# WORKING AGILE TO SPEED UP RESEARCH WITH INDUSTRY: FIVE INDEPENDENCE PRINCIPLES



Citation for the original published paper (version of record):

Panarotto, M., Isaksson, O., Söderberg, R. (2023). WORKING AGILE TO SPEED UP RESEARCH WITH INDUSTRY: FIVE INDEPENDENCE PRINCIPLES. Proceedings of the Design Society, 3: 3919-3928. http://dx.doi.org/10.1017/pds.2023.393

# WORKING AGILE TO SPEED UP RESEARCH WITH INDUSTRY: FIVE INDEPENDENCE PRINCIPLES

**Panarotto, Massimo;**
**Isaksson, Ola;**
**Söderberg, Rikard**

Chalmers University of Technology

## ABSTRACT

One of the obstacles to the ability of research to make an impact on industry resides on the research process itself. Today, there is a need to accelerate the means for research to support industrial transformation. At the same time, there is the need to maintain scientific rigorousness, which often requires time. To solve this trade-off, this paper evaluates existing research approaches through the lenses of agile development. The analysis is based on a simulation of research process architectures, and on observations made over several research projects with industry. The results of this analysis highlight five light-but-sufficient rules of research project behavior to keep momentum, motivation and trust when doing research with industry. The paper demonstrates the use of these five rules in a "research sprint" conducted iwith two automotive OEMs.

**Keywords**: Design methods, Design methodology, Research methodologies and methods, validation, agile development

**Contact**:
Panarotto, Massimo
Chalmers University of Technology
Sweden
massimo.panarotto@chalmers.se

**Cite this article:** Panarotto, M., Isaksson, O., Söderberg, R. (2023) 'Working Agile to Speed Up Research with Industry: Five Independence Principles', in *Proceedings of the International Conference on Engineering Design (ICED23),* Bordeaux, France, 24-28 July 2023. DOI:10.1017/pds.2023.393

# 1 INTRODUCTION

There is still a debate about how to 'use' Design Research (DR) to make an effective impact on industry (Gericke et al., 2016; Cross, 2018). One of the reasons reside on the ways of working in DR, which tend to "slow down" the research process reducing the opportunity for impact (Gericke et al., 2020).

When engaging with industry, design researchers adopt a portfolio of different qualitative and quantitative research methodologies with the aim of combining scientific rigour and relevance via practical application (Mårtensson et al., 2016). Typical examples are Action Research (AR; Lewin, 1946) and Design Research Methodology (DRM; Blessing and Chakrabarti, 2009). These methodologies were originated or applied in other scientific fields such as social sciences (Mumford, 2001), human relations (Lewin, 1947) and innovation management (Guertler, et al., 2020). However, DR presents some differences compared to these scientific fields, which risk to reduce the efficiency and effectiveness of DR.

One of these differences is that DR is often conducted via "use cases" provided by an industrial partner (Isaksson, 2016), which intend to mimic realistic situations and problems encountered in design practice (Wallin et al., 2014). Although described as comprehensively as possible, these use cases still represent an idealization of real design problems (e.g., they are conducted in shorter time frames and by a smaller team) (Cash et al., 2022). This increases the risk that a researcher 1) spends a long time in capturing all the aspects that differentiate the use case from the "real" design problem 2) makes coarse assumptions based on needs observed on the "idealized" case studies. In addition, any contextual complexity of the problem risk to be supressed, including e.g., resource prioritisation, impact of real-world disruptions and so forth in the social dimension and technological complexity, system dependencies and similar on the technology side.

Another differentiating element is that in DR, the focus on "prescription" is more pronounced than other scientific fields such as social sciences (Cantamessa, 2003). One common way for DR to transfer its results is in the form methods, tools, guidelines or processes (Gericke et al., 2020). For these reasons, PhD students spend a considerable part of their PhD on developing methods (Cantamessa, 2003; Jagtap et al., 2014) to meet scientific novelty expectations. While this is necessary, it often requires the researcher to learn specific techniques to develop the desired support, in many cases far from their original educational background (e.g., programming). This slows down the "action" process, delaying (or even leaving out) a proper validation of the design method within the (often bounded) time available (Barth et al.,2011; Isaksson et al., 2020). These obstructive factors stress the need to accelerate design research for more effective impact on the industrial practice, while ensuring rigour and relevance.

This paper starts from the observation that such problems are also encountered by industry, which often find itself in the situation of developing and validating a system fast *while still* maturing its solutions (Martin, 2020). In industry, the need to go fast has resulted in the definition of "agile development" (Agile Manifesto 2001), a group of methods in which requirements and solutions evolve through short design loops. Agile development is intended to promote adaptive planning, evolutionary development, early delivery, and encourages rapid and flexible response to change (Cockburn, 2006). Agile methods are commonly deployed in software development but are also increasing in popularity in other more hardware-intensive industrial contexts (Douglass, 2015). Still, the application of agile within DR is limited (Da Silva et al., 2011). It is therefore appealing to find out how design researchers can apply agile development when conducting research in extensive projects with industry. The same trend is observed in other design contexts such as healthcare (Keijzer-Broers and Reuver, 2016).

