How MAD are we? Empirical Evidence for Model-driven Agile Development

Sebastian Hansson, Yu Zhao, and Håkan Burden

Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
Sweden
gussebasha@student.gu.se, guszhaoyu@student.gu.se, burden@cse.gu.se

Abstract. Since the launch of the Agile Manifesto there has been numerous propositions in how to combine agile practices with Model-driven Development. Ideally the combination would give the benefits of agile – e.g. rapid response to changes and shorter lead times – with the promises of Model-driven development – such as high-level designs and automation. A commonality among the proposals is that they lack in empirical evaluation. Our contribution is a systematic literature review to find out to what experiences there are of Model-driven Agile Development, MAD, from an empirical context. Among our conslusions is that MAD is sitll an immature research area and that more experience reports from industry are needed before we can claim to have understood the possibilities and drawbacks of MAD.

Keywords: Systematic literature review, Agile practices, Model-driven development

1 Introduction

Model-driven Agile Development, MAD, which aims to combine the benefits of agile practices with the positives of Model-driven development, MDD, has been a dream for many years. Mellor et. al. [15, 14] discuss the possibilities of combining agile and MDA[16], proposing that many of the agile practices are just as suitable for MDA. For MDA, executable models is a key feature for successful MAD since they can replace code and use expressions closer to the language of the customer. The idea that executable models serve as a better communication media than code is supported by R. Gomes et al. [7] and Stahl and Völter [21]. In addition, the latter claim that software is developed faster since tedious, recurring implementation tasks are automated. Transformations will also ensure that changes in the problem formulation are consistently propagated through the solution much faster than in a code-centric context. Selic proposes the usage of heterogeneous models by combining high-level modeling languages with detail-level action languages, so that the benefits of MDD can be extended "*to the full*

development cycle, including its use in agile development techniques based on multiple iterations" [20]. Rumpe argues that MAD will be successful since the models will enable static analysis, rapid prototyping, code generation, automated tests, refactoring and transformation as well as documentation [19]. Kaim, et al. also argue that efficient model transformations are cruical if MAD is to be achieved [21]. All contributions have in common that there is no concrete empirical data backing the claims.

1.1 Research Topic

To see what empirical evidence there is for MAD we decided to conduct a systematic literature review [10]. The resulting contribution delivers an initial analysis from the collected publications in order to answer our research questions

RQ1: What is the state of the art of MAD from an empirical point of view?
RQ2: What is lacking in the empirical literature regarding MAD?

Our results tell us that MAD is still too immature to claim wide-spread success or a state of the art over another and that we need more reports on industrial experiences of MAD in order to close the current gap in the literature.

1.2 Overview

The rest of our contribution is structured as follows; in the next section we will explain our methodology, including the identification and analysis of relevant publications. In section 3 we present our findings based on the papers that passed both our inclusion and exclusion criteria as well as the quality criteria. Section 4 synthesises the findings in relation to the research questions while threats to validity are given in section 5. Finally, in section 6 we conclude and propose future work.

2 Collecting and Analysing the Publications

Following the guidelines presented by Kitchenham et al. [10] we have performed a systematic literature review in order to answer our two research questions.

2.1 Collecting the Relevant Publications

We used three different digital libraries – IEEE explore, ACM digital library and the SpringerLink library applying the search string ("agile" AND "model") to the listed titles. Publications explicitly mentioning MAD in an empirical setting where included while publications discussing the combination of agile development and MDD as a theoretical contribution were excluded. Only contributions published after 2001, after the Agile manifesto was launched, were included while those not written in English were excluded. If a publication was published in multiple forms we selected the most comprehensive one, meaning that journals

Table 1. Included papers mapped to bibliographic reference and publication venue – **J**ournal, **C**onference and **W**orkshop.

Paper Nr	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13
Bibl. ref.	[25]	[3]	[11]	[18]	[12]	[4]	[9]	[5]	[17]	[22]	[8]	[23]	[24]
Venue	J	C	C	J	C	C	C	C	C	W	C	C	J

had precedence over book chapters, chapters over conference preceedings which in turn had precedence over workshops. Only peer reviewed publications were considered.

