COTS and Open Source Software Components: Are They Really Different on the Battlefield?

Piergiorgio Di Giacomo

European Software Institute,
Parque Tecnológico Edificio #204,
Zamudio, Bizkaia, Spain
Piergiorgio.DiGiacomo@esi.es

Abstract. When referring to Open Source Software (OSS) components, researchers, coders and managers do not feel comfortable in defining them as COTS. Many discussions have been aimed to decide whether or not OSS can be considered a COTS without reaching the unanimous consensus of the different international communities. This paper abandons any theoretical aspect of that question and focuses on the practical steps to follow when assembling component-based systems using also OSS components. All the activities normally performed when integrating COTS in a in-house built software are reviewed with the intention of underlining if the availability of the source code (and its possible exploitation) makes any difference. Moreover this article analyzes all the activities to perform when using OSS in a component-based system that are not necessary when using COTS. The purpose of this paper is to provide a guideline for the correct use of OSS within component-based systems, and not to answer whether OSS are considered or not COTS, leaving this task to the reader.

1 Introduction

The aggregation of different software components is nowadays a very common practice in the creation of complex systems. Not all the components in a system have to be inevitably built in-house. Outsourcing Commercial Off-the-Shelf (COTS) components can result convenient when an organization does not have time, internal competencies or resources to develop a particular functionality.

Outsourcing commercial components is often object of numerous debates and has produced different opinions among researchers, coders and managers. One of the main limitations when using COTS is the fact that the source code is not available. Consequently they cannot be entirely trusted and exploited. In fact, the unavailability of the source code is only one of the disadvantages the user has to face when using COTS. The impossibility to drive the evolution of the product, the obligations to upgrade the product and the conformity of the new version to the wrap code already written (and many others) are also aspects that often create discontent among the COTS users.

In many cases the adoption of Open Source Software (OSS) components can be a way of overcoming such problems. The first part of this paper summarizes the

X. Franch and D. Port (Eds.): ICCBSS 2005, LNCS 3412, pp. 301 – 310, 2005.

benefits and the disadvantages when using OSS components. It also clarifies the most common myths about OSS and OSS world. The second part analyzes the activities to be performed when outsourcing software components. After recalling the usual activities of software development, it compares the COTS vs. OSS activities and their similarities or differences. Moreover, the peculiar tasks of OSS component integration are exploited in deep detail.

2 OSS Myths and Mystifications: Why Management Do not Trust OSS

The confidence that coders have with the OSS world, their motivation when developing in communities and their enthusiastic view of the OSS, often supported by bad feelings against big corporations are making OSS to penetrate the industrial sector faster than initially thought.

Nevertheless, management is often more than reluctant to adopt OSS solutions. This cultural gap between coders and management is sometimes based on misinformation on the technical aspects of OSS.

The following sections analyze all the myths about open software and the truths in each of them. This clarification about the OSS and its implication is essential when deciding to implement on OSS solutions for both risk analysis and decision taking task.

2.1 The Money Factor

When Open Software solutions are exploited, their price is certainly attractive but it is also source of concern for the management. Total cost of ownership can be proven to be not equal to zero, as OSS components have to be integrated, tested and maintained just like COTS. When acquiring a COTS, managers go through a process that is not different by any other acquisition: they prepare requirements, select suppliers and their products, sign a contract and, most important, pay for what they buy.

The fact that a physical entity is providing the needed component provides indeed more assurances to the management. If the component does not work there is somebody to blame and the fee paid for the component is reassuring about the value of the component itself.

On the other side, a component acquired for free gives the impression of something created by amateurs. Even if in some cases this can be true, OSS has originated by the need to reach goals that can be achieved only by the cooperation of many parties:

1. An open group of developers and users who join their strengths to create an open software component since the ones already on the market are not suitable for them for economical or technical reasons.
2. A closed group of partners (normally SME[1]) establishing a small consortium for the creation of software which code is available only to the partners, with the double intention of decreasing the cost of the commodity software and opening of a new market in a short time period.

[1] Small-Medium Enterprise.