The use of cognitive causal mapping as an aid to professional reflection

Robert T.Hughes, Abdullah Al Shebab, Marian Eastwood
School of Computing, Mathematical and Information Sciences,
University of Brighton, Brighton, BN2 4GJ United Kingdom

Email r.t.hughes@bton.ac.uk

Our background is broadly in software engineering, typically but not exclusively within information systems domains. Our individual research interests have included software effort estimation, software project risk, and development process models which may reduce risk. Two of us are committed to teaching students at undergraduate and graduate levels. As busy academics, we like research challenges that have the potential to enrich our students' learning experiences, as well as, in the longer term, being of interest in the wider world.

Schön's work resonates with those teaching vocational subjects such as software engineering, but is also frustrating. The observation of professionals at work can offer some insights into practice, yet much remains concealed or unclear. As it says in one case study: *'…the Resident discerns in the Supervisor's performance a knowing-in-practice that he values, but is frustrated in his attempt to grasp it*'. Schon (1983). One of us (Hughes 1996) felt a similar frustration during an attempt to elicit how software developers produced estimates of development effort.

As teachers we would like in some way to capture practitioners' expertise so that it can be presented as 'professional knowledge' to our students. One recent emphasis in the UK has been on 'life-long learning, a term often difficult to interpret in practice, but generally held to be a 'good thing'. Our interpretation is that one aspect of this is that students ought to be taught to learn from their experience, as well as from pre-packages modules of learning material, and that this will require them to be able to reflect on their practical experiences.

To this end, as educators, we have recently focussed on the production of software prototypes by students as a vehicle for learning (Eastwood and Hughes, 2004). In industry, prototypes are long-established as a way of learning and reducing uncertainty in the context of a specific project (see, for example, Lichter et al. 1997). This type of 'industrial' learning needs to be focussed and fast. However, at the same time, practitioners are acquiring skills and knowledge that can be applied to later projects. This longer term learning is unlikely to be a concern of the project manager who has more immediate anxieties. One can therefore idealise a number of learning cycles of different lengths progressing concurrently.

Higher education ought to have more concern with the longer cycle learning. The use of prototyping in this context should allow the student to exercise some short cycle skills in software development and also, one hopes, learn some more abstract, long cycle, lessons about design and project management.

The use of a prototype to solve a particular local problem is a process that is very visible, not just to tutors in an academic environment who may wish to assess it, but to users in the 'real world'. A more difficult problem is to assess long cycle learning. A 'traditional' way has been to compel the student to complete a reflective report describing the way that the particular problem was solved. Our experience is, however, that the 'better' students will often respond by describing the process which they believe the tutors want to hear, normally about the slavish following of some standard methodology, rather than genuinely reflecting on their actual experience.

If one turns to the ‘real world’ of software development a striking feature is the prevalence of failed projects (see, for example Flowers, 1996). Part of the reason for this may be the innate difficulties of software, as identified by Brooks (1995), namely its relative invisibility, its complexity, the flexibility that makes it have to conform to the whims of clients, however illogical, and also make it subject to constant change. Despite this, one can still sometimes think that if, in this field, the ‘professionals’ are building up Schonian ‘repertoires’ based on past experience, then the process does not seem to be overwhelming successful. We suspect that one reasons for this prevalence of failure is that all types of large organisation are likely at some point to be subject to information systems implementation projects, and we cannot expect all managers in all organizations to turn themselves into project management professionals at a moment’s notice. We also suspect that there is often a lack of longer cycle organizational learning. In such environments we suspect that project failure is often caused by influential stakeholders having preconceptions about the factors at work that will influence the success or failure of the project, and then these preconceptions not being matched by reality.

Our proposal for ameliorating this mismatch between the preconceived and the actual is to attempt to make more explicit to stakeholders the nature of these preconceptions by employing collaborative cognitive causal mapping techniques. These are influenced by Kelly’s constructivist model of human motivation (Kelly, 1955). This sees the being in the world trying to predict and control the outcome of events by using past experience to identify those variables that are likely to dictate the future course of events.

These variables are perceived as dichotomous indicators that reflect the degree to which a concept has a value between two extremes, for example, large and small, cheap and expensive, fast and slow. These variables are represented as nodes in a network, and directed arcs between the nodes indicate that one variable can influence the value of another. For example, Figure 1 reflects the thinking that incomplete requirements may lead to the functionality of an application being increased to cover the missing requirement when found. This in turn increases the overall cost

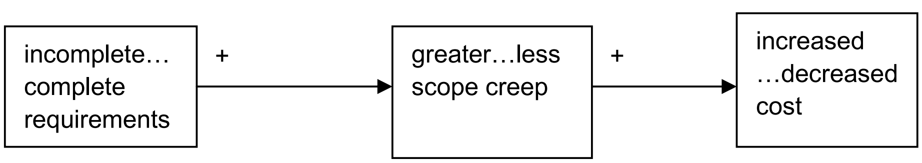

Figure 1 A simplistic causal model

What might happen is that in order to overcome this unwelcome chain of events, a prototype is planned, which it is hoped will help elicit a full set of requirements from the users - see Figure 2.

