Problem decomposition: the key to Agile sprints in Design Research

Massimo Panarotto, Arindam Brahma, Ola Isaksson

Chalmers University of Technology, Gothenburg, Sweden

Abstract: In design research, parallelization and anticipation of tasks are crucial to having an effective impact on industry and society. In practice, however, parallelization and anticipation are difficult due to high inter-dependency among research problems and design supports. This paper proposes a framework to use the Design Structure Matrix combined with Axiomatic Design to isolate research problems and their design supports. This modular independence is an important building block to plan for quick and parallel agile design research "sprints". The application of the approach is illustrated through two practical examples. The results highlight the importance to control a limited number of interfaces to work "agile" in design research without wasteful rework.

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

1 Introduction

There is still a debate about how to 'use' Design Research (DR) to have an effective impact on the industry (Gericke et al., 2016; Cross, 2018). One of the reasons for the debate lies in the ways of working within DR, which tends to "slow down" the research process thereby reducing the opportunity for impact (Gericke et al., 2020). When engaging with the industry, design researchers often 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 include Action Research (AR; Lewin, 1946) and Design Research Methodology (DRM; Blessing and Chakrabarti, 2009).

In principle, DR is intended to be an iterative, learning-focused process, looping in between the description of a design problem, the prescription of a design support and the validation of the design support on the problem identified. In DR, the focus on iteration is essential. For example, the authors of DRM suggest that (it) *"…is not to be interpreted as a set of stages and supporting methods to be executed rigidly and linearly"* (Blessing and Chakrabarti, 2009; p. 17). For this reason, DRM emphasizes the need to plan stages to be partly executed in parallel. For example, the evaluation of a support (Descriptive Study II) is suggested to be conducted during and not after the development of the support itself (Prescriptive Study). The research plan should therefore focus on identifying which parts of the support can be developed and evaluated concurrently to speed up the research process.

In practice, however, the complexity arising from reality makes research problems highly interdependent to each other and to contextual factors. This makes parallelization (and anticipation) of research tasks very difficult in practice, as it is difficult to isolate research problems and their design supports. This difficulty in ensuring independence risks that 1) DR follows a classic "waterfall" model, hindering iteration and learning (Eckert et al., 2003) or 2) brings a considerable amount of rework and delay, which may not be conducive to strict timelines of such research projects.

It is therefore arguably important to find out how DR can be made more effective and iterative. For example, Panarotto et al (2023) proposed a few practical guidelines to conduct intervention-based DR inspired by "agile development", drawing from the works such as Beck et al. (2001). This set of guidelines, called "Agile Design Research" (Agile DR), is based on two main principles 1) the research problems are decomposed and made *as independent as possible* from each other, so that a whole DRM "loop" can be conducted for one problem at a time. A logical question consequently arises about how such *independence* among the research problems can be achieved.

The main premise of this paper is that applying an agile approach to design research benefits from thoughtful decomposition into independent problems that can be addressed in sprints. Therefore, this paper proposes an approach using Design Structure Matrices (DSMs) in conjunction with the 'independence axiom' suggested by Suh (1995) to control a limited amount of interfaces between design problems, as also commonly applied in product modularity (Otto et al., 2016).

2 Background

2.1 Methodologies of design research in literature

In one of the many early works on developing research methodology specifically for engineering, Antonsson (1987) argued for the use of scientific methods in design research. The method involved six steps, starting with a proposal that a set of rules for design that can explain a part of the design process, followed by the development of those rules. The third step was to have novice designers learn those rules and apply them in practice while measuring their design productivity in the fourth step. The last two steps involved the evaluation of the results to confirm or refute the hypothesis, and its refinement respectively. Although the process is seemingly sequential, Antonsson points out that a hypothesis itself requires huge amounts of exploratory research. Much later Duffy and O'Donnell (1999), as a part of a wider design research approach proposed a research methodology consisting of six steps. Though the methodology provides a high-level approach to design research, the authors did not go into detail or discuss cases where steps have to be repeated or modified when required. The sub-topic of validation has its own set of literature. The works of Pedersen et al., (2000) and Seepersad et al., (2006) are examples of such focused work.