The publications were then included or excluded in a three-step process. First, titles and abstracts were analysed according to our inclusion/exclusion criteria. A total of 291 publications were found – 86 publications originated from Springerlink, ACM Digital Library yielded 84 publications while IEEE Xplore contained 121 publications. Second, the introduction and conclusion sections of the included papers were studied to further refine the selection. This resulted in 78 included publications. Third, the criteria were applied to the full publications. After removing eleven duplicates and five contributions that were published at multiple venues we ended up with thirteen papers that mention empirically documented cases of MAD. We then went through all references in the thirteen included papers, without finding any secondary studies that matched our inclusion and exclusion criteria. The included publications are presented in Table 1 together with their respective bibliographic reference number and type of publication venue.

2.2 Quality Assessment of the Selected Publications

To ensure that we can get quality data from the selected papers we need sufficient contextual and methodological information [6]. We therefore defined seven quality criteria that represented 1) the aim of using MAD, 2) the strategy for achieving MAD, 3) which agile practices that were used, 4) which MDD practices that were used, 5) details regarding the team and the project, 6) in what kind of domain MAD was adopted, and 7) the impact in terms of MAD leading to success or failure. The outcome of applying the quality criteria to the selected papers is found in Table 2. Papers P4, P6, P7, P8, P10, and P12 fail to report sufficient information about how MAD was applied, making it difficult to draw parallels between the publications or synthesise empirical evidence of MAD. The publication with insufficent context will not be a part of the results.

3 Results

In this section we present our findings based on the seven publications that passed both the inclusion, exclusion and quality criteria. The focus in this section is on the strategies that were employed, the challenges in adopting MAD and

Table 2. Results of applying the quality criteria to the included publications

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13
Aim of MAD	X	X	X	X	X	X	X	X	X	X	X	X	X
MAD strategy	X	X	X	X	X	X	X	X	X	X	X	X	X
Agile practice	X	X	X	X	X	X	X	–	X	X	X	X	X
MDD practice	X	X	X	X	X	X	X	X	X	X	X	X	X
Team/project	X	X	X	–	X	–	–	–	X	–	X	–	X
Domain	X	X	X	X	X	X	X	X	X	X	X	X	X
Impact	X	X	X	X	X	X	–	–	X	X	X	X	X

the impact of MAD while an overview of the quality criteria for Aim, Agile and MDD practices, Team and Project as well as Domain is found in Table 3.

3.1 Overview

As seen in Table 3, publication P9 deploys Scrum, Model-Driven Architecture (MDA [16]) as well as feature and mockup models to improve the handling of variability in their accounting system. The actual implementaion was done by an existing web design team.

Publication P1 is interesting in two different comparisons; First, P1 and P13 have both applied MAD within the telecom industry – which is not surprising since Zhang is an author of both publications. However, the publications differ in their motivations for MAD as well as how MAD was implemented in terms of agile and MDD practices. The second comparison is between publications P1 and P3 which are similar in context – aim and chosen practices – but not in domain. From the publications that passed the three sets of inclusion, exclusion and quality criteria there is no common theme in how and why MAD was adapted in relation to contextual factors.

3.2 MAD Strategies

The strategy used to achieve agile model-driven development differs among the selected publications. P9 and P11 suggest to build mockup models, as a mean of communication and for easy requirement gathering and handling requirement changes. P11 promotes an agile way of building models, with specific tools to easily transform the models under development. P1 suggests using iterations for MDD and P5 wants sprints and iterations but to still have a high level design. P13's strategy is to have both agility and quality built into their development process and P3 suggests a modified agile method and presents a tailored approach to address the need of managing evolution using model-based techniques. P11 and P13 use a Test-driven development approach combined with modeling. Our findings tell us that the common trend among the reviewed papers is to include a more agile way into existing model-driven engineering practices. Especially because of agile development's advantages regarding rapid response to change

Table 3. The collected data for the quality criteria

	Aim	Agile practices	MDD practices	Team/Project	Domain
P1	Shorten lead time	Scrum, XP, Iterations or incremental development	UML, Code generation	New team, no prior experience of agile or MDD	Telecom
P2	Respond to change	Iterations or incremental development	UML	Changed over time	Legacy system
P3	Shorten lead time	Scrum, XP, Iterations or incremental development, Feature-driven	Feature models	New team	Database
P5	Demanded by domain	Scrum, XP	UML	Architect, domain expert, coders, tester and client	Web application
P9	Improve variability	Scrum	MDA, Feature models, Mockup models	Web design team	Accounting system
P11	Involve stakeholders	Scrum, XP, Test-driven development	Mockup models	10 developers	Customer satisfaction system
P13	Respond to change, Involve stakeholders, Improve productivity and quality	Test-driven development	UML	60 people divided into sub-teams	Telecom

and close stakeholder involvement – a claim of MDD but where the included publications see agile more successful.

