Does the Level of Detail of UML Diagrams Affect the Maintainability of Source Code?: A Family of Experiments.

Ana M. Fernández-Sáez[1,5], Marcela Genero[1], Danilo Caivano[2,3], and Michel R.V. Chaudron[4,5]

[1]ALARCOS Research Group, Department of Technologies and Information Systems,
University of Castilla-La Mancha
Paseo de la Universidad 4, 13071, Ciudad Real, Spain
+34 926295300 Ext. 3740
ana.fernandez@alarcosqualitycenter.com
Marcela.Genero@uclm.es

[2]Department of Informatics
University of Bari
Via E. Orabona 4, 70126 Bari, Italy
+39 080 544 2300
danilo.caivano@uniba.it

[3]SER&Practices s.r.l.
Spin Off Company of the University of Bari
Via E. Orabona 4, 70126 Bari, Italy
+39 080 544 2048
danilo.caivano@searandp.com

[4]Joint Computer Science and Engineering Department of Chalmers
University of Technology and University of Gothenburg
SE-412 96 Gӧteborg, Sweden
chaudron@chalmers.se

[5]Leiden Institute of Advanced Computer Sciences
Leiden University
NielsBohrweg 1, 2333 CA Leiden, The Netherlands
+31 715277065 (secr 7061)
{fernande,chaudron}@liacs.nl

Abstract. Although the UML is considered to be the de facto standard notation with which to model software, there is still resistance to model-based development. UML modeling is perceived to be expensive and not necessarily cost-effective. It is therefore important to collect empirical evidence concerning the conditions under which the use of UML makes a practical difference. The focus of this paper is to investigate whether and how the Level of Detail (LoD) of UML diagrams impacts on the performance of maintenance tasks in a model-centric approach. A family of experiments consisting of one controlled experiment and three replications has therefore been carried out with 81 students with different abilities and levels of experience from 3 countries (The Netherlands, Spain, and Italy). The analysis of the results of the experiments indicates that there is no strong statistical evidence as to the influence of different LoDs. The analysis suggests a slight tendency toward better results when using low LoD UML diagrams, especially if used for the modification of the source code, while a high LoD would appear to be helpful in understanding the system. The participants in our study also favored low LoD diagrams because they were perceived as easier to read. Although the participants expressed a preference for low LoD diagrams, no statistically significant conclusions can be drawn from the set of experiments. One important finding attained from this family of experiments was that the participants minimized or avoided the use of UML diagrams, regardless of their LoD. This effect was probably the result of using small software systems from well-known domains as experimental materials.

Keywords: UML diagrams, software maintenance, level of detail, controlled experiment, replication, family of experiments.

Empirical Software Engineering, February 2016, Volume 21, Issue 1, pp 212–259 (DOI: 10.1007/s10664-014-9354-4)