MUNICH AGILE MBSE CONCEPT (MAGIC)

Salehi, Vahid; Wang, Shirui

Munich University of Applied Sciences

ABSTRACT

Model-based systems engineering (MBSE) is well-known in gaining the control over the complexity of systems and the development processes, while agile is a project management methodology originally from software development that uses short development cycles to focus on continuous improvement in the development of a product or service. In this paper, we adopt the concept of agile into MBSE and then proposed the new approach - Munich Agile MBSE Concept (MAGIC). The highlights of the MAGIC approach can be concluded as 1) the requirements which have been defined in the first stage will be examined and traced at each following stages; 2) communication between every 2 stages always exists in order to have a close connection during each system development phase; 3) the idea of Industry 4.0 has been included and reflected to achieve automation and data exchange with manufacturing technologies; 4) the concept of IOT (Internet of Things) is also considered when it comes to the usage and service of the system to satisfy the customer's needs; 5) the whole spirit of agile is reflected as the iterative and incremental design and development

Keywords: Systems Engineering (SE), Product modelling / models, Virtual Engineering (VE)

Contact:
Salehi, Vahid
Munich University of Applied Sciences
Mechatronics
Germany
salehi-d@hm.edu

Cite this article: Salehi, V., Wang, S. (2019) 'Munich Agile MBSE Concept (MAGIC)', in *Proceedings of the 22nd International Conference on Engineering Design (ICED19),* Delft, The Netherlands, 5-8 August 2019. DOI:10.1017/dsi.2019.377

1 INTRODUCTION

Model-based systems engineering (MBSE) is well-known in gaining control over the complexity of systems and the development processes (Hooshmand *et al*., 2017). The basic idea of MBSE is to formalize the system description as well as to connect the relevant information - needed for the creation of various artifacts during the system development - in a system model (Kaufmann and Schuler, 2016). Furthermore, a standardized and formalized language like OMG Systems Modeling Language (OMG SysML) is necessary for the formal definition of the system (Friedenthal *et al*., 2015).
However, traditional MBSE approach usually works professionally for the project on a relatively large scale (Salehi *et al*., 2018). Therefore, in most of the cases, MBSE approach consumes a lot of time. Additionally, the entire process chain with production and manufacturing stage as well as the usage and service stage are usually neglected. Moreover, as increasing implementation and integration have been applied to MBSE, it is noticeable that the requirement which builds up the foundation of the whole project should be tracked at the entire lifecycle of the development chain. Accordingly, to pursue a new approach, in many cases, seems necessary.
Agile is a project management methodology that uses short development cycles called "sprints" to focus on continuous improvement in the development of a product or service (Cao and Ramesh, 2007). Although agile methods were developed by and for software developers, the discourse on their relevance for other types of development projects has been intensifying the past years and receive increasing attention in the development of mechatronic products, however as the boundary conditions are different (e.g. virtual vs. physical prototypes) modified forms established themselves (Cao and Ramesh, 2007). Therefore, in this paper, we intended to adopt the concept of agile from software development and combined with a MBSE approach into cyber-physic product design. A new approach - Munich Agile MBSE Concept (MAGIC) has been proposed. MAGIC concerns about the requirement at each stage of the product development lifecycle. Moreover, besides the traditional development stages such as system functional design, system architecture design, and system verification and validation, production stage, product usage, and service stage is also been considered in the entire lifecycle. Different from typical V-model used widely in systems engineering, MAGIC has the shape of infinity loop to demonstrate that process chain of each stage does not end when the product has been delivered and applied by the customer. On the contrary, it will keep going back to the first requirement stage when the user has new demand for the product.

2 STATE OF THE ART

2.1 Agile product development

To assure that evolving customer requirements can be met and costs from late changes can be avoided, companies increasingly develop products on the basis of agile product development processes. Agile processes are characterized by a sequence of iterative sprints, in which parallelized activities are undertaken. At the end of each sprint, an incremental prototype of the product or various subsystems is created so that during the entire process the degree of maturity of the product is continuously and successively increased (Lindlöf and Furuhjelm, 2018). Each incremental prototype increases the functionality or features or confidence of the previous prototype generation. This enables an early evaluation of subsystems or components. The incremental development allows also for early feedback of customers. Agile development teams need to possess all the necessary skills to develop and validate the product. Popular agile approaches are briefly listed as a knowledge base for this goal:

- **Design Thinking** - which focusses on gaining a deepest possible understanding of the end user's needs, and aims at a rapid conception of solutions, an implementation in prototypes and early testing for further knowledge gain (Brown and Wyatt, 2010).
- **Human Centred Design** - which works with the users to gain a holistic understanding of the design problems that are addressed to develop systems, which are useful, usable and desirable for humans with the consistent focus on higher levels of humans' needs (Zhang and Dong, 2009).
- **ASD** - Agile Systems Design is a holistic, structured approach for the agile development of mechatronic systems, the associated product strategy, validation systems, and production systems, consisting of principles, methods, and processes of PGE (Product Generation Engineering). It

https://doi.org/10.1017/dsi.2019.377 Published online by Cambridge University Press

focuses on the consequent integration of PGE into the development process and creates a situation-specific balance between structuring and agile elements (Albers *et al*., 2018).

