Multi-Source Design: A Method for Physical-Digital Product Development Through Agility, Modularity, and Decoupling

Søren Christian Madsen
Mjølner Informatics A/S
Aarhus, Denmark
scm@mjolner.dk

Bent Bisballe Nyeng
Mjølner Informatics A/S
Aarhus, Denmark
bbn@mjolner.dk

Jens Bæk Jørgensen
Mjølner Informatics A/S
Aarhus, Denmark
jbj@mjolner.dk

***Abstract*—At Mjølner, we apply a method for Physical-Digital Product Development, which we call *Multi-Source Design* (MSD). This builds on several existing well-known best-practices, which we have combined and applied on a wider scope encompassing adjacent disciplines to software development such as hardware and mechanical development. We see the MSD method as a contemporary approach to physical-digital product development by integrating agility, modularity, and decoupling. MSD adapts principles from software development to create resilient and adaptable physical-digital products, addressing vulnerabilities such as supply-chain disruptions, high maintenance costs, and inflexible product architectures. This paper describes the motivation behind MSD, introduces the method itself, describes experiences from industries such as automotive or electronics and considers the benefits of the method.**

***Index Terms*—agility, modularity, decoupling, design method, software and hardware, multi-source design.**

I. INTRODUCTION

The COVID-19 pandemic highlighted systemic vulnerabilities across global supply chains, particularly in industries dependent on electronic components and physical-digital products. Widespread component shortages, extended lead times, and reliance on single suppliers severely disrupted production [1, 2] pointing to a need for manufacturers to source alternate components from alternate vendors, also known as *multi-sourcing*. These events underscored the need for techniques and strategies to dynamically adapt software and hardware components based on system or environmental changes.

In this paper, we present the *Multi-Source Design* (MSD) method, which addresses these challenges by applying agility, modularity, and decoupling principles to physical-digital product development. MSD bridges traditional gaps between design, procurement, and production, fostering a software ecosystem-like [3] approach, where physical-digital products can be continuously adapted to shifting supply conditions and market demands. This method builds on concepts long-established in software development [4, 5], applying them for hardware and procurement workflows using modular, scalable, and flexible embedded architectures.

This paper is structured as follows: In Section II, we discuss the motivation for MSD, such as avoiding vendor lock-in, product miniaturization, and uncertain requirements. Section III introduces the MSD method, explaining its integration of agile methods, modularity, and decoupling through reduction of interdependencies in hardware and software development. Section IV presents experiences of using MSD and related techniques. Section V highlights the benefits of applying the MSD method. Finally, Section VI includes a discussion and conclusion and points to potential areas for future research and exploration, both in academia and industry.

II. MOTIVATION FOR THE MULTI-SOURCE DESIGN METHOD

Products are increasingly at risk of relying on specific suppliers or components that may not remain available throughout the product lifecycle. MSD ensures that the system architecture is flexible, enabling components to be easily swapped after launch with minimal impact. This flexibility is crucial not only when new components must replace existing ones due to end-of-life issues or supply shortages but also as a strategy to future-proof designs, avoiding the recurrence of similar problems in subsequent versions [6].

Traditionally, the focus during new product development tends to be on the goal rather than the journey, often locking in key decisions early, particularly in the context of hardware and mechanics. Figure 1 illustrates this by showing an agile software approach running in parallel with a traditional flow for the hardware and mechanical development.

The reason behind the traditional flow has typically been the time and cost penalties associated with manufacturing of hardware, making it seem desirable to keep iterations for these disciplines to a minimum.

