Model-Driven Development Processes and Practices: Foundations and Research Perspectives

Regina Hebig[1], Reda Bendraou[1], Markus Völter[2], Michel Chaudron[3]

[1] Université Pierre et Marie Curie, LIP6, Paris (France), forename.surname@lip6.fr
[2] independent/itemis (Germany), voelter@acm.org
[3] Chalmers Technical University and University of Gothenburg (Sweden), michel.chaudron@cse.gu.se

Abstract. **MD^{2}P^2** is a workshop about the interrelation of model-driven development (MDD) and development processes. The workshop provides a forum for researchers and practitioners to exchange experiences on the questions how processes need to adapt or can be adapted when model-driven techniques are applied. We argue that the interrelation between MDD and development processes can be crucial for the success of MDD. For example, the need to adapt a process when introducing MDD can be a reason to decide against an MDD adoption. Further, we aim to give an introduction to foundations and research perspectives. **MD^{2}P^2** is co-located with ACM/IEEE 17th International Conference on Model Driven Engineering Languages & Systems.

1 Introduction

Model-driven development, which includes the synthesis of executable systems from models or the use of abstract languages, e.g. UML, Simulink, or DSLs, and software development processes are used to reach similar targets, such as increasing the quality of software or the efficiency of software development. Although, these goals are approached differently, MDD and software development processes are not totally independent.

In fact, the literature has a multitude of proposals for MDD specific development processes. Proposals for the use of MDD in context of established development processes, such as the V-Modell XT or SCRUM, however, sometimes include significant adaptations of the development process. Stakeholders' responsibilities might be as affected as quality assurance activities, which need to respect the structure of the artifacts used. Partly, even process phases and sprints are adapted to enable a combination with MDD. It seems that most adaptations have pragmatic reasons, and aim at supporting a fruitful combination.

However, it is also possible that MDD and a particular development process do not fit together well. For example, the benefit of an agile approach might be reduced, if long running transformation chains enforce long running sprints.

Further, an MDD approach that requires to build languages and transformations first before ”business value” can be delivered to the customer, conflicts superficially with some agile processes.

Thus, companies that aim at adopting MDD also have to face the questions, how appropriate development processes have to look like as well as whether and how existing development processes can be adapted. It is important to consider that development process adaptation can be cost intensive (e.g. due to required training of developers or changes in the involvement of stakeholders).

In summary, the questions whether and how an established development process must be adapted when MDD is introduced are crucial, since they can impact the efficiency of the development processes as well as the costs of the MDD introduction. In face of the growing number of MDD techniques and the variety of MDD approaches applied in practice, there is a need for systematic guidelines or best practices that help with adjusting or tailoring of development processes, when MDD is introduced.

In the following we give an introduction into foundations of the workshop's topic and present various research perspectives. We then provide an overview on this first edition of the **MD^{2}P^2** workshop.

2 Foundations

Software processes define various aspects of development, such as phases or tasks, but also documents that need to be created at different points in time. Furthermore, software processes are concerned with “soft” aspects, e.g. teams, skills, communication and roles. Similarly in MDD there are technical aspects, such as automation, through e.g. generators or model transformations, that can predefine fine-grained sequences of activities between manual and automated tasks. Further, just as processes, also MDD can, due to concepts such as abstraction, affect soft aspects as the skill set of developers. Thus, both MDD and software development process can affect similar and related aspects of development. However, the question how these interrelations can lead to mutual constraints or to synergy effects is only rarely studied.

Research on the keyword *software process tailoring* (i.e. on the question how processes need to be customized due to environmental factors) is rarely concerned with MDD directly. However, as summarized in the survey of Kalus et al., programming languages and tool infrastructure are known to be criteria for process tailoring [4].

A bit more attention was drawn to this topic in the modeling community [2]. For example, Aranda et al. found out that the division of labor changed within General Motors when MDD was introduced [1]. Further, Heijstek et al. learned in a case study with at an international IT service provider that MDD usage can lead to an increased need for collective code ownership. Further they observed changes in communication, required skills, and tooling [3].