This paper is based on an agile focused assessment of existing research approaches, as well as on observation made in several research projects between academia and industry. Based on these results, central concepts of planning and evaluation in DR are revisited discussed, which result in a proposal of practical guidelines to conduct intervention-based DR (called "Agile Design Research").

# 2 OVERVIEW OF AGILE DEVELOPMENT

Agile development was envisioned to create adaptive processes to include dynamic customer input into iterative and incremental software development. The core of agile - emphasised also in similar approaches such as design thinking (Brown, 2008) – is the concept of 'change embracing' (Cockburn, 2006). While 'heavyweight' (Dingsøyr et al., 2012) development approaches such as Systems Engineering (Kapurch, 2010) aim at minimizing the risk of change through well-defined requirements



https://doi.org/10.1017/pds.2023.393 Published online by Cambridge University Press

and planning, agile development accepts changes as an inevitable part of the process and strives to develop one functionality of the system at a time. Validation is often conducted by real demonstration. When validated, this new 'piece' of functionality is integrated with the already developed part of the system (Dingsøyr et al., 2012). The agile development community has defined twelve principles, intended to convey the general points of difference between agile and classic "waterfall" models (Table 1).

*Table 1. The twelve principles of agile development*

| | | |
|---|---|---|
| 1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software. | 2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage. | 3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale. |
| 4. Business people and developers must work together daily throughout the project. | 5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done. | 6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation. |
| 7. Working software is the primary measure of progress. | 8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely. | 9. Continuous attention to technical excellence and good design enhances agility. |
| 10. Simplicity–the art of maximizing the amount of work not done–is essential. | 11. The best architectures, requirements, and designs emerge from self-organizing teams. | 12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly. |

In practice, the implementation of agile processes has focused on a few techniques, such as sprint and scrum (Dingsøyr et al., 2012). A sprint is a repeated design-code-test-release cycle lasting from one week to one month, where the team creates and updates a predefined set of lists of features. Scrum is intended to facilitate the daily coordination of a cross-functional team, by assigning tasks to individuals from a prioritized list of features (named the "backlog"). Teams therefore work in reducing this backlog, through sprint cycles, where a new release of working software is done after each sprint.
While agile was conceived to reduce the extensive focus on planning that characterizes typical waterfall models, it has been recognized that for large systems some degree of coordination and planning is necessary (Douglass, 2015; Hernandez et al., 2019). Therefore, coordination mechanisms such as the Scaled Agile Framework (SAFe) were conceived to facilitate the coordination from a single team to multiple teams (Bajpai et al., 2019).

## 3 REQUIREMENTS FOR 'AGILE' IN DESIGN RESEARCH

In previous studies (e.g., Guertler et al., 2020), the requirements for a research methodology have been established after literature analysis and sound discussion. This paper utilizes process modelling to elicit requirements for 'agile' in DR (to follow an "agile" approach itself). Process simulation of agile vs. traditional "waterfall" processes has been already conducted in software development (Mitsuyuki et al., 2017). Also, process modelling has been used in product development to compare different process architectures on some objective functions (e.g., time and cost) (Browning and Eppinger, 2002). Process modelling is used also to evaluate the "robustness" of a process architecture to uncertain conditions (such as changes in requirements) which may propagate through the system causing delays (Maier et al., 2014). It is therefore interesting to evaluate the impact of an agile DR process compared to existing approaches (such as Design Research Methodology or Action Research).
The main variables of interest in this process modelling and evaluation are different process research architectures (e.g., DRM) described as workflows (detailed in section 3.1). The objective functions of the problem are the total duration time of the project (i.e., the lead time), from research clarification to the



https://doi.org/10.1017/pds.2023.393 Published online by Cambridge University Press

final review (e.g., a PhD defence or a final project review). Also, the number of undetected "errors" during the research is an objective function. In the process modelling performed in this paper, these errors are in form of uncertain conditions occurred during the research problem (and detailed in section 3.2). Under these conditions, the analysis evaluates different research process architectures in terms of duration and the ability to handle uncertain research conditions (as detailed in the next section).
The comparison between the process architectures in terms of lead time and "errors" allows to derive requirements for an agile DR process.

### 3.1 Research process architectures studied

Figure 1 a-b shows two of the research architectures considered in this study. These architectures have been modelled as variations of a general DRM process (Blessing and Chakrabarti, 2009). DRM is chosen because of its popularity within the DR community, and because it is often displayed as a system development process, which makes it easier to visualize the differences with an agile way of working.