Another typical product development challenge is downsizing. The end product may need to fit into a very small form factor, which may slow progress and make debugging and testing more difficult. MSD counters this by allowing development to proceed with a larger form factor during early stages. This enables developers to focus on functionality and prototyping before the constraints of a smaller form factor are introduced. Such a strategy ensures that decisions regarding

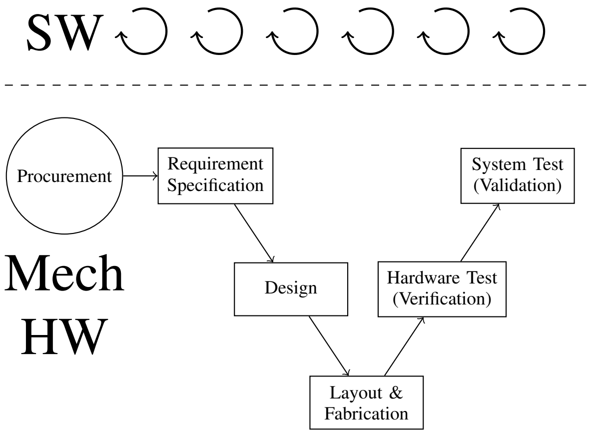

Fig. 1. Traditional mechanical/hardware and software co-design workflow

size are made later in the process, based on empirical observations of firmware needs rather than speculative decisions based solely on cost considerations [7].

The inherent uncertainty of product development further complicates the process. As is common in agile development, the exact requirements for hardware components or product features are often not clear at the outset. MSD facilitates the creation of flexible platforms that can evolve over time, supporting the iterative addition of components and features. This approach provides room for experimentation with options such as wireless connectivity or battery backup and allows teams to adjust design choices as the product matures [8].

Testability, both for daily development and automated quality assessment, is another challenge. In traditional hardware development, the lack of easy testing mechanisms can slow progress. MSD addresses this issue by enabling the creation of hardware variants with probes and test points, facilitating integration into automated Hardware-in-the-Loop (HIL) test setups. This flexibility allows test points to be added or removed as needed, ensuring that the product is adaptable to ongoing testing requirements without affecting the final product [9].

The coordination of multiple teams with varying requirements also presents a challenge. In complex development projects, different teams may have different needs regarding the hardware platform, which can lead to conflicts if not properly managed. MSD allows for the creation of numerous concurrent variants of the product, each tailored to the specific needs of different teams. This ensures that one team's requirements do not constrain or limit the options for another, fostering greater collaboration and efficiency [10].

Finally, the pace of development can be significantly hampered by cumbersome hardware processes. MSD improves development speed by enabling large portions of the system to be simulated, reducing both time and costs. Additionally, it facilitates easier debugging and testing, which accelerates the overall development cycle and leads to faster product iterations [11].

III. THE MULTI-SOURCE DESIGN METHOD

MSD aims to address the complexities inherent in the co-development of mechanical design, hardware such as electronics and software systems. Advancements in prototyping technologies, particularly in the realm of printed circuit board (PCB) manufacturing, plays a critical role in enabling synchronized development cycles. These innovations allow hardware development to adopt iterative practices, aligning closely with agile software methodologies and enabling rapid responses to change.

The core principles of MSD implementation are:

- Agile methods across all disciplines
- Modularity through facet-oriented design
- Decoupling multidisciplinary dependencies

Each of these principles and their impact are described in Sections III-A to III-C. Moreover, in Section III-D, we present guidelines for how the principles can be operationalized.

A. Agile methods across all disciplines

Agile methodologies have traditionally been associated with software devlopment. Moreover they have previously been shown to also work in hardware development [12, 13]. MSD now extends agility to the software, hardware, and procurement processes as a whole. Recent advancements in prototyping technologies, encompassing not only PCB manufacturing but also methods such as 3D printing, have been pivotal in enabling the iterative workflows central to MSD. These innovations have significantly accelerated the development cycles for hardware, software, and mechanical components in conjunction, facilitating a more integrated and agile approach to product design.

Where prototyping lead times previously spanned weeks or even months, modern techniques now potentially reduce these cycles to mere days. For example, rapid PCB manufacturing allows for quick iterations in electronic design, while 3D printing enables the swift creation of mechanical prototypes. This reduction in lead time empowers multidisciplinary teams to experiment, test, and refine their designs at a much faster pace.