DevOps is one of the most famous agile approaches and combines software development (Dev) with information technology operations (Ops). The goal of DevOps is to shorten the systems development life cycle while also delivering features, fixes, and updates frequently in close alignment with business objectives. The DevOps approach is to include automation and event monitoring at all steps of the software build and is the combination of cultural philosophies, practices, and tools that increase an organization's ability to deliver applications and services at high velocity: evolving and improving products at a faster pace than organizations using traditional software development and infrastructure management processes. This speed enables organizations to better serve their customers and compete more effectively in the market (Bang *et al*., 2013).

2.2 Model-based systems engineering (MBSE) with V-model

INCOSE (2007) defines MBSE as "the formalized application of modelling to support system requirements, design, analysis, verification and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases. MBSE is part of a long-term trend toward model-centric approaches adopted by other engineering disciplines, including mechanical, electrical and software". MBSE is not only considering the different needs of different stakeholders (Anderl *et al*., 2012) but also supporting the entire development process (Weilkins, 2008). In this section, we will make a short evaluation and summary of 4 widely use V-model versions to demonstrate the different aspect of product or system development methods due to changes in technology, organizational and social environment. Before we start the evaluation, a few questions regarding the characteristic properties need to be elaborated and answered as shown in table 1(Gräßler *et al*., 2018). The main aspects we are going to explore in existing V-model are marked in grey from question 7 to 10.

Table 1. Overview of characteristic properties related questions

1. Focused fields of application - Q: What is the field of application for the reference model? Are there special focuses or users?
2. Sequential or iterative approach - Q: Is the development process run in a sequential or iterative order? Does the model point out the possibility of iterations?
3. Verification and validation - Q: Is the assurance of properties represented as forward or backward process?
4. Life cycle representation - Q: Does the approach refer to or include life cycle concepts?
5. Decomposition into System levels - Q: Does the model refer to different system levels? Are there hints at the decomposition of the system?
6. Adaptability for domains - Q: Is the approach adaptable to the different engineering disciplines or usable just for overall systems?
7. The requirement is traceable in each stage - Q: Is the requirement defined in the first stage will be traced at each following stage?
8. Consider agile concept - Q: Is the V-model adopt any agile concept in product development process?
9. Consider the production and test stage - Q: Does the V-model consider the production and test stage for the product/system?
10. Consider the usage and service stage - Q: Does the V-model consider the usage and service stage for the product/system?

Figure 1 (a) is the V-model of the VDI 2206 (VDI, 2004) guideline fromthe year 2004 which offered a comprehensive model for engineering mechatronic systems. This V-model basically divides the development process into three sections: 1) The decomposition on the left side of the V-model

https://doi.org/10.1017/dsi.2019.377 Published online by Cambridge University Press

describesthe transformation of requirements, which are presented as an input, into a system design. 2) The engineering in different disciplines, the domain-specific design process. 3) Integration of the disciplines on the right side of the V-model during the system integration, verification, and validation. (Gräßler *et al*., 2018). Figure 1 (b) is the extended V-model from Eigner. He combines the model for the approach to development with a virtual and data-based approach (Eigner *et al*., 2012). One of the highlights is the integration of a PLM backbone and the addition of an additional right wing for virtual testing which testifies a strong focus on model-based approach and virtual, hybrid or physical testing. Moreover, on the left wing, he adopts RFLP (Requirement, Function, Logic solution element, physical element) approach to decompose the system design process. Comparable to Figure 1(a), in this V-model the "V" through the development process as well as the overall life cycle is shown.
Figure 1 (c) is the V-model by Bender 2005 (Bender, 2005). This V-model has divided the system into 3 level: system-level, subsystem-level, and component-level. It is noticeable here that the separation of the different domains does not take place at the top of the model, but already at the level of the subsystems. (Gräßler *et al*., 2018). Figure 1 (d) is the V-model of INCOSE. It is illustrated as a sequential method used to visualize various key areas for SE focus, particularly during the concept and development stages. The "V" highlights the need for continuous validation with the stakeholders, the need to define verification plans during requirements development, and the importance of continuous risk and opportunity assessment (Walden *et al*., 2015).