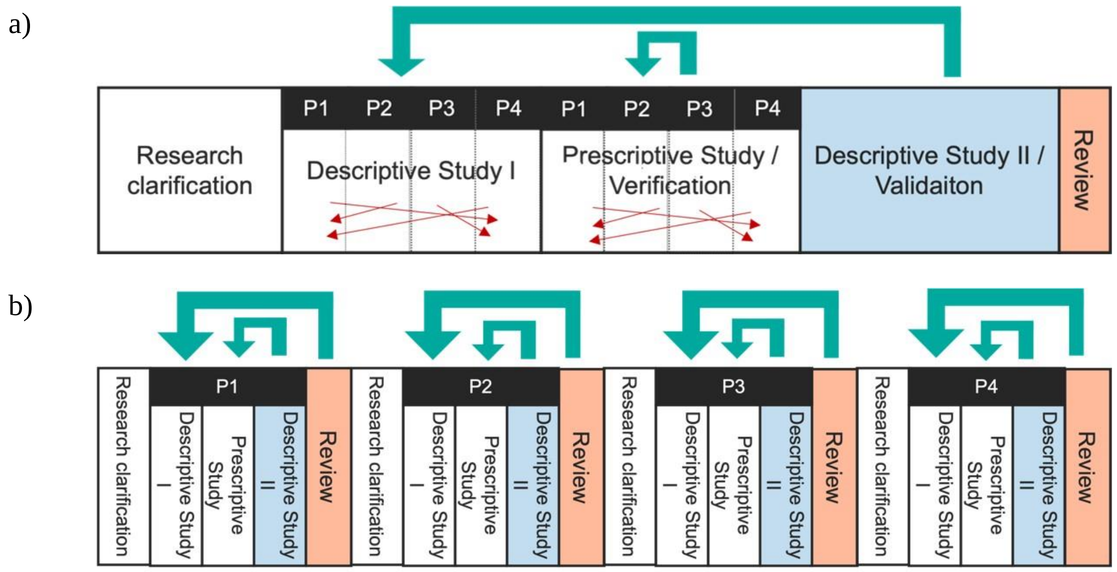


*Figure 1. Process architectures of a) "waterfall" DRM and b) hypothetical agile DR*

The two architectures shown in Figure 1 a-b are intended to represent two "extremes" of a hypothetical portfolio of research process architectures. Figure 1-a shows an example of "waterfall" DRM. Although DRM is intended to be iterative in nature (Blessing and Chakrabarti, 2009; p. 14), it often risks to follow a classic "waterfall" model for complex research projects (Eckert et al., 2003). Figure 1-a intends to model this situation. In this model, a set of research problems (P1, P2, P3, P4 in Figure 1-a) are identified after the Descriptive Study I. An important assumption made in this model is that due to the complexity of reality, these research problems are often inter-connected to each other. This assumption is also motivated by research in DRM (Blessing and Chakrabarti, 2009; p. 21), soft system methodology (Checkland and Scholes, 1999) and action research (Lewin, 1947). In Figure 1-a, these interconnections are represented symbolically as intersecting red arrows. In this research architecture, a set of requirements for each of these problems is established after the descriptive study (Blessing and Chakrabarti, 2009; p. 152), which leads the development of the design support for these problems (Prescriptive Study). During the prescriptive study, the researcher verifies the design support against the requirements. If requirements are not met, the design support is iterated. Another key assumption made in this model is that only after the Descriptive Study II (validation) that the real assessment about whether the support answers the research problems is conducted (see Isaksson et al., 2020 for the differences between method verification and method validation). If the support is not valid, the Descriptive Study I is conducted again (rework).
Figure 1-b shows an example of a hypothetical Agile DR process instead. The main difference with the "waterfall" DRM is that now 1) all the problems have been made independent from each other (as discussed in the next section) and 2) a whole DRM "loop" is conducted for each of these problems once at a time. These represent two key requirements for an agile DR process. This way of working is followed for example by research in geometry assurance and robust design (Lindau et al., 2016). In



https://doi.org/10.1017/pds.2023.393 Published online by Cambridge University Press

this research group, a software (RD&T, 2011) has actively been used over the years. This software-enabled research group does not follow traditional "waterfall" models with many long "semi-industrial testing" aiming at increasing TRL. Starting from the premise that *"customers do not know exactly what they want from the beginning, only roughly"*, test is conducted directly on end users (sometimes with a beta version released to a smaller group before releasing it to everyone).

It should be noted that Figure 1 a-b shows two "extreme" research process architectures, but the model can also support hybrid architectures (e.g., iteration between Prescriptive Study and Descriptive study). The same hybrid models are considered both in agile process simulation (Mitsuyuki et al., 2017) and when studying agile and design thinking processes in practice (Cocchi et al., 2021). It should also be noted that this paper focuses on the impact of research process architectures of time and risk of rework. For this purpose, looking only at these two extremes can already provide insights, although more a detailed break-down between several research process architectures - for example organization action research (Aguinis et al., 2009) and educational action research (Newton and Burgess, 2016) - is necessary yet left for future work.

## 3.2 Uncertain research conditions and lead time comparisons

Figure 2 shows the comparison in terms of lead time between the "Waterfall" DRM and Agile DR. To keep the model simple, the duration for the activities has been set equal for both the "waterfall" DRM and the hypothetical agile DR process (4 months each for all the four DRM phases, split in 4 for Agile DR). Therefore, the theorical total duration is the same (20 months). However, the performances of the research process are affected by rework and error propagation introduced by uncertain conditions that are part of the research process. The uncertain conditions are:

1. **Uncertain condition 1:** One of the research problems is very difficult to diagnose (Isaksson, 2016): this is modelled by increasing the duration of the Descriptive Study for Problem P2 by 2 months. This difference is made small on purpose, although in reality this delay can be much longer.
2. **Uncertain condition 2:** Developing the design support for one problem requires trial and error, because a new technique needs to be learned (e.g., programming): this is modelled by reworking the Prescriptive Study of P2.
3. **Uncertain condition 3:** The validation of the design support is not valid, and the whole Descriptive Study I need to be reconducted (Eckert et al., 2004; Isaksson et al., 2020): this is modelled by reworking all the Descriptive I, Prescriptive I and Descriptive II activities once again.