The broader range of prototyping technologies can promote deeper integration between disciplines. Hardware teams can align their development schedules with software sprints, while mechanical components can be iterated alongside both, ensuring a synchronized and cohesive development process. These advancements collectively enable organizations to respond dynamically to changing requirements, reduce integration challenges, and enhance system-level coherence, embodying the agility at the core of MSD.

The MSD workflow spans the disciplines of mechanics, hardware, procurement, and software development. Each discipline progresses independently yet remains interconnected through feedback loops. Since digital hardware typically requires software to operate, verification becomes a multidisciplinary effort, requiring software developers not only to develop functionality but also to support hardware verification. This is illustrated in Figure 2.

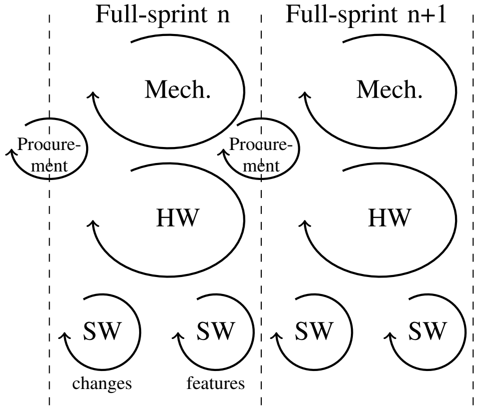

Fig. 2. Iterative workflow for MSD integrating mechanical, hardware, and software development, including touch-points with procurement

The Figure shows how the overarching principle of agile sprints are orchestrated across disciplines with the following key deviations from traditional approaches:

- **Hardware & Mechanics iterations:** Hardware and mechanics are developed in iterative sprints, akin to software cycles, allowing continuous refinement until requirements are met.
- **Procurement integration:** Procurement teams actively participate in hardware sprints, enabling real-time decision-making for alternative components, and supply-chain considerations [14].
- **Multi-disciplinary synergy:** E.g., one or more software iterations align with each hardware sprint, ensuring the functionality required for hardware validation is available when needed.

B. Modularity through facet-oriented design

Facet-oriented design is a term we use for the organization of development around distinct cohesive functional subsets that span hardware, software, and mechanics. Each facet denotes a cross-disciplinary functionality (e.g., "mobility") rather than just a single module. For instance, the mobility facet may include mechanical elements (e.g., gearbox), electrical components (e.g., motor & power-supply), and software for control. This principle mimics the well-known *separation of concerns principle* and fosters resilience to changes while reducing dependencies between system components, enabling faster iterations and streamlined integration. Flexibility is a key aspect of MSD and a high level of modularity is a key requirement to achieve this. By decomposing e.g. software components into smaller, independently testable units, a modular architecture facilitates parallel development and iterative refinement. This principle ensures that changes in one component can be isolated and evaluated without impacting the broader system, isolating problems and enhancing adaptability. Within the context of MSD, this principle extends beyond software to encompass hardware and procurement, ensuring that all aspects of the development process align with the overarching system goals.

C. Decoupling multidisciplinary dependencies

To mitigate the problems of supply-chain disruptions and over-dependency on specific components, a key goal of MSD is to reduce interlocking dependencies across disciplines. Preventing such dependencies to the highest degree possible is crucial in order to enable multi-sourcing. The following actions are proposed to mitigate this:

- **Dependency reviews:** Designs are regularly reviewed to identify and minimize interlocking dependencies across modules.
- **Alternative designs:** Each development phase considers alternative components and configurations, deliberately changing key module elements to enhance adaptability.

This approach ensures robust designs capable of accommodating diverse suppliers, fostering resilience in procurement and manufacturing [15]. Adopting a multidisciplinary iterative workflow helps avoid common pitfalls of sub-optimization within one discipline, which may introduce unnecessary constraints for the adjoining disciplines.

D. From principles to guidelines

In order to apply MSD to a project, the mentioned core principles should be considered and followed. In practice, this could be operationalized in the the following preliminary guidelines:

- Synchronize scrum sprints across all disciplines.
- Schedule periodic touchpoints involving all disciplines.
- Focus on improving the ability for adjacent disciplines to work efficiently.
- Share responsibility for iteratively refining product requirements.
- Design for modularity and change-readiness across all disciplines.
- Track and reduce the number of singularly essential components.