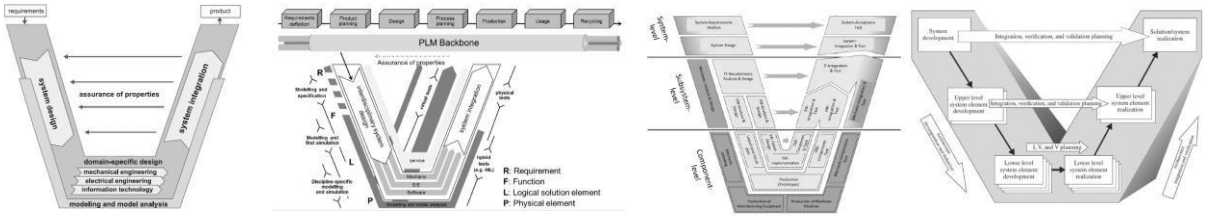

Figure 1. (a) V-model of the VDI 2206:2004 guideline "Design methodology for mechatronic systems" (VDI, 2004); (b) Extended V-model for model-based systems engineering (Eigner et al., 2012) (c) V-model by Bender 2005, translated from Bender (2005) (d) International Council on System Engineering (INCOSE) V-model (Walden et al., 2015)

All above-selected models are well established in their specific applications and frequently cited for scientific publications. As we summarized in table 2, these models have some features in common such as verification and validation process (Gräßler *et al*., 2018). However, these V-models also varies in many different characteristic properties. For example, VDI 2206:2004 focus on the application of the mechatronic system. On the other hand, Eigner's model is built up for MBSE and Bender' model focuses on embedded systems. In our previous work, we have also adopted the V-model in system development process of the digital factory and other use cases (Salehi *et al*., 2017).

Table 2. Overview of V-models by characteristic properties

Characteristic properties	a) VDI 2206: 2004	b) Eigner, Gilz, Zafirov	c) Bender	d) INCOSE SE4.0
1. Focused fields of application	Mechatronic systems	MBSE	Embedded Systems	Systems Engineering
2. Sequential or iterative approach	Hints for iterations	Iterative	Iterative	Sequential, straightforward process
3. Verification and validation	Right to left	Right to left	Left to right	Left to right
4. Lifecycle representation	No	Yes, backbone	No	No

https://doi.org/10.1017/dsi.2019.377 Published online by Cambridge University Press

5. Decomposition into System levels	Not illustrated	Not illustrated	Yes, three levels	Yes, three levels
6. Adaptability for domains	Yes, very generic	Yes	No, explicit description	Yes, very generic
7. The requirement is traceable in each stage	No	No	No	No
8. Consider agile concept	No	No	No	No
9. Consider the production and test stage	No	No	No	No
10. Consider the usage and service stage	No	No	No	No

As marked grey in table 2, we could see that current existing V-model versions didn't consider the agile concept, production and test stage as well as usage and service stage. Moreover, the requirement which is defined in the first stage is not able to trace in the following steps of the model. Additionally, when we use V-model as the method of MBSE, we could notice there are many challenges such as 1) non- uniform understanding of the overall system; 2) no consistency with regard to data, processes, and systems; 3) component-oriented thinking; 4) only discipline-specific models; 5) inadequate methodology; 6) lack of transparency and traceability of information; 7) a central system model for an interdisciplinary approach is not available; 8) there is a lack of formal and neutral languages, which are clear and interpretable. 9) Inconsistent relationship information. Therefore, in the following chapter, we adopt the concept of agile development method into MBSE and then proposed the new approach - Munich Agile MBSE Concept (MAGIC).

3 MUNICH AGILE MBSE CONCEPT (MAGIC)

MAGIC approach derived from the shape of DevOps infinity loop model and consists of 6 levels: system goal and requirement level, system functional level, systems architecture and logical level, system behavior analysis and verification and validation, system production and additive manufacturing and test, system usage and service level. In the following parts, we will describe each level in detail as well as the application of the MAGIC approach.

