

Comprehensive Documentation Made Agile – Experiments with RaPiD7 in Philips

Ko Doods¹ and Roope Kylmäkoski²

¹ Philips, Building SFJ-3, Glaslaan 2, 5616 LW Eindhoven, The Netherlands
ko.doods@philips.com

² Nokia, Hatanpäänvaltatie 30, 33101 Tampere, Finland
roope.kylmakoski@nokia.com

Abstract. This paper addresses the almost never-ending headache the role of documentation has given for software projects. *Working software* has been given recently a focus over *comprehensive documentation*, yet the required documents should be authored. This paper “revisits” the approach developed by Nokia improving the documentation work without scarifying the quantity or quality of documentation. The method is called RaPiD7. The cases presented are from Philips Digital Systems Laboratory. This paper elaborates the method by providing insights to applying RaPiD7 in practice, explains the encouraging results of the experiments and gives tips for practitioners of the method by explaining the lessons learned in Philips.

1 Introduction

In recent years agile methods (see for example [3]) have been the focus of discussion in the area of software engineering. Agile methods seem to be a response to the heavily prescriptive processes (see for example [5]) that dominated the field in the early 1990s. Agile methods focus more on the human aspects of software engineering than these so-called prescriptive processes, and place human interaction over tools and processes [1]. Agile methods also state that documentation is often a heavy and unneeded form of communication.

In [4] another method for documentation work is presented that combines to a certain extent the good sides of both directions, in other words, from prescriptive processes and from agile methodologies. The method addresses the challenge of creating understanding, sharing understanding and storing the created understanding in software projects. These challenges are addressed by a stronger focus on planned human interaction and early joint decision-making by the project development team. In practice, this method is about planning the needed human interaction and decision-making as part of project planning in the form of facilitated workshops, and then subsequently carrying out this plan. The method is called RaPiD7 (Rapid production of documentation – 7 steps), and it has been developed by Nokia during 2000-2001.

Similar methods to RaPiD7 exist (see for example [2], [7], [6] and [8]), therefore the intention of this paper is not to provide a totally new or unique approach for

software engineering. Consequently, this paper is not comparing the different methods, but is rather a lessons-learned paper providing insights on a particular case with a selected collaborative method for software related documentation and design. A similar method to RaPiD7 called JAD [6] has existed since the 1970s and recently agile methods have been addressing the same field especially by a method called Agile Modeling [2]. However, it does not appear that any of these methods, be it JAD, AM or RaPiD7, have gained the industry de-facto status they should have. We believe that methods like RaPiD7 should be an integral part of almost any software process. Thus we also see publishing the usage results as paramount. There have been too few concrete steps taken on ensuring effective human interaction in the field of software engineering. Teamwork has often been encouraged, but the real support has been missing.

Early results from the use of RaPiD7 have already been presented in [4]. Work is still continuing inside Nokia, with more results to be published. RaPiD7 use has now spread outside Nokia too. First trials of the method outside Nokia were carried out in Philips during April 2004. This paper explains this case in detail, giving concrete suggestions how the method can be applied (something that was not presented in [4]) with some encouraging results similar to what has been published by Nokia before. Furthermore, the analysis of the applicability of the method in another environment besides Nokia is briefly discussed, too.

This paper first briefly explains RaPiD7 and the baseline status in Philips. Then the actual cases from Philips are presented with details about the planning and implementation phase. The results from the cases are presented and analyzed. Finally conclusions are drawn from the cases and from the whole paper.

2 RaPiD7 in Philips – A Case Study

2.1 Brief Introduction to RaPiD7

In RaPiD7, creating understanding for the specifications is mostly done jointly, information sharing is done continuously early on and document writing is done as far as possible jointly, too. Furthermore, the quality assurance is built into the way of working. In practice, all this is carried out in the form of facilitated workshops. The approach is presented briefly in Fig. 1.

Furthermore, RaPiD7 provides a three-layer structure. First the *project layer* describes how human interaction and joint decision-making is planned for software projects. In practice this means identifying the cases for applying RaPiD7. The *case layer* describes how the selected cases such as documents are to be created in consecutive workshops, and the *workshop layer* describes how the actual work is carried out in form of facilitated workshops.

RaPiD7 workshops comprise from seven steps. The steps aim at providing information on how to organize efficient workshops in software projects. The steps of RaPiD7 are shown in Fig. 2.