Further details can be found at `http://msd-guide.org`

IV. EXPERIENCES WITH MULTI-SOURCE DESIGN AND RELATED TECHNIQUES

The principles underlying MSD have been applied by Mjølner for a few years. This section examines experiences from diverse sectors, such as measurement instruments, automotive, and manufacturing, to illustrate how the application of MSD principles can lead to improved resilience, adaptability, and efficiency. These examples also highlight the cross-domain relevance of MSD, extending its utility beyond any single industry. The example presented in section IV-A is based on our own experience whereas the examples presented in sections IV-B and IV-C are from the literature where we observe that MSD-like principles are applied.

A. Development of a measurement instrument in an industrial company

A case encountered by the authors at a major British electronics measurement device company highlights the foundational motivation behind the development of the MSD concept. A large portion of their products used an old Bluetooth module, which reached end-of-life. Using the MSD method, a replacement was developed: a pin-compatible daughterboard with a small microcontroller that translated the commands of the old module into those of the new module. This allowed the company to continue production without modifying the existing products. Furthermore, the design ensured that if the new module was to be end-of-lifed in the future, a replacement could be easily integrated without starting from scratch. This case exemplifies how MSD can foster resilience and adaptability in product development, motivating the authors to formalize these principles into the MSD method.

B. Component multi-sourcing in Tesla

A notable example, as described in [16], shows application of similar principles as those described by us in this paper in Tesla's approach to vehicle development. Tesla employs a highly modular architecture in its electric vehicles, enabling rapid adaptation to component availability and technological advancements. During the global semiconductor shortage, Tesla re-engineered its software to support alternative chipsets, leveraging modular software and hardware interfaces to maintain production continuity. This approach aligns closely with MSD's emphasis on multi-sourcing and modularity, demonstrating how these principles can mitigate supply-chain disruptions. Furthermore, Tesla's agile development processes, characterized by frequent over-the-air software updates, exemplify the iterative design cycles integral to MSD, allowing for continuous adaptation and improvement of their products.

C. Standardized platforms at LEGO

In the manufacturing sector, the concept of platform-based product development reflects the benefits of MSD principles. Companies such as LEGO employ a standardized platform approach, allowing for the integration of interchangeable components across product lines. This strategy not only reduces production costs but also enhances resilience against supply-chain disruptions. By designing components to meet common specifications, LEGO ensures compatibility across a wide range of products, embodying the multi-sourcing and modularity principles central to MSD [17]. Additionally, this approach fosters collaboration between design and production teams, enabling the rapid introduction of new products without compromising quality or efficiency [18].

V. BENEFITS OF USING MULTI-SOURCE DESIGN

The benefits observed across these industries — resilience to supply-chain disruptions, adaptability to changing requirements and efficiency in iterative development — demonstrate the practical impact of MSD principles. While these examples span diverse sectors, the underlying strategies of modularity, multi-sourcing, cross-functional collaboration, and iterative processes provide a common foundation.

A. Resilience through modularity and multi-sourcing

MSD enables businesses to build resilience by designing systems that accommodate alternative components and suppliers. The automotive industry's transition to modular platforms, as seen with MQB and TNGA, the modular platforms used by Volkswagen and Toyota respectively, highlights how standardization can reduce dependencies on specific suppliers [19, 20]. Similarly, the electronics industry, which frequently encounters rapid component obsolescence and technological advancements, benefits from modular architectures that facilitate the replacement and upgrading of key components, such as chips or microprocessors [21]–[23]. The ability to easily replace or upgrade critical components within a standardized framework not only extends the lifecycle of devices but also reduces problems associated with supply chain disruptions, allowing for a more flexible response to market and technological changes [24].