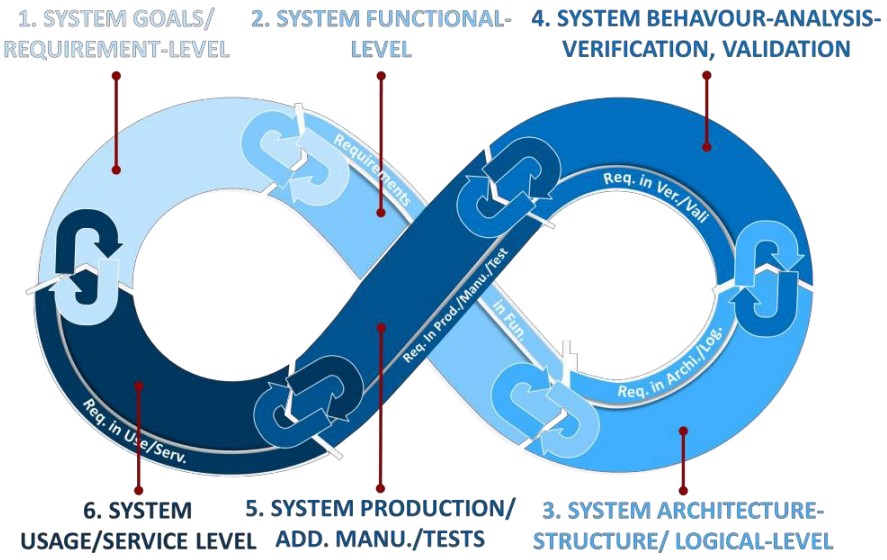

Figure 2. The MAGIC approach according to Salehi

https://doi.org/10.1017/dsi.2019.377 Published online by Cambridge University Press

1. **The system goal/requirement level** contains all the customers need and wishes related to the product and its requirements (Salehi *et al*., 2011). That means that the challenge during the first stages of the product development is to understand and to translate the customers need into technical requirements. A specification may refer to an explicit set of requirements to be satisfied by a material, design, product, or service. In the classical engineering approach, sets of requirements are used as inputs into the design stages of product development. Requirements are also an important input into the verification process since tests should trace back to specific requirements (Salehi *et al*., 2009).
2. **The system functional level** is a method for the understanding of the total product and to define the structure and behaviour of a system. A system is structurally represented by the functional or structural relationships between the individual components or sub-functions with regard to the technical requirements. With the creation of a functional system architecture, it is possible to identify the different functions in the entire system. Once the functional structure is illustrated, the technical system architecture is created with consideration to the functional structure.
3. **System Architecture-Structure/Logical level** includes components which have logical ports with the logical behaviour of a system. The technical system architecture of MAGIC approach describes a rough geometry, components design, and concepts. It defines "how" the functional requirement will be achieved — and there can be multiple ways to achieve the requirements. The required logical architecture to provide functional services (Salehi *et al*., 2011). On the logical level, the realization of the functions through the interaction of mechanic, electronic, and software components modelled.
4. **System Verification** means generally checking whether the way in which something is realized coincides with the specification. When checking the validity of a program, reference is also made to program verification. The verification is generally realized in a formal manner. **System Validation** is to be understood as meaning testing whether the product is suitable for its intended purpose or achieves the desired value. Validation comprises, for example, checking whether the description of an algorithm coincides with the problem to be solved. It generally does not have to be carried out in a formal manner.
   Verification is the "confirmation, through the provision of objective evidence, that specified requirements have been fulfilled." (ISO 9000, 2015). "Verification is a set of activities that compares a system or system element against the required characteristics. This includes, but is not limited to, specified requirements, design description and the system itself" (ISO/IEC TS 24748- 1, 2016).

   Verification proves whether the system is created right (NASA, 2007, Balci, 2003).

   Validation is the "confirmation, through the provision of objective evidence, that the requirements for a specific intended use or application have been fulfilled." (ISO 9000, 2015). Validation is the set of activities that ensure and provide confidence that "system is able to accomplish its intended use, goals and objectives in the intended operational environment." (ISO/IEC TS 24748-1, 2016).

   Validation proves whether the right system was created (NASA, 2007, Balci, 2003).

5. **System Production/Additive Manufacturing/tests level -** In order to achieve automation and data exchange with manufacturing technologies, in this stage, Industry 4.0 will be reflected. It includes cyber-physical systems, the Internet of things, cloud computing and cognitive computing. Production and logistics get right to the heart of the supply chain, from sourcing raw materials to delivering finished products to customers. In this level, system/product manufacturing and the test will be carried out. **Additive manufacturing** uses data computer-aided-design (CAD) software or 3D object scanners to direct hardware to deposit material, layer upon layer, in precise geometric shapes. As its name implies, additive manufacturing adds material to create an object (Salehi *et al*., 2009).
6. **System Usage/Service level -** In this level, we could conceptualize it in three dimensions: usage frequency, usage function, and usage situation. The concept of IOT (Internet of Things) is also considered when it comes to the usage and service of the system to satisfy the customer's needs. The focus is the impact of these usage dimensions on different application scenario with customer satisfaction. System usage could include the following aspects:

   – Usage frequency refers to how often the product is used, regardless of the product functions used, or the different applications for which the product is used.

https://doi.org/10.1017/dsi.2019.377 Published online by Cambridge University Press

– Usage function refers to what extent the product features/ functions are utilized by the consumer, regardless of how often the product is used.
– Usage Situation refers to the different applications for which a product is used and the different situations in which a product is used regardless of either usage frequency or usage function.