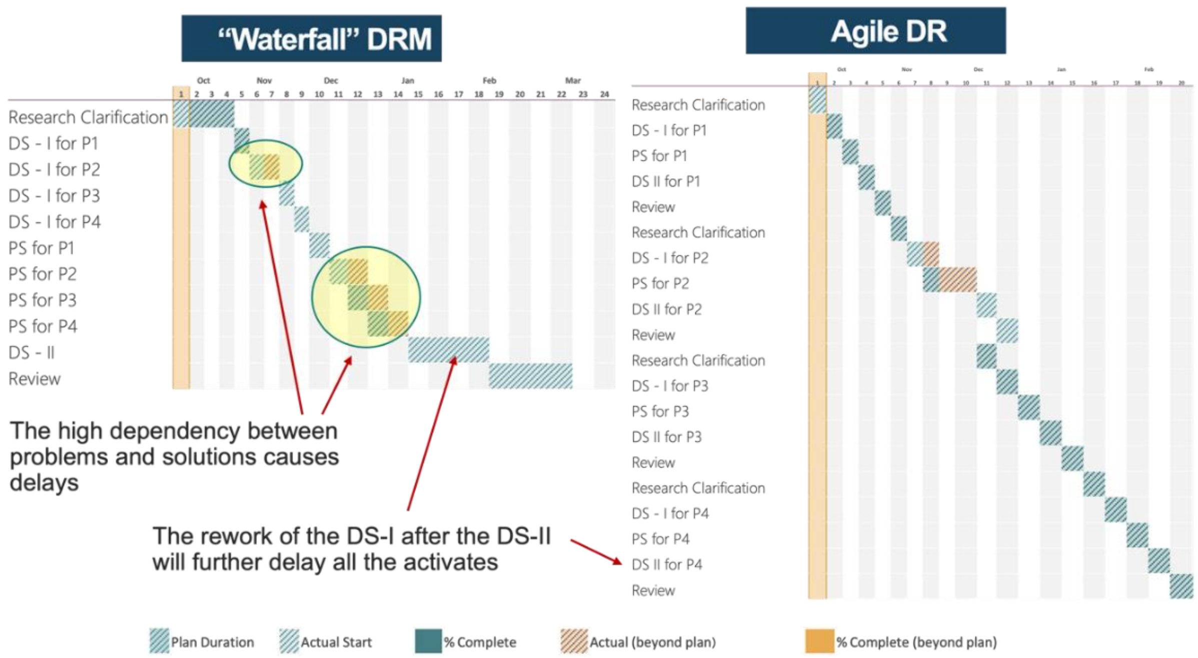


*Figure 2. Lead time comparison between "waterfall" DRM and Agile DR.*



https://doi.org/10.1017/pds.2023.393 Published online by Cambridge University Press

Figure 2 shows how, considering a small increase due to the uncertain Condition 1 and 2, Agile DR presents a lower lead time than the Waterfall DRM (20 months compared to 22 months) while still offering more iterations. This is because in the Waterfall DRM, the delay in the Descriptive Study I requires the Prescriptive Study to be delayed as well. Also, the high dependency between problems and solutions means that the trials and errors during the development of one "piece" of the Prescriptive Study are affecting the other "pieces" of the design support (i.e., to solve the other research problems). Agile DR instead, having made the problems and the solutions independent from each other, allows a concurrent development of the design supports.
What Figure 2 is not showing is the impact on the lead time caused by Condition 3. Agile DR, based on early demonstration, allows the validation of Descriptive Study II to be run for one design problem at the time. This means that eventual reworks are impacting only the specific DRM "loop" concerning that specific problem. In the waterfall DRM instead, validation is conducted at the end, which causes a re-run of Descriptive Study I. This rework causes delays in all the subsequent activities, and the Lean Time is impacted considerably (reaching month 30, in this simple example).

## 4 AGILE DESIGN RESEARCH: FIVE LIGHT-BUT SUFFICIENT RULES

Based on the agile focused assessment of existing research approaches and on observation made on several industry-academia research projects (detailed in the next section), this paper proposes five practical light-but sufficient rules to conduct "Agile Design Research" (Agile DR, Figure 3).

