Prototyping as software development approach

The world is changing faster thanks to technology advances. There is cutting-edge technology nowadays that does not include or use hardware and computational solutions. Such hardware and software technologies require continuous software development. The ability to deliver more competitive products depends on solutions that respect user requirements (Floyd, 1984 in Carr, M.,1997). So, it is necessary to carefully plan and deliver software solutions. One of the ways is prototyping.

The prototyping is defined as arbitrary steps that are followed to achieve the required functionality of the software (Arnowitz, J., 2007). These strategies are innovative because of the joint between the willingness of customer and the state of the product. The prototype is itself an intermediary stage of the application which can be either approved or refused by the client. It can be in a computing format or in a planned way (Carr, M.,1997).

Companies have own prototyping roadmap for the development of their software. Every such process starts with the proposal of a first prototype, which tries to match the current customer needs. The process continues with one or more iterations, depending on the prototyping goals such as experimental, exploratory or evolutionary types (Lichter, H., 1993). Essentially, each iteration has a similar structure. The event that triggers this kind of repetition is the evaluation of the prototype such as a presentation in front of the client (e.g. user review, see Carr, M., 1997). It is a challenging task because the process leads to two directions. In an optimistic case, if the software passes the check, then it is ready to move into further steps. Otherwise, the prototype runs through several revision(s) and refinement activities, and after that, it returns to the evaluation process.

From the description above, prototyping in software development is a lifecycle paradigm that leads to several approaches depending on how clearly and promptly the clients communicate their ideas. Meanwhile, the software prototype is capable of building upon all previously agreed decisions. Prototyping can be an efficient method for software development when the customer is actively involved.

Reference

- Arnowitz, J., Arent, M. and Berger, N., 2007. *Effective prototyping for software makers*. San Francisco: Elsevier Inc
- Lichter, H., Schneider-Hufschmidt, M. and Züllighoven, H., 1993, May. Prototyping in industrial software projects—bridging the gap between theory and practice. In *Proceedings of the 15th international conference on Software Engineering* (pp. 221-229). IEEE Computer Society Press.
- Carr, M. and Verner, J., 1997. Prototyping and software development approaches. *City University of Hong Kong. Retrieved January*, *8*, p.2008.