Of specific interest to this paper, and also one of the most comprehensive and widely used design methodologies is the DRM by Blessing and Chakrabarti (2009). The methodology was first published in 1992 (Blessing et. al., 1992) and has appeared in many other publications over the years. Another methodology often used in design research is Eckert et al., (2003)'s eightfold path method. The method aimed at taking a wider view of how individual research projects converge towards a bigger agenda. The methodology encompasses eight steps, or a logical circle, which are - (1) Empirical studies of design behaviour, (2) Evaluation of empirical studies, (3) Development of theory, (4) Evaluation of theory, (5) Development of tools and procedures, (6) Evaluation of tools and procedures, (7) Introduction of tools and procedures, and (8) Evaluation of Dissemination. As far as individual projects go, the authors mention that normally they do not go beyond a few steps, and as a whole, unlike a research group which would have a bigger agenda. Eckert et al., (2003) also compare their methodology with DRM, and comment that *"…the concept of criterion is very rigid in DRM, and in our opinion too restrictive"*, and that *"early fixation on measurable criteria can lead the researcher to miss the real issues by selecting over-specific methods."*

2.1 Agile Design Research

Based on the agile-focused assessment of existing research approaches and observations made on several industry-academia research projects, Panarotto et al. (2023) proposed "Agile Design Research" (Agile DR, Figure 1) to keep momentum, motivation and trust when doing research with industry (while preserving scientific rigour).

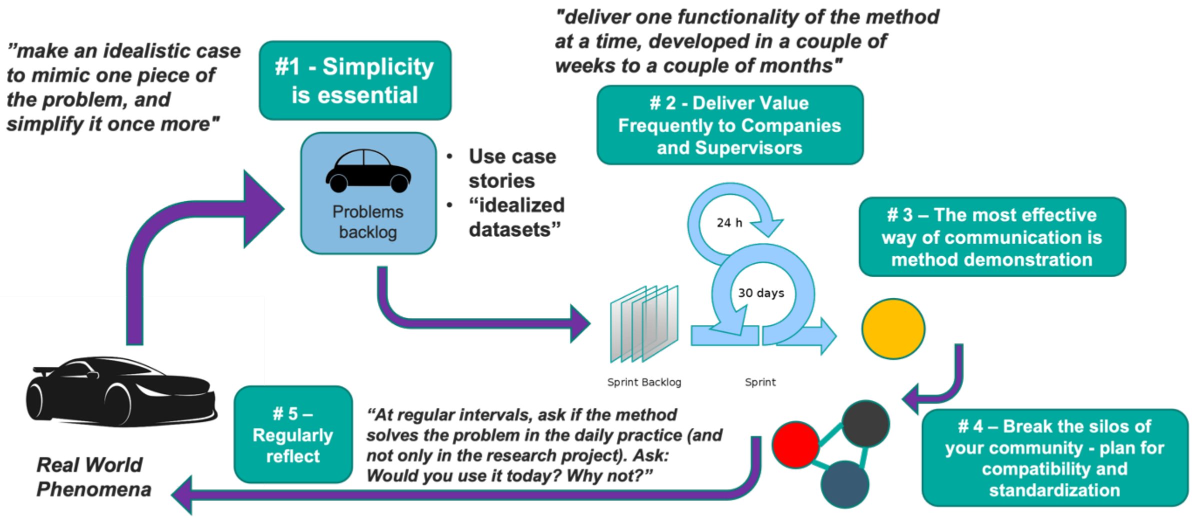

Figure 1. Agile DR (Panarotto et al., 2023), mapped onto a scrum framework

Agile DR is intended to be a complement to established research methodologies (such as DRM) through five light-but-sufficient rules of project behaviour. These five rules are adapting the twelve original principles of agile development (Beck et al., 2001) to the DR context. The rules are mapped onto a traditional scrum framework (Dingsøyr et al., 2012), that should be read from left to right. The main difference with more "waterfall" approaches is that in Agile DR 1) all the problems have been made independent from each other and 2) a whole DRM "loop" is conducted repeatedly for each of these problems one at a time. Agile DR, based on early demonstration and problem independence, allows the validation of problem-specific design supports to be run for one design problem at a time, and eventual reworks are impacting only the specific DRM "loop" concerning that specific problem. In more "waterfall" approaches instead, validation is conducted at the end, which causes delays and reworks if changes and uncertainties in the research process arise.

This paper focuses particularly on Principle #1 of Agile DR *"Simplicity is essential"*. The idea of this principle is to decompose the complex real-world problem and create a "problems backlog" of independent research problems to be solved. Such independence is fundamental to applying the other four principles. For example, without independence, a design research sprint cannot be run without consequences. Once the piece of functionality developed is considered valid and integrated into the whole design support, it may have created dependencies for the next piece of functionality (once the new research problem is taken from the backlog and resolved in a new sprint). To ensure independence among research problems, this paper proposes to use the 'independence axiom' suggested by Suh (1995). The next section briefly summarizes this axiom.