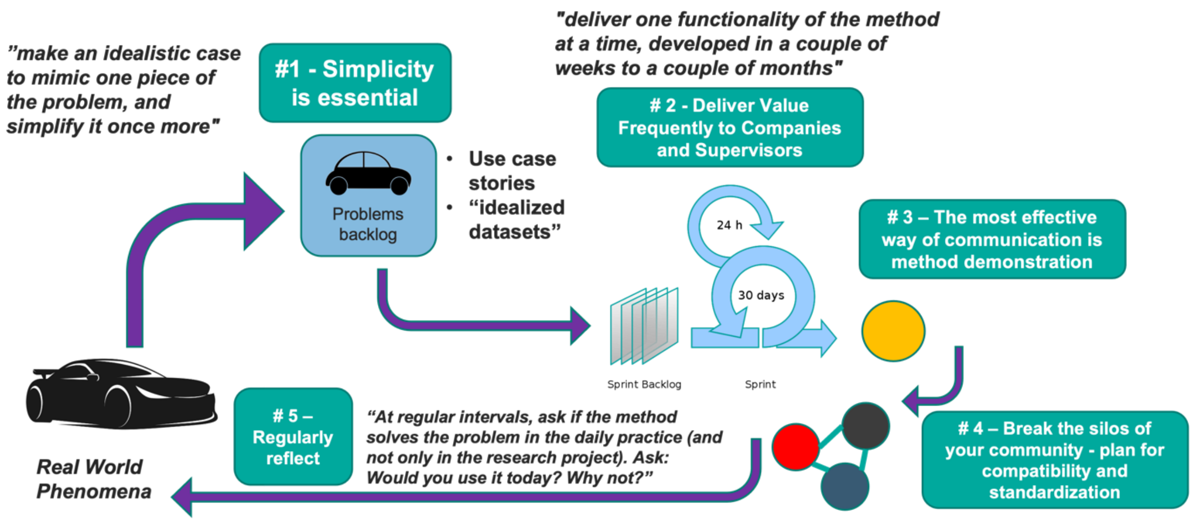


*Figure 3. Five light-but sufficient rules for Agile DR, mapped onto a scrum framework*

These rules are adapting the twelve original principles of agile development onto the DR context. The rules are mapped onto a traditional scrum framework (Dingsøyr et al., 2012), that should be read from left to right. This research has identified the following five rules:

1. ***Simplicity is essential:*** often, DR with industry is already conducted via "use cases" provided by the industrial partner, which intend to mimic realistic (yet "idealized") design situations (Wallin et al., 2014). The idea with this principle is to decompose further this idealized (yet complex) use case, in order to create a "problems backlog" to be solved. These problems should be as much as possible self-containing (i.e., ideally, they should be ***independent*** to other problems in the backlog) and represent only one piece of the real use case provided by the company. A crucial question is how this independence can be secured. In this context, the 'independence axiom' suggested by Suh (1995) in the design of mechanical system can result helpful and effective. For example, one useful exercise between supervisors and PhD students could be to assess whether the design problem is axiomatically interdepended from others, and – if dependent – to reformulate to ensure independence. Another useful exercise in to use Design Structure Matrix to control a limited amount of interfaces between design problems, as commonly applied in product modularity (Otto et al., 2016).
2. ***Deliver value frequently to companies and supervisors:*** from the problems backlog, the researcher engages with a "research sprint" to solve only one standalone problem, which results



https://doi.org/10.1017/pds.2023.393 Published online by Cambridge University Press

into one functionality of the method. Ideally, the prescriptive study should be conducted within 30 days (otherwise the problem needs to be further decomposed). Also, the researcher should commit to report about one small progress of the developed method within 24 hours, as a general rule of good project behaviour.

*3.* ***The most effective way of communication is method demonstration:*** this principle, reflecting the principle 6 and 7 of the agile manifesto, aims at fostering practical demonstration of the developed functionality of the method (rather than an extensive verbal or visual description). Such demonstration does not necessarily have to be a prototype, but it should convey the "working" mechanisms of the developed support (e.g., it could be an exercise or a roleplay).

4. ***Break the silos of the community:*** often, design methods are built in isolation (Gericke et al., 2020) which risk misunderstandings since the work is not built on each other (Vermaas, 2013). Therefore, this principle strives for planning compatibility to other methods, over the (sometimes natural) temptation to challenge other methods.

5. ***Frequently reflect:*** this rule is inspired by the principles 2 and 12 of the agile manifesto. Here, the reflection should be conducted at two levels. One is to reflect whether the problem being "idealized" and captured in the problems backlog is a relevant representation that the problem in the use case offered by the company in the research project. Second, the reflection should be focused on the relevance of the support for the *actual* design practice (beyond the use case and beyond the industrialists involved in the research project). A crucial question here is not only whether the method supports the problem, but also ask openly whether the users will use it in their daily practice, and why not. This seems an obvious reflection to be made, but it is very difficult to accept failure in a publish-or-perish culture of academia (Gericke et al., 2020). Therefore, design methods are often validated in ideal scenarios, and reflections against the proposed method are left out. The idea with Agile DR is that failure can be accepted, since it is conducted in shorter loops.

It is important to note that these principles should be considered a complement to established research methodologies such as DRM (this is the reason behind the term "framework"), emphasizing some light-but-sufficient rules of project behaviour to keep momentum, motivation and trust by companies (principle 5 and 11 of the agile manifesto).

## 5 APPLYING AN AGILE DESIGN RESEARCH "SPRINT" IN PRACTICE

This section intends to demonstrate the application of the five rules in a "research sprint" within a project with industry conducted by researchers from a Swedish university. Amongst others, the case chosen is derived from a research project (Fernández et al., 2020) because it provides the core gist of an Agile DR sprint. The project is conducted in collaboration between the university and two automotive OEMs (one manufacturing cars and the other manufacturing trucks) and is focused on trading off between the flexibility of a product platform when introducing new technologies. Figure 4 shows an example of how an agile DR "sprint" was applied in this project. The sprint lasted approximately one month.

