Chapter X
Transitioning from Code-Centric to Model-Driven Industrial Projects: Empirical Studies in Industry and Academia

Miroslaw Staron
IT University of Göteborg, Sweden

ABSTRACT

Introducing Model Driven Software Development (MDSD) into industrial projects is rarely done as a “green field” development. The usual path is to make a transition from code-centric (CC) development in existing projects into MDSD in a step-wise manner. Similarly to all other software development activities, software quality assurance needs to be adjusted to meet the new challenges arising when using models instead of the code for the mainstream development. In this chapter we present a set of empirical data on the issues related to transitioning from CC to MDSD projects in industry. First, we present results from a set of experiments evaluating how a domain specific notation affects the effectiveness and efficiency of reading techniques used for inspecting models. Second, we present a comparison of productivity increase when changing to MDSD projects from one of the large Swedish companies. Finally we present a short survey on the prioritization of products, projects, and resource metrics in MDSD projects.

INTRODUCTION

Introduction of new development paradigms and technologies is never a simple task. It is even harder when we consider large software development organizations with a long history of using other methods and with a portfolio of long-lasting software products. The long-term nature of these projects coupled with their continual development requires stable and reliable development methods. In contrast, the global economy with its competition drive companies to seek out and adopt new methods and tools to improve productivity and enhance their competitive position with innovative
products of higher quality and rapid development cycles. Using modeling in software development promises improved quality and productivity through increased automation of the software development process.

Model Driven Software Development (MDSD) comes in many flavors – starting from using general-purpose modeling languages such as UML (Unified Modeling Language, Object Management Group, 2004), and ending with a set of integrated Domain Specific Modeling Languages (DSLs). The main characteristic of MDSD projects, regardless of the modeling notation used is that models play the central role in the process. Models are used for code generation, but also for early quality assessment activities (e.g. software inspections, testing executable models), or for estimations.

This chapter addresses the problem of providing empirical evidence on how much improvements could be expected in the first projects conducted according to the principles of MDSD. It also addresses the issue of which aspects should a project manager consider when undertaking the first projects in MDSD, and which metrics should be customized for MDSD already for the first project.

In order to address the problem we analyze a set of empirical studies performed both in industry (case studies at Ericsson) and in academia (experiment with software inspections). By providing empirical evidences and experiences from industry we support managers of future software projects in making informed decisions concerning adoption of MDSD.

The chapter presents experiences of improvements brought by model-driven development in industrial projects and the expected increase of effectiveness of software inspection of models elicited through experiments.

The chapter is structured as follows. Section 2 presents the background for the claims presented in the chapter, outlines the existing problems in detail and overviews the existing literature in the area. Section 3 is the core of the chapter and presents the empirical studies, in the end discussing their validity. Section 4 presents a short meta-analysis of the series of studies presented in Section 3. Section 5 contains conclusions. The chapter concludes with a section on future research directions related to using reading techniques as a quality assurance technique for models, and research in productivity assessment in MDSD projects.

**BACKGROUND**

Based on the roadmap for research on MDSD (France & Rumpe, 2007) it shows that MDSD is not yet a fully established technology and it will still evolve. Therefore, an issue could be raised whether it is mature enough to be adopted or whether it delivers on its promises. The main challenge in the industrial adoption of MDSD is that MDSD needs investments to be effective: the larger the investments, the larger the benefits. In large software projects and in large companies the adoption of MDSD is burdened with all the problems of immature technology (how to justify real expenses based on promises?) and organizational resistance (how do we know that the technology actually improves our way of working?). Herein lies a challenge – how to gradually build up the confidence that using models in a project can help to increase productivity (or quality, or ideally – both). As we are able to show in the case study at Ericsson in Section 3.3, in addition to investing in technology, the investments should also contain costs of coaching (making sure that modeling knowledge is in place), model migration, or gradual migration process.

Transitioning of software practices from document and code centric into model driven can take several years, which is shown in a recent study from Motorola (Baker, Loh, & Well, 2005). The length of time depends on the size of the organization and the range of the products of the company. The long time span of the adoption