

Study of the Use of Rapid Prototyping in Industry

Horst Lichter, ABB DECRC

1 Motivation and Background

This contribution presents the results of case studies of industrial software projects in which explicit use was made of prototyping. Our major concern was to analyze the experience gained in the projects with the use of prototyping, and in particular to juxtapose these findings and the claims made for prototyping.

2 Projects analyzed

We have analyzed the following projects: a customer advice system, a distributed information system, a planning system, a document management system, a process control system which succeeded and a control system which failed.

The projects ranged from a large-scale project with a total budget of 240 person-years to small projects in the order of 2 person-years. The developer teams involved in the projects included both the computer departments of large industrial corporations and small to medium-sized software manufacturers. A more detailed description of the case studies can be found in [1].

3 Results and Consequences

In each of the projects analyzed, the prototypes used exhibited specific characteristics and effects on the software development process. We

have attempted to generalize the results obtained, thus enabling these findings to be drawn on for other projects. In the rest of this section we present and explain the results obtained.

Result 1: Clients' knowledge about prototyping is low

One basic problem we found was that the clients, in particular, know very little about the scope of prototypes and the benefits they offer. They often expect far too much of the first prototype, which is merely designed for presentation purposes. Particularly in cases where the prototype models the user interface along with a rudimentary functionality, they are easily led to believe that most of the work is done.

Result 2: Domain knowledge is essential

Another fact which emerged was that the developers must have sufficient knowledge of the application area and the work environment to prevent them from building a prototype whose behavior is so far removed from that of the proposed target system as to make it unsuitable for further development. System analysis is therefore a necessary prerequisite for successful prototyping, and it is imperative that an expert from the application area participate in this analysis. The analysis should also help establish a uniform terminology for users and developers alike.

Result 3: Documents provide no sufficient background

We occasionally encounter the situation where a team of developers embark on a prototyping project with nothing in their hand but existing documents and written requirements. The problems caused by their lack of familiarity with the actual application situation are further exacerbated by the implicit limitations imposed on the terminology used and the range of conceivable solutions open.

Result 4: End user participation is essential

A further misconception encountered throughout relates to the question of how and to what extent users can be involved in the development process. In many cases, user management is reluctant to allow the actual end users of an application system to participate in the evaluation of prototypes, let alone in discussions on design options. An important help in clarifying the requirements for an application system can be provided by scenarios which are drawn up by the users.

Result 5: Prototyping is a learning process

Analysis of successful development projects has shown that, despite the improvement in communication brought about by prototyping, a factor of major importance is the need for developers to familiarize themselves with the problems and working routines of the application area. This only goes to substantiate the view of software development as a mutual learning process among all the groups involved. An optimal situation for such a learning process on the part of the developers is where there are members of the developer team who are doubly qualified, both as software engineers and application experts.

Result 6: Face to face communication is important

If we regard the active involvement of all relevant groups as crucial to the success of a prototyping project, then this also means that efforts must be made to ensure that these different groups are located as near to each other as possible. Our experience has shown that, despite all the available telecommunication facilities, personal discussion remain indispensable for evolving a specific project identity and project culture.

Result 7: New types of contracts are needed

The use of prototyping throughout the development process is largely incompatible with the milestone documents of a classical life cycle plan. Performing planning and project control along the lines of milestones, which are chiefly for the benefit of management, often proves an obstacle to project progress. Specific paid for and documented activities that are crucial to the development process itself, such as a prototype used in requirements analysis or a formalized glossary of terms, etc., would appear more suitable here. But provision should also be made in the development contract for prototyping-specific activities such as the reimplementation of the initial prototype, insofar as it merely served acquisition purposes and was not designed to be reusable.

Result 8: Documentation is indispensable

The documentation continues to be the Cinderella of software development – that is something even prototyping cannot change. Even if prototyping appears, at first sight, to make written documents largely superfluous – the prototype as an operative object replacing

the specification – the effects of inadequate documentation soon make themselves felt in practice, too.

Result 9: Screen forms are only part of the story

Our analysis shows that a uniform approach to prototype construction – from the user interface "downwards" – is not helpful. The crucial factor here is the nature of the problem in hand. If the functionality of an application system is largely prescribed and experience in using computers for similar problems is already available, the frequently propagated form of "horizontal prototyping" is likely to be the best solution. On the other hand, where a system's functionality is still largely unclear, this is precisely the point at which prototyping must come in.

4 Concluding Remarks

The examination and analysis of the projects have shown that – particularly when used in conjunction with an evolutionary development strategy – prototyping is conducive to the quality of both the product and the development process. Another fact which has emerged is that the positive effect of prototyping can be enhanced in cases where all the parties involved in the development process are aware of both the potential and the limitations of this approach, and where both the organizational framework and the contractual arrangements are tailored to this development model.

A final factor of crucial importance is the use of suitable tools to support prototyping. In many cases the use of prototyping is only possible if these technical and organizational conditions are fulfilled. There are, no doubt, situations in which prototypes can be successfully built using existing traditional tools and methods, but an appropriate development environment is a prerequisite for the rapid development cycles that largely account for the attractiveness and efficiency of prototyping.

References

[1] H. Lichter, M. Schneider-Hufschmidt, H. Züllighoven (1994): Prototyping in Industrial Software Projects - Bridging the Gap Between Theory and Practice. IEEE Trans. on SE, vol. 20, no. 11, pp 825 - 832.