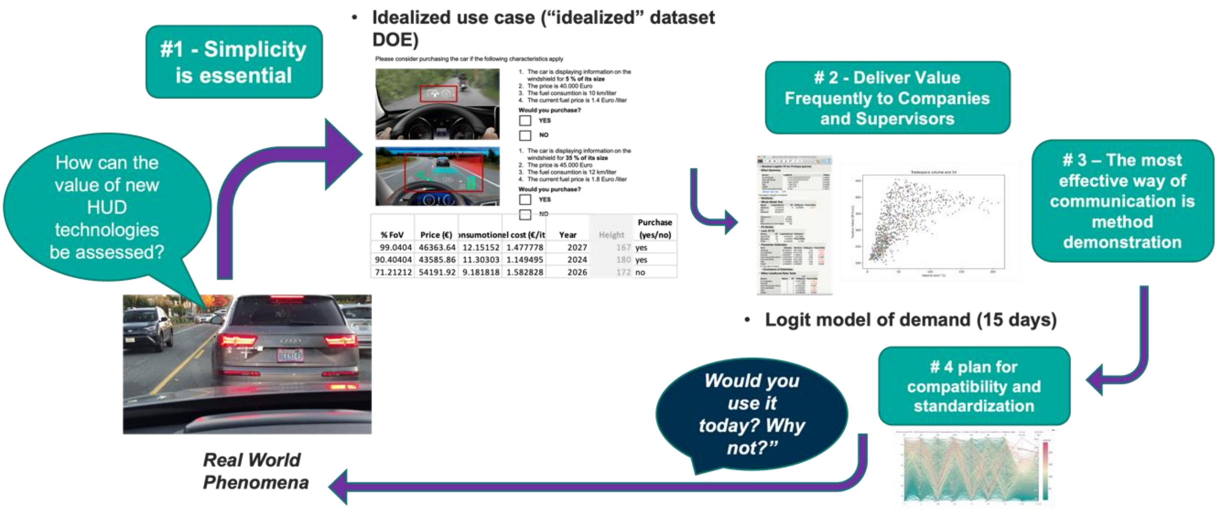


*Figure 4. Application of agile AR in the head-up display (HUD) case study.*



https://doi.org/10.1017/pds.2023.393 Published online by Cambridge University Press

The research sprint was related to the assessment of Head-Up Display (HUD) technology alternatives in different automotive platforms. The Head-Up Display (HUD) provides information such as speed, cruise control functions, navigation, traffic sign information (bottom left of Figure 4). The information is projected at the base of the windshield in the driver's field of vision. The purpose of this feature is to see information without averting the driver's eyes from the road by needing to look down at the instrument panel.

From interviews with practitioners, it emerged that one big problem when integrating new technologies in the platform is to evaluate the value (or utility) of new technology alternatives. In the case of the HUD, the performance of the system is intimately linked to the space available. Basically, it can deliver a larger and more accurate image to the driver if the components are bigger. Therefore, increasing the performance of the system needs to be balanced against the space available as well as the performance and cost of the components around the HUD (e.g., more space could be made available if a different steering system was used, such as a steer-by-wire (SBW) system). The problem with making these trade-offs is that: 1) it is difficult to know quantitatively the customer preferences regarding these trade-offs (e.g., a bigger image vs. an increase in price) and 2) the technologies for a HUD may be very different, therefore it is difficult to model the performances for each technology and 3) designers are working at product planning level, therefore there is only a limited knowledge about the desired system (e.g., no detailed CAD model is available).

As it can be sensed by the outcome of this early Research Clarification and Descriptive Study I (in DRM terms), all these problems are very interconnected with each other, and attempting to solve all of them at once would have implied a long Prescriptive Study with many risks of "ripple effects" and changes.

Instead, the Agile DR sprint began by idealizing and decomposing one problem (rule #1, Figure 4). The problem chosen is how to know quantitatively the customer preferences regarding product trade-offs. However, to make this problem independent from the others, further idealizations were made. It was assumed that all the performances of HUD technologies are known, as well and their costs. It was also assumed that 100 customers have filled in an "ideal" discrete-choice questionnaire were images with sizes of information displayed were shown, and the customers have chosen whether they would buy that car or not (therefore, a binary choice), depending on the image size and other attributes (e.g., the car price, the fuel cost etcetera). This approach is already followed in other research about perceived quality (Stylidis et al., 2020). Further, it was idealized that the survey was designed to vary all these attributes according to a Latin Hypercube Design of Experiment technique (LHD, Wang, 2003). The answers to this idealized survey were provided by the researchers themselves, with the only purpose of demonstrating the method to the industrial partners in a concrete fashion. Therefore, an "idealized" dataset was used to develop a demand model of future cars, based on a logistic regression (Wassenaar and Chen, 2003). This model was built in 15 days. Although not statistically relevant, the quantitative model developed during this "sprint" allowed to quickly demonstrate the envisioned method to the industrial partners (rule #2 and #3) to gain feedback. The whole point of this sprint was to keep momentum with the company, while reducing the time for rework. At the same time, integration with other solutions (developed by by other researchers) was coordinated in parallel (e.g., to solve the problem of modelling the performances for each technology). Thanks to the "independence" of the research problems, only a few interfaces and data structures needed to be defined and standardized (rule #4).

