Chapter 69
Measuring the Progress of a System Development

Marta (Pląska) Olszewska
Åbo Akademi University, Finland & Turku Centre for Computer Science, Finland

Marina Waldén
Åbo Akademi University, Finland & Turku Centre for Computer Science, Finland

ABSTRACT

For most of the developers and managers, the structure and the behaviour of software systems represented in a graphical manner is more understandable than a formal specification of a system or than plain code. Our previous work combined the intuitiveness of UML with the development rigour brought by formal methods and created progress diagrams. In progress diagrams, the design decisions within a system refinement chain are assisted by the application of patterns and illustrated in a comprehensible and compact manner. In order to rigorously assess and control the design process, we need to thoroughly monitor it. In this chapter we show how the application of generic refinement patterns is reflected in measurements. We establish measures for the evaluation of the design progress of the system, where the progress diagrams are assessed from the size and structural complexity perspective. Our motivation is to support the system developers and managers in making the design decisions that regard the system construction.

INTRODUCTION

Nowadays software systems are being built using various technologies, appropriate to their application. Project leaders can choose between flexible agile methods (Martin et al., 2001), a spiral methodology (Boehm, 1988) for large and expensive projects, a waterfall model (Royce, 1970) for iterative developments, or prototyping originating from industry, among many others. Moreover, more rigorous approaches, like formal methods (Ladkin & Thomas, 2009), can be used for large, complex and critical system developments, e.g. those intended for space, aircraft or military sectors. Achieving high quality software is an aspiration of every development (Hayes & Jarvis, 1999). Hence, it is essential for software measurement to be an elementary part of the quality assurance program within every development process, including rigorous ones. When the

DOI: 10.4018/978-1-4666-4301-7.ch069

measurements encompass several metrics, like size, complexity or effort, they can be regarded as reasonable quality indicators.

Since software changes are inevitable, it is crucial to be able to handle them efficiently (DeMarco, 1986). Monitoring the progress of the development benefits in better control and quicker reaction to the changes being made, e.g. by modifying the current modelling approach or refining the model. Given that some measures, like complexity (McCabe & Butler, 1989), can be computed early in the development cycle, they add significant value to managing the software process, as well as the software itself. Moreover, they can contribute to cost reduction and increase the maintainability and modularity.

We believe that the overall quality of the development process, including the systems fundamentals in the design phase strongly influences the quality of the final system. We are interested in modelling large and complex critical systems, embedded systems in particular, and focus on the initial stages of the development. We think that performing the changes at this point is efficient in the sense of time and money. Managing the complexity at this point influences the overall quality of the system and its enhancements. It also impacts specific quality attributes, such as dependability or, more specifically, maintainability or resilience. Therefore, it is possible to diminish the threat against dependability, for example, by constantly controlling the development quality with measurements.

The measurement activities in a software development project should already be applied by the designers. These are the personnel responsible for the modelling decisions, who have to take into account the specificity of the project, knowledge about the application domain and the environmental settings. Also at this point the analyst or quality assurance managers should have access to the collected data in order to present their findings to the project manager, who then makes higher level decisions about the project at its early stage. The testers, as a separate development group, should be aware of the outcome of measurements. The measurement findings can then be juxtaposed with testing results and, as a consequence, testers can propose possible changes.

The combination of well known methodologies that have been proven to be efficient and successful needs to be supported with a good engineering practice. In our research we benefit from formal methods and the Unified Modelling Language, which are merged with measurements for the purpose of quality assurance. A certain quality, including dependability, can be guaranteed and achieved by continuously monitoring the system with measures. The value of the system being created lies in system planning, anticipation of future needs and feasibility of redesign. However, this is not to be accomplished without the innovative use of technologies and continuous supervision of development progress.

One of our goals in this chapter is to show how generic refinement patterns are reflected in measurements. Moreover we want to demonstrate how design decisions influence the measurements and the quality of the system being constructed. Specifically we concentrate on evaluation of progress diagrams and pattern-based development with respect to size and complexity measures. The approach can be extended to tackle a more generic problem of statechart diagrams assessment. Our research results are targeting developers (particularly designers and system architects), as well as project and system managers. The methodology is intended for the design stage of the development and can assist in the system documentation.

This chapter describes a method to tackle complexity in system development using a combination of rigorous approach and graphical notation. The former allows the developer to accurately identify system functionality, while the latter helps to maintain an insight and comprehension of the complexity and size of the system. The progress