As our survey on processes that had been adapted for MDD has shown, influences between both MDD and software process can be most diverse [2]. While in some cases mainly roles had been adapted within the processes, in other cases the structure of phases or sprints of the process changed. For example, Loniewski et al. describe an adaptation of OpenUP (a variant of RUP), where new roles, such as “model analyst” had been introduced [6]. In contrast, Kulkarni et al. adapt SCRUM by adding a meta sprint for long running tasks [5].

Summing up, there is an awareness that software processes might need or have the potential to adapt, when MDD is applied. However, there is still much research to be done before we fully understand this interrelation.

3 Research Perspectives

This first workshop on model-driven development processes and practices aims at calling attention to the question how development processes can be integrated with an MDD approach.

First of all, this is an empirical research question. Therefore, the **MD^{2}P^2** workshop provides a forum for researchers and practitioners to exchange and discuss experiences on how the use of MDD affects the development process in practice. For example, there is so far not much knowledge about how MDD is affected by a maturing development processes. Further, the workshop tries to uncover what happens to the development processes in practice, when MDD is introduced. Empirical data or case studies from practice can help to approach diverse questions:

- Which stakeholders are involved in modeling tasks & which stakeholders are not affected by the integration of MDD? – These questions are interesting for two reasons. On the short term, it might be less expensive to train a smaller set of developers and stakeholders to the new technologies. On the long term, however, only stakeholders who are involved in modeling also have potential to benefit from the higher level of abstraction and the improved automation.
- Which (modeling) artifacts are subject to quality assurance activities, e.g. reviews or testing? – MDD defines the set of artifacts that represent the system under development. This includes the potential for quality assurance, when certain checks can be performed earlier in development. However, it can also change the skills that are required for e.g. reviews and with it the roles in the process.
- Are development process phases adapted? Does the number or frequency of iterations change? – The structure of process phases or the iterations are essential for many processes. Changing them might have a major impact on the characteristics of the process. Since there are hints that such changes sometimes occur due to MDD, the question becomes pressing, when and why these adaptations happen.
- Is there empirical evidence that the intended MDD effects occur, e.g. does front-loading actually reduce the number of errors in later phases? – As indicated above, MDD is associated with certain hopes. However, only empirical evidence can prove whether these goals are reached.

In addition to the empirical perspective, it is also necessary to approach the topic more theoretically. Such research can cover systematic investigations of the mechanisms that drive impacts from MDD approaches on development processes or in turn define constraints on MDD approaches that are implicitly defined by development processes. Based on these investigations, researchers might foster the success of MDD, e.g. by providing guidelines, methods, or tools that support practitioners in reusing or adapting development processes when MDD is introduced are required.

Investigations on what aspects of a process are affected by MDD can concern diverse aspects, e.g.: How are different stakeholders integrated in the modeling activities? Can modeling tasks be split over multiple roles and phases? What is the effect of automated verification methods on testing methodologies and philosophies defined in development processes (e.g. in test-driven development processes)? Is there a need to adapt test and quality assurance activities in development processes, such that the various modeling artifacts are covered appropriately? When is it necessary or beneficial to adapt the number of development process phases or to change the frequency of iterations?

One motivation to answer these questions, is to identify combinations of MDD and processes that do not fit together, i.e. where the benefits of the process or of the MDD approach cannot fully be used. This is a first step towards tailoring MDD or processes, such that a better fit can be reached. A second motivation is that it might be possible to achieve synergies between MDD and development processes. Associated questions are: How can the combination with an MDD approach increase (or decrease) the benefits of a process? How can the choice or adaptation of a process increase (or decrease) the benefits of an MDD approach?

Ideally this research leads to guidelines and methods that can support practitioners in reusing or adapting development processes when MDD is used. Furthermore, researchers might come up with tool support for the integration of MDD into a given process. Finally, as both the development processes and MDD evolve and mature, there is the question how synergistic effects can be maintained over time, i.e. how co-evolution or co-maturation of MDD and development processes can be supported.