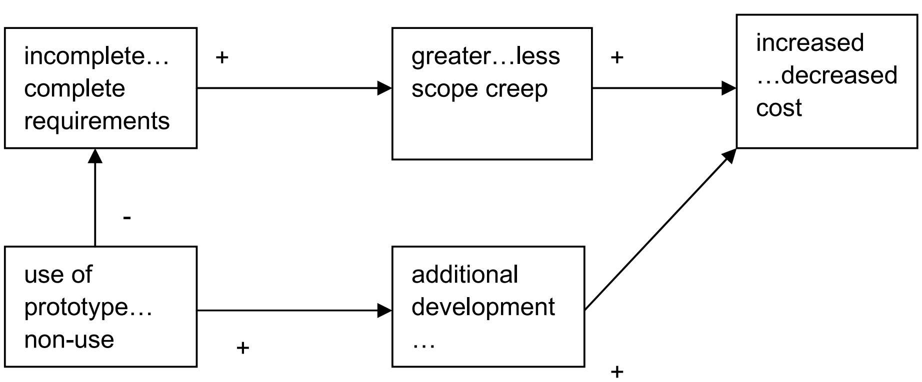

Figure 2 Adding the use of a prototype

The use of the prototype, while reducing the possibility of incomplete requirements, will also contribute to cost as it requires additional effort to create the prototype. Where there is a negative, inhibiting, link in a causal chain then, unless there is a compensating second negative link, the causal chain between two nodes will be a negative. Thus the causal chain between 'use of the prototype' via 'incomplete requirements' to 'increased…decreased cost' is a negative one, that is, the use of prototypes should decrease costs. However, the second causal chain between 'user of prototype' and 'increased…decreased cost' is a positive one: the prototype will add to costs. This is an example of an *unbalanced* causal map. Unbalanced causal maps indicate that some outcomes are uncertain, which indicates that there is an element of risk. Another diagnostic feature of causal maps is that they can detect feedback loops and also *unstable* systems where an increase in an external variable can cause some of the variables to grow in an uncontrolled manner. For example, a stream of changes, where there is no effective change control mechanism, might cause runaway development costs.

The production of causal maps can be done retrospectively, at the post-mortem stage of a development project, in order to diagnose the reasons for unsatisfactory and satisfactory outcomes of the project. It can also be used at the beginning of a project to help stakeholders identify potential difficulties and possible policies to reduce those difficulties. Clearly using the output from post mortems from previous projects to inform the risk analysis for future projects could facilitate learning, at both the level of the individual professional and at the organizational level.

Cognitive mapping approaches are well established - one early use (Axelrod, 1976) was to analyse how diplomats and government officials decided and applied policies, particularly in the field of foreign policy. In the field of project management and operational research, they have been used extensively - see, for example, Eden 1988, Eden et al. 1992, Brown 1992, Williams et al 1995.

Often cognitive maps are used as a preliminary stage to the creation of a quantitative systems dynamics model (see, for example, Abdel-Hamid and Madnick 1986, Williams et al 1995). The addition of quantitative data could, for example, resolve whether the use of a prototype in the fragment in Figure 2 would be worthwhile. Quantitative considerations can enrich the causal models by reflecting the strength of the influence of one variable upon another; they can also indicate the presence of time delays in the influence of one variable being felt by another. Senge (1990) has advocated the use of animated, quantitative models as a means of fostering systems thinking in organizations. The ability of such models to demonstrate the long term effects of policies is particularly stressed by such advocates. We are, however, rather cautious about attempting to enhance the modelling to take on quantitative aspects, despite our computer science backgrounds. Partly this is because we are aware of the pitfalls associated with measurement (Hughes 2000). It is also because we have a suspicion that showing that a computer model produces the results it was programmed to produce might demonstrate skilful programming rather than any underlying new knowledge.

We are currently using these techniques in research into project risk, by carrying out retrospective mapping of industrial projects, as an aid to the diagnosis of project short-comings. We are also planning to train our undergraduate and graduate students in the technique so that the can apply the approach both at the planning and reflective stages of their student projects as an aid to learning. Masters students in information systems and software engineering have, for example, to conduct a project for an outside client as the last element of their courses. We would like to require these students to produce cognitive maps as a way of justifying their planned approach to the project and also at the end of the project to support their reflection on the project.

References

Abdel-Hamid, T.K. and S.E.Madnick. 1986 Impact of schedule estimation on software project behavior. *IEEE Software* 34(4) 70-5

Axelrod, R. (ed.) 1976 *Structure of decision: The cognitive maps of political elites*. Princeton University Press, Princeton, NJ

Brooks, F.P. 1987. No silver bullet, essence and accidents of software engineering. *IEEE Computer* April 10-19

Brown, S. 1992. Cognitive mapping and repertory grids for qualitative survey research: some comparative observations. *Journal of Management Studies* 29(3) 287-307

Eastwood, M. and Hughes R.T. 2004. Software prototyping as experiential learning. SQM/Inspire conference. Canterbury, Kent. April

Eden, C. 1988 Cognitive *mapping European Journal of Operational Research* 36 1-13

Eden, C., F.Ackermann, and S.Cropper. 1992. The analysis of causal maps. Journal of Management Studies 29(3) 309-324

Flowers, S., 1996 *Software failure, management* failure. Wiley and Sons, Chichester

Hughes, R.T. 1996 Expert judgement as an estimating method *Information and Software Technology* 38(3) 67-75

Hughes, R.T. 2000. *Practical software measurement*. McGraw-Hill, Maidenhead

Kelly, G.A. 1955 *The psychology of human constructs. Volume 1.* W.W.Norton and Co. New York

Lichter et al 1997. Prototyping in industrial software projects - bridging the gap between theory and practice. *Proceedings of the 15th International Conference on Software Engineering* May 1997 221-229

Schon. D.A. 1983. *The reflective practitioner: how practitioners think in action*. Basic Books

Senge, P.M. 1990. *The fifth discipline*. Random House, London

Williams, T., C. Eden, F. Ackermann. and A.Tait. 1995 The effect of design changes and delays on project costs. *Journal of the Operational Research Society*. 46 809-18