The most interesting and comprehensive strategy we found is in P1. They use a methodology called System Level Agile Process (SLAP). SLAP is a Scrum-based agile methodology, constructed by Motorola that includes XP practices. In SLAP the software lifecycle is split into short iterations, where each iteration includes three sprints – requirements, architecture and development – and then system integration feature testing. They conclude that "*From MDD perspective the key to success is to maximize automation using the MDD tools chain to enable mistake-free (high-quality) development and significant productivity in-*

crease" and "*From the Agile perspective, the key is to efficiently achieve end-to-end iterations, from system engineering all the way to system testing. This requires streamlining different process activities such as system engineering, development, and testing*".

3.3 Challenges

Zhang et al. state in P1 that MAD "*is still relatively new in real software development. The learning curve is sharp for any new organization to adopt due to process, culture, methodology, and other related changes. Thus, adopting a new agile MDD process is not likely to produce a short-term benefit. But, for the long-term, it's ultimately worth it for large projects with multiple releases*". P12 also mentions the steep learning curve, this time for web modelling tools.

To provide lightweight-agility in development seems to be a solution for Kulkarni et al. who in P3 argues that traditional agile development is not suited for larger teams or projects.

3.4 Impact

When it comes to impact, publications P1, P2, P3 and P9 state that they successfully combined agile practices and MDD. P1 and P2 argue that the combination could be useful for future development teams, due to the fact that the learning curve is steep it is not likely to produce short-time benefits. P2 and P13 noticed an increase in commitment, quality and productivity from the developers – despite having different aims and strategies – while P11 report on improvements regarding both time and satisfaction. P5 state that their approach would not be effective on large projects while P13 states that their approach could be beneficial and possible for any-size projects. Still, the overall assessment is that the authors fail to provide detailed descriptions on what was successful and why.

4 Synthesis

The data drawn from seven publications can now be further refined into a synthesis answering the research questions that motivated our systematic literature review.

4.1 What is the State of the Art of MAD from an Empirical Point of View?

Based on our findings, we can conclude that the area around MAD is immature when it comes to empirical evidence and industrial experiences. To answer this research question, it is not enough to look at the theoretical part, we need evidence to prove any best practice or state of the art, though the evidence seems to be lacking. The strategies used among the papers we found is in many cases also contradictory to each other – providing information about different

domains, goals and strategies, making it difficult to claim any best practice or success over someone else. Though the most common goals seem to be the agile value of rapid response to change and contact to external stakeholders. Another commonality is that a majority of the included publications want to include agility into their MDD practices – and not extend agile practices with MDD features – a situation that is seen in other areas since agile is the trend in current software development.

Only when the area is more mature will it be possible to claim a state of the art regarding the combinations of agile practices and MDD. Though claiming that the area is not yet mature does not mean it is not possible to adapt. As Zhang and Patel say in P1, MAD requires investments in both learning and technology and the success will not be immediate. Other authors such as Lee et al. argue in P5 that a pure agile methodology or a pure MDD approach is not longer enough "*we can not apply agile processes to web application development directly. UML is not sufficient for modeling the navigation of Web applications*".

4.2 What is Lacking in the Empirical Literature Regarding MAD?

After removing articles based on our inclusion and exclusion criteria we only had thirteen papers left mentioning any concrete empirical evidence. This is a rather low value for an systematic literature review, which in turn suggests that the area is still immature. According to the publication venues, this conclusion is supported by the fact that there are only three articles published in journals, suggesting again that the more mature and empirical research contributions regarding MAD are still to be written. The search for secondary studies did not result in any additional publications that matched our inclusion, exclusion nor quality criteria – again suggesting that more research in the area is needed.

There are authors (e.g. Ambler [1, 2]) who discuss MAD from a more abstract empirical setting, drawing from their own experiences without giving specific details about a certain project or case and therefore lacking information how a certain company or a specific team adopted MAD into their existing context. Six out of the seven papers that passed our quality criteria also fail to deliver detailed information how they performed MAD. They barely mention their team-setup, what practices they combined and what tools they used. Only in P5 is the team described in detail with information about team members and roles.

Future empirical publications would contribute substantially to our understanding of MAD if they provided information about:

Teams/project: What was the size of the team? Which responsibilities did the team members have? Did the team members ahve any earlier experience of agile or MDD? For how long did the project run?

Used practices: Which agile and MDD practices were used?

Strategy: How were the agile and MDD practices combined? Were the practices easy to combine? Which where the challenges of introducing MAD?

