

Using Metaphors in eXtreme Programming Projects

Yael Dubinsky¹ and Orit Hazzan²

¹ Department of Computer Science, Technion, Israel

² Department of Education in Technology & Science, Technion, Israel

Abstract. Metaphor is one of the twelve practices of Extreme Programming (XP), and definitely among the more difficult ones to teach and use. We present our experience with the use of metaphors and conclude with suggested guidelines for teachers.

1 Introduction

Metaphors are used in order to understand and experience one specific thing using the terms of another thing ([3]). Communication which uses the metaphors world of concepts to improve our understanding of the world of concepts of the specific thing we are trying to understand, refers not only to instances in which both worlds of concepts correspond to one another, but also in cases in which they do not. If both worlds of concepts are identical, the metaphor is not a metaphor of that thing, but rather the thing itself. According to [1], a good metaphor for XP is the process involved in learning to drive. In software development, the process is controlled by the execution of many small adjustments, similar to driving a car, but there is no teamwork involved in driving. Still, the understanding of this metaphor is worthwhile. Our own experience has led us to map the metaphor as a practice that requires a high level of cognitive awareness when implemented ([2]). When we talked about metaphors with students who are working together on software projects, it sometimes seemed that the practice is forced and unnatural. The way in which metaphor improves communications was not immediately apparent. This encouraged us to further investigate the use of metaphors in various academic courses.

2 Using Metaphors in XP Projects

We studied how metaphors were used in XP projects executed by students in three different courses. Data were gathered from the lecturers notes, observations, videotapes, interviews, and electronic forums. Due to space limitations, we focus here on a specific CS-major course, named Projects in Operating Systems, in which XP teams of 12 students worked on a project. During the first planning sessions of one of the teams, after listening to the customer stories, most students were uncertain about how to think about the file navigator, the

projects topic. The customer emphasized that he was not comfortable with the current tree hierarchy of files, and specifically with the lack of automatic synchronization between related files. One of the students suggested the navigator be treated as if it were an association graph. An association graph is made up of nodes that represent specific subjects, and connecting arcs, which denote equivalent relationships between the nodes. The idea was to regard each such node as representing a file, and each arc between two files as denoting a first-degree relation with respect to the file contents. The student explained that this is the way in which our mind works. During a discussion with the supervisor, in which the students used the above metaphor to explain how our mind relates to data, they began to realize how the project could be planned. Eventually, the metaphor 'File organization is Mental association' accompanied the groups discussions throughout the semester. Following are students expressions that are instances of this metaphor: "The distance between one file and another can be infinite"; "We can ask which files have a level-2 relation to a specific file".

3 Conclusion

Following are several lessons that we find appropriate to share with other teachers. The first guideline is to be aware of metaphors. Students, like other people, use metaphors naturally, and we can learn from those metaphors about students understanding and use it in our teaching. When we explain about metaphors, and students begin to become exposed to the understanding and new horizons that metaphors opens to them, they respond positively and want to extend their use. This leads to the need to extend the use of metaphors, talk about them and encourage their use. The second guideline is represented by the metaphor More is Better. In this case, the more metaphors we use, the more we improve the understanding of the project and get students to talk in general, and about the various aspects of the project in particular. The more metaphors we use, the more we improve our understanding of the project and of the difficulties our students might encounter during the course of the project. No special difficulties arise when using several metaphors for different parts of the project, using each one according to need. Students are willing to use different metaphors for the same topic and it does not interfere with their progress. On the contrary, multiple metaphors provide elaboration and lead to a more confident understanding. Encouraging students to provide multiple metaphors is a worthwhile practice.

References

1. Beck, K.: Extreme Programming Explained: Embrace Change. Addison-Wesley 2000
2. Hazzan, O. and Dubinsky, Y.: Teaching a software development methodology: The case of Extreme Programming. In The proceedings of the 16th International Conference on Software Engineering Education and Training. Madrid, Spain 2003
3. Lakoff, G. and Johnson, M.: Metaphors We Live By. The University of Chicago Press 1980