MAGIC has the shape of infinity loop to demonstrate that process chain of each stage does not end when the product has been delivered and applied by the customer. On the contrary, it will keep going back to the first requirement stage when the user has new demand for the product. Additionally, we could notice there is also a small loop between each stage which indicates the short-term feedback exchange between adjacent 2 stages.

4 3 DIMENSIONS OF MAGIC

The 3 dimensions of MAGIC as we can see in figure 3, are disciplines, tools, and systems, data and methods which also complies with the 3 dimensions structure of MBSE. These 3 dimensions describe the different aspects of MAGIC approach in product development process. Furthermore, based on different disciplines (mechanics, electric/electronics, software), the MAGIC approach in each field of discipline could interact with each other in different stages through information flows. Moreover, it is noticeable that the portion of each discipline takes a different amount. As we know, from the last 50 years, the importance of software is increasing largely. Additionally, with the development of an electric and electronic component, the involvement of software in this discipline is also getting higher. Design and design methods of this 3 disciplines should be put to the test and their suitability for a modern interdisciplinary design approach should be checked and integrated into an interdisciplinary Method and process. IT solutions are partially available, but require further development in terms of applicability and integration.

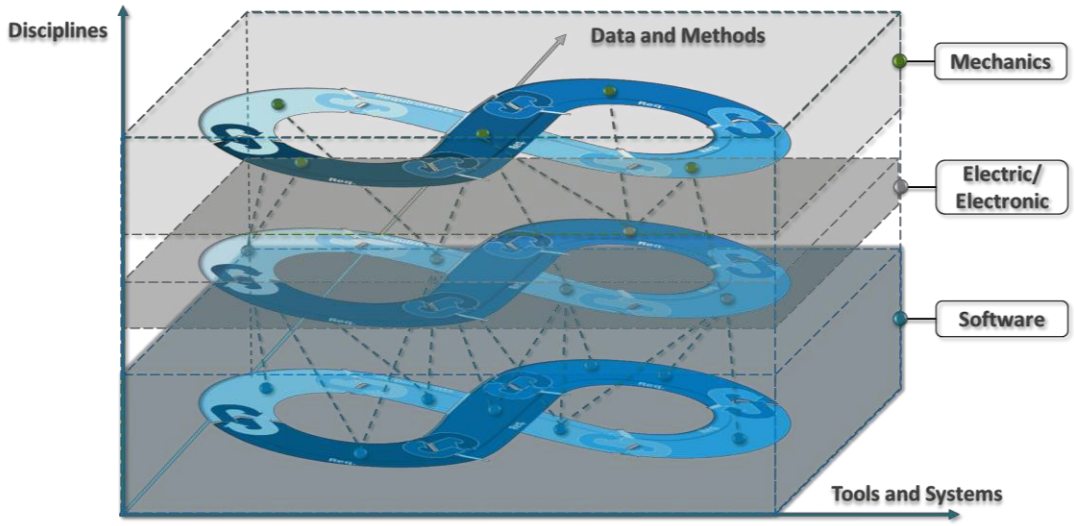

Figure 3. 3 Dimensions of the MAGIC approach according to Salehi

4.1 Disciplines

– **Mechanics:** Mechanics is that area of science concerned with the behaviour of physical bodies when subjected to forces or displacements, and the subsequent effects of the bodies on their environment. Almost all procedural models established for mechanics assume a product development process consists of four main phases: requirements/task clarification, planning design, designing and working out, detailing (Eigner and Roubanov, 2014)

– **Electronic/Electric:** The consideration of design methods in electrical engineering and electronics (E/E) gives a broader picture than in mechanics. Since 1) the very different applications in the field of electrical engineering and electronics; 2) the fundamentally different approach of the designer in electrical engineering and electronics with respect to mechanics, which includes a design level of the schematic design before the layout (layout). This schematic level already has simulated functional-logical elements; 3) the rapid technological change, especially in digital circuit design.

https://doi.org/10.1017/dsi.2019.377 Published online by Cambridge University Press

Complexity and integration density are constantly increasing. The requirements for higher packing density, smaller structure size, performance, smaller space requirement, higher clock rate, and lower power consumption are increasing; 4) the evolution of automation techniques in terms of logical and physical design as well as verification (Eigner and Roubanov, 2014).