B. Accelerating time-to-market

Modular and agile workflows have the potential to significantly reduce development timelines. By reusing standardized components, manufacturers can bypass extensive redesign processes. For instance, Tesla's modular battery packs facilitate rapid iterations, allowing for the incorporation of emerging technologies without delaying production [25].

C. Reduction of cost and improved sustainability

The reuse of modules across product lines not only lowers development costs but also reduces waste and energy consumption, contributing to more sustainable practices. The aerospace industry, for example, increasingly adopts modular avionics systems to streamline upgrades and extend the lifespan of aircraft [26].

VI. DISCUSSION & CONCLUSION

Implementing MSD requires a shift in organizational culture towards collaboration and adaptability. Traditional silos between design, procurement, and production must be dismantled to enable integrated workflows. Companies such as Toyota demonstrate how cross-functional teams can drive innovation while maintaining efficiency [20]. Adopting MSD can be a transformative yet challenging process. Organizations must overcome technical, cultural, and managerial barriers to fully integrate its principles into their workflows. A significant cultural challenge arises from the need to break down siloed structures, requiring cross-functional collaboration and a shift in accountability. Resistance to these changes often stems from organizational inertia and a lack of clear integration strategies [27].

Technically, the emphasis on modularity and multi-sourcing demands precise interface definitions and robust risk management. Poorly planned integration or inconsistent supplier quality can erode the benefits of MSD [28]. Iterative processes, while central to MSD, may conflict with traditional project management approaches, particularly in highly regulated industries [29].

Common pitfalls include an overemphasis on modularity without proper integration planning and prematurely adopting MSD without fully understanding system interdependencies [21]. Over generalisation is often seen in the industry due to an over eagerness to introduce modularity without considering the actual needs of the context. Addressing these challenges requires proactive leadership, technical training, and careful planning, alongside iterative learning to adapt MSD principles effectively to specific contexts.

This paper has described the MSD method as a method for addressing critical challenges in physical-digital product development. By integrating modularity, agility, and procurement collaboration, MSD offers a pathway to more resilient and adaptable manufacturing processes. Through examples from industries such as automotive and electronics, we have highlighted how modular systems and iterative workflows enhance supply-chain resilience, reduce costs, and accelerate innovation. While implementing MSD requires cultural and technical adjustments, its long-term benefits make it a compelling strategy for industries seeking to thrive in an increasingly dynamic market.

Future work should explore the economic impacts of introducing MSD into an organization, both the costs and savings endured. Also, the potential impacts for consumers or end-customers in terms of lifetime and serviceability of products would be interesting to explore. Finally, we see the MSD method as a work in progress with potential for expansion, refinement, and standardization.

ACKNOWLEDGEMENTS

We acknowledge the use of OpenAI's ChatGPT in the preparation of this paper. ChatGPT was utilized for refining text and assisting with the drafting process of converting an existing handwritten sales-oriented white-paper [30] and presentation material describing the MSD principles. All output from ChatGPT was carefully reviewed and modified by the authors to ensure the final content meets the academic standards of this paper.

REFERENCES

[1] S. Chopra and M. S. Sodhi, "Managing risk to avoid supply-chain breakdown," *MIT Sloan Management Review*, 2004.

[2] D. Ivanov, A. Dolgui, and B. Sokolov, "Pandemic-induced supply chain disruptions: Where are we now and what is next?" *Transportation Research Part E*, vol. 136, p. 101922, 2020.

[3] J. Bosch, "From software product lines to software ecosystems," in *Proceedings of the 13th International Software Product Line Conference*, ser. SPLC '09. USA: Carnegie Mellon University, 2009, p. 111–119.

[4] B. et al., "Manifesto for agile software development," https://agilemanifesto.org, 2001.

[5] J. Highsmith and A. Cockburn, "Agile software development: The business of innovation," *Computer*, vol. 34, no. 9, pp. 120–127, 2001.

[6] D. Sculley *et al.*, "Machine learning: The high interest credit card of technical debt," *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1423–1432, 2019.

