

Chapter 2.24

An Incremental Functionality–Oriented Free Software Development Methodology

Oswaldo Terán

ENDITEL; Centro de Micro Electrónica y Sistemas Distribuidos and Centro de Simulación y Modelos, Universidad de los Andes, Venezuela

Johanna Alvarez

CENDITEL, Venezuela

Blanca Abraham

CEMISID Universidad de los Andes, Venezuela

Jose Aguilar

CENDITEL; Centro de Micro Electrónica y Sistemas Distribuidos, Universidad de los Andes, Venezuela

ABSTRACT

This chapter presents a methodology used as reference model for a free software factory that is part of the National Centre for Free Technologies in Venezuela. This centre is oriented at promoting free software development for serving mostly the public sector in order to promote endogenous development and technologic autonomy. Under this strategy, strengthening the software small and medium size enterprises and cooperatives, by

allowing them to participate in different projects (improving their know-how) and providing them with a methodology for increasing their capabilities and software quality, is necessary and urgent. This methodology plans the development process incrementally, based on a prioritisation of the software functionalities development in accordance to the functionalities risks, development urgency, and dependencies. It combines aspects of the two styles of free software development,

namely cathedral and bazaar. The development process is centralised, in essence collaborative, and continuously allows source code release.

INTRODUCTION

The Free Software Factory (FSF) of CENDITEL (Venezuelan national centre for promoting free technologies) has been conceived and created as part of the efforts of the Venezuelan State aiming at increasing endogenous development and technological sovereignty. In particular, it intends to strengthen the national software sector, especially the small and medium software enterprises (including the cooperatives), by allowing them to access the technology and participate in the software market, on one hand, and to increase their capabilities and software quality, on the other hand.

Two styles exist for developing free software: the cathedral style and the bazaar style. In the cathedral mode, software is developed from a unified *a priori* project that prescribes all the functions and the features to be incorporated in the final product. Programmers' work is centrally coordinated and supervised in order to assure the integration of various components. On the other hand, in the bazaar style, software emerges from an unstructured evolutionary process. Starting from a minimal code, groups of programmers add features and introduce modifications and patches to the code. There is no central allocation of different tasks; developers are free to develop a given program in directions they favor.

This chapter presents an attempt at building a free software development methodology having many characteristics of the cathedral style but keeping certain principle of the bazaar mode. The methodology has been developed at a public organisation which responds to public sector free software necessities and requirements that must be satisfied in a limited time period. Because of this, it is necessary to adopt the cathedral mode of

work while taking key advantages of the bazaar style. For instance, it is allowed that developers from outside the organisation contributes with software coding, testing, and so forth; these external developers do not follow a centrally controlled process; and the software code is made public as soon as it is tested.

This methodology assumes an organisational structure oriented towards specific processes. The processes dedicated to software development are:

- *Process # 1:* Free Software Project Management
- *Process # 2:* Specific Project Administration
- *Process # 3:* Free Software Application Development

Actions to be carried out in these processes are classified in steps and activities. In particular, steps and activities in the third process are implemented by the following six phases. This methodology has taken ideas from diverse software development methodologies, methods, and models such as the extreme programming method (Beck, 2004), the rational unified process (Kruchten, 2000; Pollice, 2001; Probasco, 2000), the watch method (Montilva, 2004; Montilva, Hamzan, & Ghatrawi, 2000), and the model of processes for software development (MoProSoft) (Oktaba et al., 2005). Due to the fact that these models and methods, except extreme programming, have been proposed proprietary software development, it has been necessary to adapt the hints, ideas, or procedures taken from them to the free software development needs.

The methodology to be proposed has been validated at the FSF of the Foundation for Science and Technology of the Mérida State in Venezuela (FUNDACITE-Mérida). This factory has permitted us to understand better, empirically, the real needs of a free software development process and has also been a source of interesting ideas. The

14 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

www.igi-global.com/chapter/incremental-functionality-oriented-free-software/29430

Related Content

A Decision Making Paradigm for Software Development in Libraries

Harish Maringanti (2022). *Research Anthology on Agile Software, Software Development, and Testing* (pp. 1444-1457).

www.irma-international.org/chapter/a-decision-making-paradigm-for-software-development-in-libraries/294526

How to Systematically Embed Cycles in Balanced Hypercubes

Hsuan-Han Chang, Kuan-Ting Chen and Pao-Lien Lai (2017). *International Journal of Software Innovation* (pp. 44-56).

www.irma-international.org/article/how-to-systematically-embed-cycles-in-balanced-hypercubes/169917

Understanding the Role of Use Cases in UML: A Review and Research Agenda

Brian Dobing and Jeffrey Parsons (2002). *Successful Software Reengineering* (pp. 111-128).

www.irma-international.org/chapter/understanding-role-use-cases-uml/29972

A Framework for Understanding and Addressing the Semiotic Quality of Use Case Models

Pankaj Kamthan (2009). *Model-Driven Software Development: Integrating Quality Assurance* (pp. 327-351).

www.irma-international.org/chapter/framework-understanding-addressing-semiotic-quality/26835

Mobility Markov Chain and Matrix-Based Location-Aware Cache Replacement Policy in Mobile Environment: MMCM-CRP

Ajay Kumar Gupta and Udai Shanker (2021). *International Journal of Software Innovation* (pp. 88-106).

www.irma-international.org/article/mobility-markov-chain-and-matrix-based-location-aware-cache-replacement-policy-in-mobile-environment/289171