– **Software:** it can be defined as goal-oriented provision and systematic use of principles, methods, and tools for the division of labor, engineering, and application of extensive software systems. Due to the high cost of creating and maintaining complex software, the development is based on a structured process or phase and process models. These models divide the development process into manageable, temporally and content-limited phases. The software will be completed step by step. The phases are closely interlinked throughout the development process. In practice, methods that sacrifice the multilevel of system analysis, system design/concept, and subsequent implementation and testing are also used. These methods are called agile software development (Eigner and Roubanov, 2014).

4.2 Tools and systems

Tools and systems are the means to handle, process and execute the data and information which are generated or collected during the system development process. It is significant to select the appropriate tool or system to build up a running environment for a different kind of data and information process. A proper tool or system can simplify the working process and accelerate the working efficiency.

4.3 Data and methods

During the carrying out each stage of MAGIC approach, the data has been generated or collected from the beginning to the end. Undoubtedly, we need to keep the valid data and work further. Besides, those data would be helpful to trace back the requirement or even perform validation and verification process.

5 MAGIC WITH MBSE

One of the core advantages of MBSE is transferring the document-based storage of data into a model-based storage of data. In this way, the data can be accessed easier and the dependencies between data are more transparent (Hooshmand *et al*., 2017). Therefore, the transparency of development steps, as well as the traceability of changes increases and the handling of complexity, improves (Kleiner and Husung, 2016). The application of MBSE is based on three pillars as illustrated in figure 4. System modelling by different roles and different teams establishes the foundation of MBSE. Methods, modelling languages, and tools are three crucial enablers for MBSE.

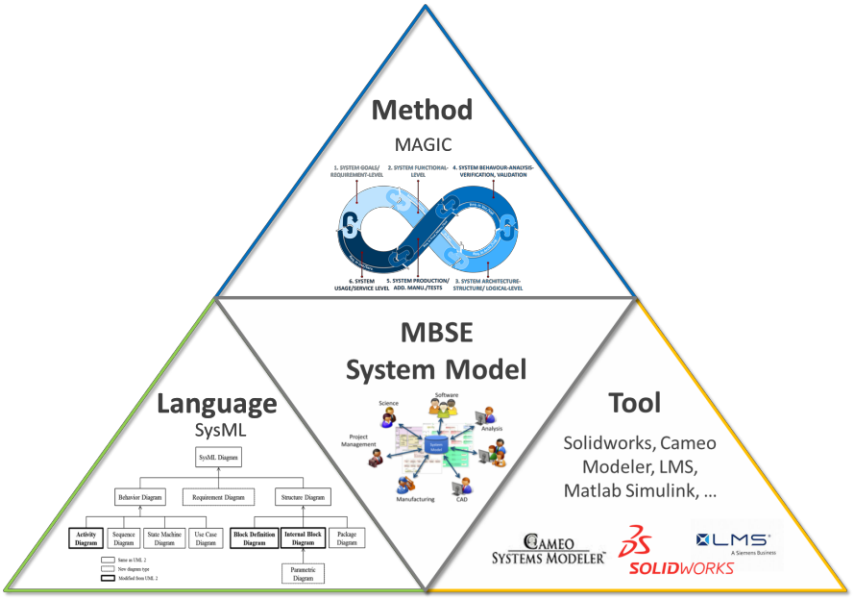

Figure 4. 3 Pillars of MBSE with the MAGIC approach as 'Method' (Salehi et al., 2018).

https://doi.org/10.1017/dsi.2019.377 Published online by Cambridge University Press

The method describes how the language has to be used to generate the actual system model. Previously, Estefan (2008) performed an extensive survey of six MBSE methods, which have been practiced in different industry projects (Estefan, 2008). Since then, new methods, in our case-MAGIC approach, have also been developed by various organizations for their internal exertion of MBSE approach (Friedenthal *et al*., 2009).