Tools: Which tools were used to achieve MAD? Where specific tools for MAD develop or could off-the-shelf tools be used?

Impact: Was MAD a success or a failure? What succeeded/failed and why?

As Ambler himself indicates "*The use of Agile methodology in model-driven development is not prevalent yet, except tailored Agile approaches, such as Agile model driven development*" [1]. Reza Matinnejad comes to the conclusion that MAD "*is a promising research context and a great practical concept, but it is not mature enough and is still in its infancy*" [13].

5 Threats to Validity

Our results may face validity issues as our search was restricted to ACM digital library, Springerlink and IEEE explore. There are other digital libraries which are widely used in the software engineering field. However, these three libraries include the journals and conferences with the highest impact factor which should imply that they represent the most mature research. Another threat to the validity is our search string, as we did a title search on the words *agile* and *model*, it is possible that we have missed publications discussing the combination of agile and model-driven development while not explicitly mentioning the fact in their title. We tried to mitigate this by searching for secondary studies among the references in our primary studies but after applying our inclusion and exclusion criteria we could not find any additional publications. This gives us some confidence that we could not have missed many articles in our initial search. Our inclusion and exclusion criteria and our quality assessment can also be a threat to validity as we were searching for actual empirical cases where the authors mention certain key information that we as researchers see as empirical evidence, such as teams, project success, MDD and agile practices among others.

There are many other papers that discuss how and why you should combine the two development methods but they lack the key information, thus not being identified as relevant publications for our systematic literature review. The publications not passing our criteria might have important information regarding MAD, but could not be included in our data extraction and synthesis. Also, the publications that passed all three sets of criteria could still contribute data of poor quality as they might pass our criteria but fail to deliver detailed information.

6 Conclusions and Future Work

In this paper we have presented the results of a systematic literature review about empirical evidence on Model-Driven Agile Development. In the seven papers which passed all our quality criteria, the authors wanted to combine agile methods and MDD in a way that could draw benefits from both worlds and at the same time avoid their respective shortcomings. In the result section, we have presented different integrations of agile and MDD practices for various purposes. There are multiple authors discussing different theoretical approaches for MAD but just a handful publications describing detailed information how a company

or a team adopted such a practice into their existing context. Six out of the seven publications that passed our quality criteria also fail to deliver comprehensive information how they performed MAD. They barely mention their team-setup, what practices they combined and what tools they used.

Discovering that the empirical contributions regarding MAD are immature gives plenty of opportunities for future research, especially when it comes to detailed experience reports. The empirical experiences regarding MAD are still few and lacking details. We aim to extend this study by including more publication databases and by providing a more comprehensive analysis of the found publications. Such an anlsyis would also seek to map the theoretical proposals to the empirical evidence for MAD.

References

1. Ambler, S.: Agile/Lean Documentation: Strategies for Agile Software Development. www.agilemodeling.com/essays/agileDocumentation.htm, accessed June 19th 2013
2. Ambler, S.: Agile Modeling: Effective Practices for eXtreme Programming and the Unified Process. John Wiley & Sons, Inc., New York, NY, USA (2002)
3. Burkhardt, R., Gruhn, V.: Agile Software Engineering: A New System for an Expanding Business Model at SCHUFA. In: 5th Annual International Conference on Object-Oriented and Internet-Based Technologies, Concepts and Applications for a Networked World. pp. 201–215. Lecture Notes in Computer Science, Springer Berlin Heidelberg, Erfurt, Germany (September 2004)
4. Cardoso, N., Rodrigues, P., Ribeiro, O., Cabral, J., Monteiro, J., Mendes, J., Tavares, A.: An agile software product line model-driven design environment for video surveillance systems. In: 2012 IEEE 17th Conference on Emerging Technologies Factory Automation (ETFA). pp. 1–8 (Sept 2012)
5. Djanatliev, A., Dulz, W., German, R., Schneider, V.: Veritas - a versatile modeling environment for test-driven agile simulation. In: Proceedings of the 2011 Winter Simulation Conference (WSC). pp. 3657–3666 (December 2011)
6. Dybå, T., Sjøberg, D.I.K., Cruzes, D.S.: What works for whom, where, when, and why?: on the role of context in empirical software engineering. In: Runeson, P., Höst, M., Mendes, E., Andrews, A.A., Harrison, R. (eds.) ESEM. pp. 19–28. ACM (2012)
7. Gomes, R., Rivera, G., Willrich, R., Lima, C., Courtiat, J.: A Loosely Coupled Integration Environment for Collaborative Applications. Systems and Humans, IEEE Transactions on Systems, Man and Cybernetics, Part A 41(5), 905–916 (September 2011)
8. Grigera, J., Rivero, J.M., Robles Luna, E., Giacosa, F., Rossi, G.: From Requirements to Web Applications in an Agile Model-Driven Approach. In: Brambilla, M., Tokuda, T., Tolksdorf, R. (eds.) ICWE 2012, 12th International Conference on Web Engineering, Lecture Notes in Computer Science, vol. 7387, pp. 200–214. Springer Berlin Heidelberg (2012)
9. Huang, Y.C., Chu, C.P.: Legacy System User Interface Reengineering Based on the Agile Model Driven Approach. In: Qian, Z., Cao, L., Su, W., Wang, T., Yang, H. (eds.) Recent Advances in Computer Science and Information Engineering, Lecture Notes in Electrical Engineering, vol. 125, pp. 309–314. Springer Berlin Heidelberg (2012)