During the reflection stage (rule #5), the companies did not question the statistical relevance of the model, rather appreciated the quick iteration loop with a model that was already interactive and "working". Their reflection was more focused on the usefulness of the method and ways to improve it. For example, it was mentioned the fact that such a model should include a way to consider the cost of redesigning the platform in the case of the technology was outside the space reservation available by the product platform (which negatively impact the whole profit for the company). Questions about how to make the model statistically relevant and how to design such surveys were also raised, nevertheless the companies validated this method "increment". The researchers then noted the reflections made and included these new research problems in the backlog (after ensuring again independence).

## 6 CONCLUDING DISCUSSION

The HUD case study highlights how Agile DR can be a complement to established research methodologies (such as DRM). Agile DR aims at keeping momentum, motivation and trust by


https://doi.org/10.1017/pds.2023.393 Published online by Cambridge University Press

companies, by adopting five light-but-sufficient rules of project behaviour during the different DRM stages. Also, the fact that Agile DR tends to make “shortcuts” (such as the “idealized” dataset with survey responses) does not undermine scientific rigorousness. The whole point here is that such idealizations are made to achieve rigorous and relevant results over time, while eliminating the time spent on change and rework (that Agile DR accepts as a natural part of the research process). This resonates well with the principle 7 of the agile manifesto (*“working software is the primary measure of progress”*) which aims at faster iteration loops, problem decompositions and practical demonstration to promote incremental development.

While this way of working is industrially accepted (especially in the software sector), it requires a certain degree of mental flexibility to be adopted in a scientific field such as DR. Therefore, this paper offered a pragmatic discussion for PhD students, supervisors, researchers and practitioners to find common ground through five “rules of the game” that can be used to structure the discussion about whether and when it is sound to take shortcuts towards scientific rigorousness, industrial impact and research output (e.g., number of publications).

## ACKNOWLEDGMENTS

The funding for this work was provided by the VISP project, co-funded by the Swedish Governmental Agency for Innovation Systems (VINNOVA), Volvo Cars Group (VCG) and Volvo Group Trucks Technology (VGTT).

## REFERENCES

Aguinis, H., Pierce, C.A., Bosco, F.A. and Muslin, I.S., 2009. First decade of Organizational Research Methods: Trends in design, measurement, and data-analysis topics. Organizational Research Methods, 12(1), pp.69-112.

Bajpai, S., Eppinger, S.D. and Joglekar, N.R., 2019. The structure of agile development under scaled planning and coordination. In DS 97: Proceedings of the 21st International DSM Conference (DSM 2019), Monterey, California, September 23rd-25th 2019 (pp. 1-10).

Barth, A., Caillaud, E. and Rose, B., 2011. How to validate research in engineering design?. In DS 68-2: Proceedings of the 18th International Conference on Engineering Design (ICED 11), Impacting Society through Engineering Design, Vol. 2: Design Theory and Research Methodology, Lyngby/Copenhagen, Denmark, 15.-19.08. 2011 (pp. 41-50).

Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R. and Kern, J., 2001. Manifesto for agile software development.

Blessing, L.T. and Chakrabarti, A., 2009. DRM: A design reseach methodology (pp. 13-42). Springer London.

Brown, T., 2008. Design thinking. Harvard business review, 86(6), p.84.

Browning, T.R. and Eppinger, S.D., 2002. Modeling impacts of process architecture on cost and schedule risk in product development. IEEE transactions on engineering management, 49(4), pp.428-442.

Cantamessa, M., 2003. An empirical perspective upon design research. Journal of Engineering Design, 14(1), pp.1-15.

Cash, P., Isaksson, O., Maier, A. and Summers, J., 2022. Sampling in design research: Eight key considerations. Design studies, 78, p.101077.

Checkland, P. and Scholes, J., 1990. Soft systems methodology in action, John Wileys Sons Inc.

Cocchi, N., Dosi, C. and Vignoli, M., 2021. The Hybrid Model MatrixEnhancing Stage-Gate with Design Thinking, Lean Startup, and Agile: Managers can use the Hybrid Model Matrix to decide when to use design thinking, Lean Startup, or Agile with Stage-Gate to boost new product development. Research-Technology Management, 64(5), pp.18-30.

Cockburn, A., 2006. Agile software development: the cooperative game. Pearson Education.

Cross, N., 2018. Developing design as a discipline. Journal of Engineering Design, 29(12), pp.691-708.

Da Silva, T.S., Martin, A., Maurer, F. and Silveira, M., 2011, August. User-centered design and agile methods: a systematic review. In 2011 AGILE conference (pp. 77-86). IEEE.

Dingsøyr, T., Nerur, S., Balijepally, V. and Moe, N.B., 2012. A decade of agile methodologies: Towards explaining agile software development. Journal of systems and software, 85(6), pp.1213-1221.

Douglass, B.P., 2015. Agile systems engineering. Morgan Kaufmann.

Eckert, C., Stacey, M. and Clarkson, P.J., 2004. The lure of the measurable in design research. Design Society.

Eckert, C.M., Stacey, M.K. and Clarkson, P.J., 2003. The spiral of applied research: A methodological view on integrated design research.

Fernández, I.A., Panarotto, M. and Isaksson, O., 2020, May. Identification of technology integration challenges at two global automotive OEMs. In Proceedings of the Design Society: DESIGN Conference (Vol. 1, pp. 2245-2254). Cambridge University Press.