The graphical modelling language defines which elements and relationships within a model. Commonly used modelling language includes SysML, UML (Unified Modelling Language), IDEF (Integrated DEFinition Methods), etc. It could help users with specifying, analysing, designing, and verifying complex systems that may include hardware, software, information, personnel, procedures, and facilities (Object Management Group, 2017b). A standardized and formalized language like OMG Systems Modelling Language (OMG SysML) is necessary for the formal definition of the system (Friedenthal *et al*., 2009). The origin of SysML is the Unified Modelling Language (UML) which was adapted by the Object Management Group Inc. to support the modelling of complete technical systems and integrating the needs of different domains. The concept of SysML consists of a central system model which can be displayed via different types of diagrams. Each of these diagrams offers a different view of the same system model and thereby supports different requirements to the view (Kößler and Paetzold, 2017).

The tool is necessary to support the generation of a system model. Basically, it provides a user interface to facilitate the generation of the model. For instance, Cameo Systems Modeller is one of the tools to realize the modelling of the system. This MBSE environment of No Magic Inc. supports engineers with engineering analyses for design decisions, evaluation and requirements verification as well as checking model consistency and tracking design progress with metrics (Inc). Some other tools like Modelica, LMS, Solidworks, etc. are also been widely used.

6 CONCLUSION AND FURTHER WORK

In this paper, we adopt the concept of agile into MBSE and then proposed the new approach - Munich Agile MBSE Concept (MAGIC). MAGIC approach derived from the shape of DevOps infinity loop model and consists of 6 levels: system goal and requirement level, system functional level, systems architecture and logical level, system behaviour analysis and verification and validation, system production and additive manufacturing and test, system usage and service level. The highlights of the MAGIC approach can be concluded as 1) the requirements which have been defined in the first stage will be traced at each following stages; 2) communication between every 2 stages always exists in order to have a close connection during the system development phase; 3) the idea of Industry 4.0 has been included and reflected to achieve automation and data exchange with manufacturing technologies; 4) the concept of IOT (Internet of Things) is also considered when it comes to the usage and service of the system to satisfy the customer's needs; 5) the whole spirit of agile is reflected as the iterative and incremental design and development.

The MAGIC approach benefits cyber-physic product design in many ways. For the further work, in order to verify and demonstrate the concept and feasibility of MAGIC, we plan to apply it to a comprehensive cyber-physic use case of Mars Rover development project.

REFERENCES

Albers, A., Bursac, N., Margot Eckert, C., Walter, B., Wilmsen, M. and Heimicke, J. (2018), *Agile Method Development: A live-lab Case Study on Product Properties for Process Planning*, pp. 713–724. 10.21278/idc.2018.0341.

Anderl, R., Eigner, M., Sendler, U. and Stark, R. (2012), Smart Engineering – Interdisziplinäre Produktentstehung – acatech DISKUSSION.

Soon, B.K., Chung, S., Choh, Y. and Dupuis, M. (2013), *A grounded theory analysis of modern web applications: knowledge, skills, and abilities for DevOps*. https://doi.org/10.1145/2512209.2512229

Bender, K. (2005), *Embedded Systems - qualitätsorientierte Entwicklung*, Springer-Verlag, Berlin, Heidelberg. https://doi.org/10.1007/b138984.

Brown, T. and Wyatt, J. (2010), "Design Thinking for Social Innovation IDEO", *Development Outreach*, Vol. 12 No. 1, pp. 29–43. https://doi.org/10.1596/1020-797X_12_1_29.

Cao, L. and Ramesh, B. (2007), "Agile software development: Ad hoc practices or sound principles?", *IT professional*, Vol. 9 No. 2, pp. 41–47. https://doi.org/10.1109/MITP.2007.27.

Eigner, M., Gilz, T., and Zafirov, R. (2012), *Interdisziplinäre Produktentwicklung - Modellbasiertes Systems Engineering*. [online] PLM portal. Available at:

https://doi.org/10.1017/dsi.2019.377 Published online by Cambridge University Press