[7] A. Hassan *et al.*, "Hardware design and prototyping challenges in the iot era," *IEEE Access*, vol. 8, pp. 134 564–134 580, 2020.

[8] J. Highsmith, *Agile Software Development: Principles, Patterns, and Practices*. Pearson Education, 2002.

[9] S. Mellor *et al.*, *Model-Based Systems Engineering: Fundamentals and Methods*. Pearson Education, 2014.

[10] R. Buehler *et al.*, *Hardware Design and Prototyping*. Prentice Hall, 2003.

[11] J. Sutherland, *Scrum: The Art of Doing Twice the Work in Half the Time*. Crown Business, 2010.

[12] N. Ovesen, "The challenges of becoming agile: Implementing and conducting scrum in integrated product development [phd thesis]," Sep. 2012.

[13] A. Atzberger and K. Paetzold, "Current challenges of agile hardware development: What are still the pain points nowadays?" *Proceedings of the Design Society: International Conference on Engineering Design*, vol. 1, no. 1, p. 2209–2218, 2019.

[14] D. Simpson, "5 ways hardware development is just… different," *Altium 365*, 2023, accessed: 2024-12-18. [Online]. Available: https://resources.altium365.com/p/why-hardware-development-is-different

[15] A. Szychulec, "Hardware product development process in 2024," *InTechHouse*, 2024, accessed: 2024-12-18. [Online]. Available: https://intechhouse.com/blog/hardware-product-development-process-in-2024/

[16] M. Johnson and A. Liu, "Tesla's response to the semiconductor shortage: A case of modularity in design," *Automotive Supply Chain Review*, vol. 12, no. 4, pp. 22–30, 2022.

[17] T. Hansen and P. Rasmussen, "Lego's platform strategy: A case study in modularity and innovation," *Journal of Product Innovation Management*, vol. 35, no. 1, pp. 56–70, 2018.

[18] R. Githens and S. Clark, *Designing Platforms: Managing Product Innovation Through Modularity*. Cambridge University Press, 2019.

[19] Automotive News Europe, "Volkswagen's mqb platform: A model for scalability," https://www.autonews.com, 2021.

[20] Toyota Global, "The toyota new global architecture (tnga)," https://global.toyota, 2020.

[21] C. Y. Baldwin and K. B. Clark, *Design Rules: The Power of Modularity*. MIT Press, 2000.

[22] S. Ghosh and D. J. Teece, "Resilience and flexibility in global supply chains," *California Management Review*, vol. 62, no. 2, pp. 62–78, 2020.

[23] J. Park, M. Lee, and J. Kim, "Advancements in electronics modularity for enhanced device lifecycle management," *IEEE Transactions on Electronics Packaging Manufacturing*, vol. 45, no. 2, pp. 112–126, 2022.

[24] Z. Li, W. Chen, and X. Wang, "Electronics modularity: A path to resilience in the semiconductor industry," *Journal of Electronics and Technology*, vol. 45, no. 3, pp. 215–230, 2023.

[25] D. Welch and E. Behrmann, "Tesla's agile model proves a disruptive force in the auto industry," Bloomberg, 2021.

[26] Aerospace Industries Association, "Modular avionics: Transforming aircraft upgrades," https://www.aia-aerospace.org, 2020.

[27] J. B. Jørgensen, H. L. Christensen, S. T. Hansen, and B. B. Nyeng, "Effective communication about software in a traditional industrial company : an experience report on development of a new measurement instrument," in *2022 IEEE/ACM International Workshop on Software-Intensive Business (IWSiB) @ ICSE'22*, 2022, pp. 39–42.

[28] A. J. Van Weele, *Purchasing and Supply Chain Management: Analysis, Strategy, Planning and Practice*. Thomson Learning, 2005.

[29] R. G. Cooper, "Stage-gate systems: A new tool for managing new products," *Business Horizons*, vol. 33, no. 3, pp. 44–54, 1990.

[30] Mjølner Informatics A/S, "Multi source design - a key to conquer market shares," https://mjolner.dk/hent-multi-source-design-e-bog, 2023.