Gericke, K., Eckert, C., Campean, F., Clarkson, P.J., Flening, E., Isaksson, O., Kipouros, T., Kokkolaras, M., Köhler, C., Panarotto, M. and Wilmsen, M., 2020. Supporting designers: moving from method menagerie to method ecosystem. Design Science, 6, p.e21.
Gericke, K., Kramer, J. and Roschuni, C., 2016. An exploratory study of the discovery and selection of design methods in practice. Journal of Mechanical Design, 138(10), p.101109.
Guertler, M.R., Kriz, A. and Sick, N., 2020. Encouraging and enabling action research in innovation management. R&D Management, 50(3), pp.380-395.
Hernandez, R., Kreye, M. and Eppinger, S., 2019, June. Applicability of agile and scrum to product-service systems. In EurOMA Conference (pp. 1-10).
Isaksson, O., 2015. A Collaborative Engineering Design Research Model—An Aerospace Manufacturer's View. In Impact of Design Research on Industrial Practice: Tools, Technology, and Training (pp. 363-381). Cham: Springer International Publishing.
Isaksson, O., Eckert, C., Panarotto, M. and Malmqvist, J., 2020, May. You need to focus to validate. In Proceedings of the Design Society: DESIGN Conference (Vol. 1, pp. 31-40). Cambridge University Press.
Jagtap, S., Warell, A., Hiort, V., Motte, D. and Larsson, A., 2014. Design methods and factors influencing their uptake in product development companies: a review. In DS 77: Proceedings of the DESIGN 2014 13th International Design Conference.
Kapurch, S.J. ed., 2010. NASA systems engineering handbook. Diane Publishing.
Keijzer-Broers, W.J. and de Reuver, M., 2016. Applying agile design sprint methods in action design research: prototyping a health and wellbeing platform. In Tackling Society's Grand Challenges with Design Science: 11th International Conference, DESRIST 2016, St. John's, NL, Canada, May 23-25, 2016, Proceedings 11 (pp. 68-80). Springer International Publishing.
Lewin, K., 1946. Action research and minority problems. Journal of social issues, 2(4), pp.34-46.
Lewin, K., 1947. Frontiers in group dynamics: II. Channels of group life; social planning and action research. Human relations, 1(2), pp.143-153.
Lindau, B., Lorin, S., Lindkvist, L. and Söderberg, R., 2016. Efficient contact modeling in nonrigid variation simulation. Journal of Computing and Information Science in Engineering, 16(1).
Maier, J.F., Wynn, D.C., Biedermann, W., Lindemann, U. and Clarkson, P.J., 2014. Simulating progressive iteration, rework and change propagation to prioritise design tasks. Research in Engineering Design, 25, pp.283-307.
Mårtensson, P., Fors, U., Wallin, S.B., Zander, U. and Nilsson, G.H., 2016. Evaluating research: A multidisciplinary approach to assessing research practice and quality. Research Policy, 45(3), pp.593-603.
Martin, J.N., 2020. Systems engineering guidebook: A process for developing systems and products. CRC press.
Mitsuyuki, T., Hiekata, K., Goto, T. and Moser, B., 2017. Evaluation of project architecture in software development mixing waterfall and agile by using process simulation. Journal of Industrial Integration and Management, 2(02), p.1750007.
Mumford, E., 2001. Advice for an action researcher. Information Technology & People, 14(1), pp.12-27.
Newton, P. and Burgess, D., 2016. Exploring types of educational action research: Implications for research validity. In The Best Available Evidence (pp. 33-46). Brill.
Otto, K., Hölttä-Otto, K., Simpson, T.W., Krause, D., Ripperda, S. and Ki Moon, S., 2016. Global views on modular design research: linking alternative methods to support modular product family concept development. Journal of Mechanical Design, 138(7), p.071101.
RD&T, 2011, RD&T Manual, Version 1.13, Gothenburg, Sweden. Retrieved online 2023-04-03: http://rdnt.se/software-tool/
Stylidis, K., Wickman, C. and Söderberg, R., 2020. Perceived quality of products: a framework and attributes ranking method. Journal of Engineering Design, 31(1), pp.37-67.
Suh, N.P., 1995. Axiomatic design of mechanical systems.
Vermaas, P.E., 2013. The coexistence of engineering meanings of function: four responses and their methodological implications. AI EDAM, 27(3), pp.191-202.
Wallin, J., Isaksson, O., Larsson, A. and Elfström, B.O., 2014. Bridging the gap between university and industry: Three mechanisms for innovation efficiency. International Journal of Innovation and Technology Management, 11(01), p.1440005.
Wang, G.G., 2003. Adaptive response surface method using inherited latin hypercube design points. J. Mech. Des., 125(2), pp.210-220.
Wassenaar, H.J. and Chen, W., 2003. An approach to decision-based design with discrete choice analysis for demand modeling. J. Mech. Des., 125(3), pp.490-497.
Wynn, D.C. and Eckert, C.M., 2017. Perspectives on iteration in design and development. Research in Engineering Design, 28, pp.153-184.



https://doi.org/10.1017/pds.2023.393 Published online by Cambridge University Press