https://www.plmportal.org/de/forschungdetail/interdisziplinaere-produktentwicklung-modellbasiertes-systems-engineering.html (accessed 12.07.2018).
Eigner, M. and Roubanov, D. (2014), *Modellbasierte virtuelle Produktentwicklung*, Springer Vieweg. https://doi.org/10.1007/978-3- 662-43816-9.
Estefan, J.A. (2008), *Survey of Model-Based Systems Engineering (MBSE) Methodologies, INCOSE*. Available at: www.omgsysml.org/MBSE_Methodology_Survey_RevB.pdf (Accessed 13.11.2017)
Friedenthal, S., Moore, A. and Steiner, R. (2009), *A practical guide to SysML, The systems modeling language, 3th Edition*, Morgan Kaufmann, Waltham. https://doi.org/10.1016/c2013-0-14457-1.
Graessler, I., Hentze, J. and Bruckmann, T. (2018), *V-Models for Interdisciplinary Systems Engineering, International Design Conference - Design*. https://doi.org/10.21278/idc.2018.0333
Hooshmand, Y., Adamenko, D., Kunnen, S. and Köhler, P. (2017), "An approach for holistic model-based engineering of industrial plants", *Proceedings of the 21st International Conference on Engineering Design (ICED17)*, Vancouver, Canada, 21.-25.08.2017, Vol. 3: Product, Services and Systems Design.
INCOSE (2015), *INCOSE Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*, 4th Edition, John Wiley & Sons, Inc., Hoboken, New Jersey.
ISO 9000 (2015), *Quality management systems - fundamentals and vocabulary*, 4th edition, ISO Copyright office, Geneva.
ISO/IEC TS 24748-1 (2016), *Systems and software engineering – Life cycle management – Part 1: Guidelinesb for life cycle management ISO Copyright office*, Geneva.
Kaufmann, U. and Schuler, R. (2016), "Systems Re-Engineering - ein Beitrag zur Integration von MBSE und PLM", *Tag des Systems Engineering, Herzogenaurach*, München, 25.-27. October 2016, Hanser Verlag, pp. 343–352. https://doi.org/10.3139/9783446451414
Kleiner, S. and Husung, S. (2016), "Model Based Systems Engineering: Prinzipen, Anwendung, Beispiele, Erfahrung und Nutzen aus Praxissicht", *Tag des Systems Engineering, Herzogenaurach*, München, 25.-27. October 2016, Hanser Verlag, pp. 13–22. https://doi.org/10.3139/9783446451414
Kößler, J. and Paetzold, K. (2017), "Integration of MBSE into existing development processes - expectations and challenges", *Proceedings of the 21st International Conference on Engineering Design (ICED17)*, Vancouver, Canada, 21.-25.08.2017, Vol. 3: Product, Services and Systems Design.
Lindlöf, L. and Furuhjelm, J. *Agile beyond software - a study of a large scale initiative international design conference, DESIGN 2018*. https://doi.org/10.21278/idc.2018.0411
Salehi, V. and Mcmahon, C. (2009), "Action Research into the use of parametric associative CAD systems in an industrial context", *International Conference on Engineering Design, ICED'09*, Stanford, CA, USA, 24 - 27 August 2009, Stanford University.
Salehi, V. and Mcmahon, C. (2009), "Development of a Generic Integrated Approach for Parametric Associative CAD Systems", *International Conference on Engineering Design, ICED'09*, Stanford, CA, USA, 24 - 27 August 2009, Stanford University.
Salehi, V. and Mcmahon, C. (March 2011), "Development and Application of an Integrated Approach for Parametric Associative CAD Design in an Industrial Context", *Journal paper, Computer Aided Design and application*, Vol. 8 No. 2, pp. 225–236.
Salehi, V. and Mcmahon, C. (2011), "Development of an evaluation framework for implementation of parametric associative methods in an industrial context", *International Conference on Engineering Design, ICED'11*, Copenhagen, Denmark, August 2011, DTU University.
Salehi, V. and Wang, S. (2017), "Using point cloud technology for process simulation in the context of digital factory based on a systems engineering integrated approach", *Proceedings of the 21st International Conference on Engineering Design (ICED17)*, Vancouver, Canada, 21.- 25.08.2017, Vol. 3: Product, Services and Systems Design.
Salehi, V., "Model Based Systems Engineering (MBSE) as a holistic Approach and Enabler for Autonomous Driving Systems," Hochschule für Angewandte Wissenschaften München, 2018.
Salehi, V., Florian, G. and Taha, J. (2018), "Implementation of Systems Modelling Language (SysML) in Consideration of the CONSENS Approach", *Proceedings of the DESIGN 2018 15th International Design Conference*, pp. 2987–2998. http://doi.org/10.21278/idc.2018.0146
VDI (2004), *VDI 2206 - Design methodology for mechatronic systems*, The Association of German Engineers (VDI), Düsseldorf.
Walden, D.D., Roedler, G.J., Forsberg, K., Hamelin, R.D. and Shortell, T.M. (2015), *Systems engineering handbook: A guide for system life cycle processes and activities*, 4th ed., Wiley.
Weilkins, T. (2008), *Systems Engineering mit SysML/UML: Modellierung, Analyse, Design, dpunkt Verlag*, 2nd. Edition, Heidelberg, Germany.
Zhang, T. and Dong, H. (2009), *Human-centred design: An emergent conceptual model*, Royal College of Art.

https://doi.org/10.1017/dsi.2019.377 Published online by Cambridge University Press