10. Kitchenham, B., Charters, S.: Guidelines for performing Systematic Literature Reviews in Software Engineering. Tech. Rep. EBSE 2007-001, Keele University and Durham University Joint Report (2007)
11. Kulkarni, V., Barat, S., Ramteerthkar, U.: Early Experience with Agile Methodology in a Model-Driven Approach. In: MODELS 2011, 14th International Conference on Model-Driven Engineering Languages and Systems. pp. 578–590. Lecture Notes in Computer Science, Springer Berlin Heidelberg, Wellington, New Zealand (October 2011)
12. Lee, W., Park, S., Lee, K., Lee, C., Lee, B., Jung, W., Kim, T., Kim, H., Wu, C.: Agile development of Web application by supporting process execution and extended UML model. In: APSEC '05, 12th Asia-Pacific Software Engineering Conference (December 2005)
13. Matinnejad, R.: Agile Model Driven Development: An Intelligent Compromise. In: SERA 2011, 9th International Conference on Software Engineering Research, Management and Applications. pp. 197–202. IEEE Computer Society, Baltimore, MD, USA (2011)
14. Mellor, S.J., Balcer, M.: Executable UML: A Foundation for Model-Driven Architectures. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA (2002)
15. Mellor, S.J., Kendall, S., Uhl, A., Weise, D.: MDA Distilled. Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA (2004)
16. Miller, J., Mukerji, J.: MDA Guide Version 1.0.1. Tech. rep., Object Management Group (2003)
17. Pohjalainen, P.: Bottom-up Modeling for a Software Product Line: An Experience Report on Agile Modeling of Governmental Mobile Networks. In: 15th International Software Product Line Conference (SPLC). pp. 323–332 (August 2011)
18. Ramos, A., Ferreira, J., Barcelo, J.: LITHE: An Agile Methodology for Human-Centric Model-Based Systems Engineering. Systems, Man, and Cybernetics: Systems, IEEE Transactions on 43(3), 504–521 (May 2013)
19. Rumpe, B.: Agile modeling with the UML. In: Wirsing, M., Knapp, A., Balsamo, S. (eds.) Radical Innovations of Software and Systems Engineering in the Future, pp. 297–309. No. 2941 in Lecture Notes in Computer Science, Springer Berlin Heidelberg (Jan 2004)
20. Selic, B.: Model-Driven Development: Its Essence and Opportunities. In: ISORC'06 Proceedings of the Ninth IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing. pp. 313–319. IEEE Computer Society, Gyeongju, South Korea (April 2006)
21. Stahl, T., Völter, M.: Model-Driven Software Development: Technology, Engineering, Management. John Wiley & Sons Inc. (2005)
22. Urli, S., Blay-Fornarino, M., Collet, P., Mosser, S.: Using Composite Feature Models to Support Agile Software Product Line Evolution. In: Proceedings of the 6th International Workshop on Models and Evolution. pp. 21–26. ME'12, ACM (2012)
23. Xufeng, L., Marmaridis, I., Ginige, A.: Facilitating Agile Model Driven Development and End-User Development for EvolvingWeb-based Workflow Applications. In: ICEBE 2007, IEEE International Conference on e-Business Engineering. pp. 231–238 (October 2007)
24. Zhang, Y.: Test-driven modeling for model-driven development. Software, IEEE 21(5), 80–86 (September 2004)
25. Zhang, Y., Patel, S.: Agile Model-Driven Development in Practice. IEEE Software (2011)