To summarize, there are many open questions related to the combination of MDD and software processes. With this workshop we want to strongly encourage more researchers to contribute to the investigation of these questions.

4 First Edition of MD^{2}P^2

For this first edition of the workshop, we received eight papers, of which four have been accepted for inclusion in the proceedings. The accepted papers can be split in two groups:

The first group of accepted papers deals with the question how processes look like that are used for the development of MDD technologies, such as transformations or DSLs. Barisic et al. present of the case study of FlowSL, how a usability concerns can be considered throughout during DSL development throughout an agile development approach where MDD tools are used [7]. Silva et al. present a survey on model transformation development approaches, discuss what phases of the development process are supported by the different approaches and present several lessons learned [10].

The second group of accepted papers provide an empirical view on processes that are used in combination with MDD. Hovsepyan et al. present an experience report on the use of MDD in development of safety-critical avionic systems [8]. Knauss et al. investigated a case study in the automotive domain and identify challenges for the use of MDD in multi-tier automotive ecosystems [9].

We hope that the workshop will help researchers and practitioners to build up a community that shares data and experience. In closing, we would like to thank all authors papers and reviewers, for their contributions, the effort they invested, and for making this workshop possible.

References

1. Aranda, J., Borici, A., Damian, D.: Transitioning to model-driven development: What is revolutionary, what remains the same? In: Model Driven Engineering Languages and Systems, LNCS, vol. 7590, pp. 692-708, Springer, (2012)
2. Hebig, R., Bendraou, R.: On the need to study the impact of model driven engineering on software processes. In: Proceedings of the 2014 International Conference on Software and System Process, pp. 164-168. ACM, New York (2014)
3. Heijstek, W., Chaudron, M. R.: The Impact of Model Driven Development on the Software Architecture Process. In: Software Engineering and Advanced Applications (SEAA) (2010)
4. Kalus, G., Kuhrmann, M.: Criteria for software process tailoring: a systematic review. In: Proceedings of the 2013 International Conference on Software and System Process, pp. 171-180, ACM, (2013)
5. Kulkarni, V., Barat, S., Ramteerthkar, U.: Early experience with agile methodology in a model-driven approach. In: Proceedings of the 14th international conference on Model driven engineering languages and systems (MODELS'11), eds. Whittle, J., Clark, T., Kühne, T., pp. 578 -590, Springer, Berlin, Heidelberg (2011)
6. Loniewski, G., Armesto, A., Insfran, E.: An agile method for model-driven requirements engineering. In: The Sixth International Conference on Software Engineering Advances (ICSEA 2011), pp. 570-575 (2011)

7. Barisic, A., Amaral, V., Goulo, M., Aguiar, A.: Introducing usability concerns early in the DSL development cycle: FlowSL experience report. In: Proceedings of the 1st International Workshop in Model-Driven Development Processes and Practices (MD^2P^2), pp. 8-17 (2014)
8. Hovsepyan, A., Van Landuyt, D., Op de Beeck, S., Michiels, S., Joosen, W., Rangel, G., Fernandez Briones, J., Depauw, J.: Model-Driven Software Development of Safety-Critical Avionics Systems: an Experience Report. In: Proceedings of the 1st International Workshop in Model-Driven Development Processes and Practices (MD^2P^2), pp. 28-37 (2014)
9. Knauss, E., Damian, D.: Towards Enabling Cross-Organizational Modeling in Automotive Ecosystems. In: Proceedings of the 1st International Workshop in Model-Driven Development Processes and Practices (MD^2P^2), pp. 38-47 (2014)
10. Silva, G. C., Rose, L., Calinescu, R.: A Qualitative Study of Model Transformation Development Approaches: Supporting Novice Developers. In: Proceedings of the 1st International Workshop in Model-Driven Development Processes and Practices (MD^2P^2), pp. 18-27 (2014)