that does not allow for iterations to accommodate new knowledge.

The view of a project as a meta-community of practice reveals the necessity for care in dissemination of information to ensure the same meanings are applied and maintained throughout the project, and for iterations in all areas as new information is gained.

Both the structured view and the human activity view have their strengths and weaknesses. They are both models of reality and as such will always lose detail in the search for generalisation. Used in combination, the two models provide better guidance to project managers than either in isolation.

Conclusion

Test and evaluation has been shown to be an integral part of the "observe and reflect" sections of the experiential learning cycle. An important consideration is that a large proportion of the information on which project development is based is tacit and may only be revealed during the test and evaluation stages of the project. For that reason test and evaluation assumes critical importance in achievement of project success.

This paper has used two models to describe the engineering development process. The two models are structured stages and communities of practice. It suggests that the two models can be applied in a complementary manner to increase the probability of project success. Test and evaluation has an integral place in both models as the way in which the experiential-learning cycles can be traversed and the learning consolidated in a constructivist sense.

However, the most critical finding of the paper is that tacit knowledge comprises a large proportion of the totality of a project and is embedded in all aspects of a project.

References

Cebrowski, A. (2001), *Special Briefing on Force Transformation,* November 2001. www.defenselink.mil/news/Nov2001.

Kolb, D. A. (1994), *Experiential Learning: Experience as the source of learning and development*, Englewood Cliffs, NJ: Prentice Hall.

Nonaka, I. & Takeuchi, H. (1995), *The Knowledge Creating Company, How Japanese Companies Create the Dynamics of Innovation,* New York: The Oxford University Press.

Polanyi, M. (1958) *Personal Knowledge*. London: Routledge & Kegan Paul.

Rechtin, E. (1991), *Systems Architecting-Creating and Building Complex Systems*, Prentice-Hall. ISBN 0 13 880345 5.

Rich, B. and Janos, L., (1994), *Skunk Works, A Personal Memoir of My Years at Lockheed*, Little, Brown and Company, ISBN 0-316-74330-5

Wenger, E. (1998) *Communities of Practice-Learning, Meaning and Identity.* Cambridge University Press ISBN 0 521 43017 8.