

Cognitive and Organizational issues in Programming in the Large: Preliminary findings from a Case Study

Patrick Waterson and Chris Clegg

MRC/ESRC Social and Applied Psychology Unit
Department of Psychology
University of Sheffield,
Sheffield, S10 2TN, UK

pclpew@sunc.sheffield.ac.uk
C.Clegg@sheffield.ac.uk

Extended abstract

This paper reports early findings from a series of on-going longitudinal studies of the cognitive and organisational factors involved in building large scale software programs. In particular we focus upon one software project within a large UK bank. Our approach has been to concentrate upon the distribution of expertise and knowledge on the project; the organisation of project work; the use of CASE tools; and the problems faced by project members and mechanisms for dealing with these. As a means of partially integrating organisational and cognitive approaches to this area, as well as providing a framework for our research, we introduce a model of what we term the "division of cognitive labour".

1. Background

As a number of authors have pointed out (eg. Soloway, 1986), most of the research into the psychology of programming has so far concentrated upon individual programming activities, often within the context of the laboratory. Few studies exist of groups of programmers working together on a program, and even fewer of such activity taking place within an industrial or commercial context. Perhaps the most thorough investigation of programming in the large so far is due to the work of Curtis, Krasner and Iscoe (1988).

Curtis et al (1988) carried out a field survey of 19 large scale software projects. Their interviews with software developers led them to conclude that three main factors contributed to the many problems involved in building complex programs within tight deadlines: (1) the thin spread of application domain knowledge; (2) fluctuating and conflicting requirements; and (3) communication and coordination breakdowns within projects. All three factors were seen as the result of a complex interaction between organisational, social and cognitive phenomena.

One inherent limitation in Curtis et al's survey is its breadth. More detailed studies of large scale programming are needed to unpack each of the issues that are pointed to in their study. With this in mind we have concentrated our attention upon an intensive and detailed investigation of each of the three factors, amongst others, as they affect individual projects. In addition we plan to follow projects through until they are implemented and in operation by end-users.

2. Context and methods of Study

The project we describe here is located within the information technology department of a large UK bank based in the North-West of England. The Charging project as we shall call it, is made up of a number of software developers with a range of skills and expertise together building a program to manage the administration and updating of information relating to corporate accounts. Part of the work involves conversion of an existing charging system as well as liaising with an external contractor responsible for

most of the program coding. The program itself is very large and complex primarily because many of the banks customers have idiosyncratic and unique requirements in terms of how their accounts are charged. The project has currently been running for a total of two years and is due to be completed by Easter 1994.

So far we have carried out interviews with all of the key personnel assigned to the Charging project (eg. project managers and team leaders) as well as individuals involved in the day to day activities of the project (eg. systems analysts, program testers, programmers). At the present moment (Sept. 1993) we have carried out over thirty interviews within the bank as well as administering a questionnaire which is being completed by all system developers within the information technology department.

3. Early findings

3.1 *Distribution of expertise and knowledge on the project*

At this stage in our research we can distinguish between five distinct sources of expertise/knowledge which are distributed across the Charging project: computational knowledge (ie. relating to data structures, architectures, algorithms in the program); application knowledge (relating to typical inputs into the program and basic knowledge of what the program should do); domain knowledge (relating to day to day working practices within the finance area by end users); software engineering knowledge (relating to techniques for design, analysis, testing and the use of CASE); and project management knowledge (relating to the coordination and allocation of project resources). The interesting aspect of knowledge distribution here is the degree to which these categories overlap. Similar to the Curtis et al study, there tends to emerge a class of developers who could be said to be project "gurus" or "overseers". These individuals seem to adopt a cross-referencing strategy in their work on the program (cf. Pennington, 1987). Likewise we can discern individuals who have "local" knowledge of the program (ie. little overlap but possessing detailed knowledge of one aspect of the program), and individuals who have "partial" knowledge which overlaps between one or more of the categories of knowledge. The organisation and communication between these sources of expertise/knowledge can be seen as one reason why so far the project seems to be working effectively. In addition partial overlap between sources of application and domain knowledge has consequences for the degree of user involvement which is present on the project as well as the end product.

3.2 *The organisation of project work*

Project work seems to be organised largely around the specific tasks allocated to individuals (eg. system conversion, acceptance testing, program queries) as well as relating to the categories of developer knowledge/expertise. One interesting feature is the distribution of "overseers"; these appear to be have been carefully distributed around the project in order to ensure that problems relating to lack of knowledge in one area are catered for. Such allocation does not always correspond to seniority within the project (ie. "overseers" are not always team leaders). Likewise overall communication is viewed as poor by developers and not formalised; rather communication appears to be facilitated by informal, tacit boundary spanning activities by key project members. We plan to follow these issues up by a series of observational studies as well as further interviewing.

3.3 *The use of CASE tools*

Work on the program has been complicated by the addition of a set of CASE tools where little training has been provided for. Developers have had to learn these relatively opportunistically and in some cases this has caused problems. The design and analysis of the program for example has proved problematic mainly due to the mismatches between the working practices of programmers, the way CASE tools impose a structure on design and analysis, and existing guidelines for these activities set in place

by the bank. Reconciling these different approaches to design and analysis means that project members have to hold a set of competing representations of the Charging program.

3.4 The problems faced by project members

Project members work within very tight deadlines, a fact which often makes the Charging project a very stressful environment. Issues of time and money are compounded by the fact that the external contractors working on the main coding of the program are geographically remote; hence most communication takes place over e-mail. Although the changing requirements of the program are a constant source of concern amongst developers, most point to internal problems within the information technology department as the main "bottleneck" in terms of communication and coordination of program requirements. One upshot of this is that some aspects of the program are too powerful (ie. functionality built into the program early on is unlikely to be exploited by end users).

3.5 The "Division of Cognitive labour"

One way of viewing the Charging project is as a cognitive system involving a division of cognitive labour within and between individuals and artefacts. This includes the distribution of cognitive activities between roles (Hutchins, 1990). We are using some aspects of this conceptual framework to understand the way in which programming tasks are allocated within and between individuals, project sub-teams and the artefacts being used (ie. CASE tools). Our approach has been to examine examples of where mismatches in terms of the work of project members, and the tools they use, occur, in order to understand how the system works both in terms of its organisation as well as the cognitive demands inherent in such a highly skilled domain as large scale software development. We hope to present some examples of our model at the workshop in January.

4. Future work

We plan to carry out further interviews with members of the Charging project until the program becomes operational when we shall also interview end-users. We are also carrying out in parallel, investigations of a number of other software projects within the bank, as well as projects in other organisations. The findings from these studies will enable us to make comparisons between and within organisations.

References

- Curtis, B., Krasner, H. and Iscoe, N. (1988), A field study of the software design process for large systems. *Communications of the ACM*, 31 (11), 1268-1287. 13
- Hutchins, E. (1990), The technology of team navigation. In J. Galegher, R. Kraut and C. Egidio (eds), *Intellectual teamwork: Social and technological foundations of cooperative work*. Hillsdale, NJ: LEA.
- Pennington, N. (1987) Comprehension strategies in programming. In G.M. Olson, S. Sheppard and E. Soloway (eds.), *Empirical Studies of Programmers: Second Workshop*. Norwood, NJ: Ablex
- Soloway, E. (1986), What to do next: Meeting the challenge of programming in the large. In E. Soloway and S. Iyengar (eds.), *Empirical Studies of Programmers: first workshop*. Norwood, NJ: Ablex, pp. 263-268.