2.1 The Independence Axiom

A central concept in Axiomatic Design is the design matrix that represents the relationship between the design parameters, (DPs) and the functional requirements (FRs). The relationship can be written as:

$$\begin{pmatrix} FR1 \\ FR2 \end{pmatrix} = [A] \begin{pmatrix} DP1 \\ DP2 \end{pmatrix} \quad (1)$$

Where;

$$\begin{pmatrix} FR1 \\ FR2 \end{pmatrix} = \begin{bmatrix} A11 & A12 \\ A21 & A22 \end{bmatrix} \begin{pmatrix} DP1 \\ DP2 \end{pmatrix} \quad (2)$$

Therefore, in a design, there is often an interdependence between functional requirements and design parameters. The *independence axiom* states that a better design can be achieved if a) a functional requirement (FR) is independent or uncoupled in their own right and b) Design parameters (DP) are maintaining the FR independent – i.e., not making FR coupled. Following this axiom means uncoupling FRs and DPs through reformulation. An example is shown in Figure 2.

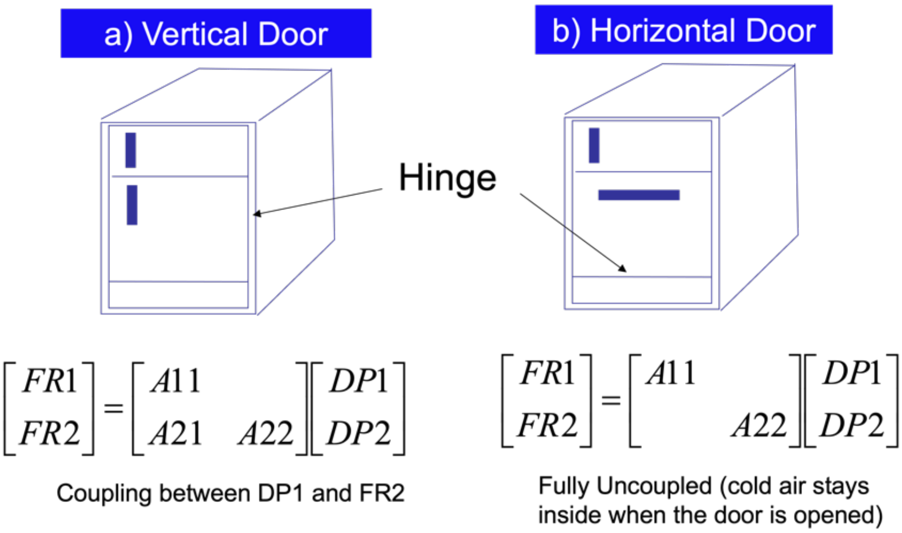

Figure 2. Independence Axiom example (adapted from Mabrok et al., 2015)

Consider the case of a refrigerator (Mabrok et al., 2015) that must fulfil the following Functional Requirements:

FR1 = Provide access to items stored in the refrigerator

FR2 = Minimize energy loss

These functional requirements could be fulfilled by the following Design Parameters (Figure 2 – a):

DP1 = Vertical door

DP2 = Thermal insulation material in the door

Looking at the dependency matrix (Figure 2 – a), one can observe that there is a coupling between the DP1 (vertical door) and the FR2 (Minimize energy loss). When opening the door, the cold air is dispersed. A more uncoupled design can be achieved with a horizontal door (Figure 2 - b). In this case, when opening the door (to fulfil FR1), the cold air stays inside (therefore, FR2 is not affected). This example shows how designers can apply the independence axiom to resolve common mistakes that happen in conceptual design (Suh, 1995), for example, the risk of coupling due to an insufficient number of DPs or the tendency of concentrating on symptoms rather than cause (i.e., not focusing enough on FRs).

The independence Axiom has been used in a wide range of applications and in different ways. It is therefore of interest to investigate how it can be applied to Agile Design Research. The following section will provide two examples.

3 Applying the Independence Axiom to Design Research: Examples

3.1 Point-based vs Set-based Concurrent Engineering

The first example is related to Set-Based Concurrent Engineering (SBCE; Sobek et al., 1999), an approach that focuses on improving the efficiency and effectiveness of the product development process. Originating at the Toyota Motor Corporation in the early 90s, the approach deviates from traditional point-based concurrent engineering (PBCE) approaches, which focus to accelerate development by iterating on a single design concept (decided as early as possible) multiple times throughout the development cycle. In contrast, SBCE promotes a more exploratory (and potentially wasteful; Sobek et al., 1999) approach to design. Instead of prematurely converging on a single concept, multiple design alternatives (or "sets") are simultaneously explored and developed. These sets are often diverse and represent a range of possible solutions, ensuring a comprehensive exploration of the design space. SBCE can be considered a good example of the “independence axiom” applied to Design Research (Table 1).

Table 1. Comparison of PBCE and SBCE in terms of Axiomatic Design

	Point-based Concurrent Engineering (PBCE)	Set-based Concurrent Engineering (SBCE)
Research Problems	RP1 = Increase development Efficiency RP2 = Increase Probability to Meet Design Requirements	RP1 = Reduce Risk of Radical Changes in the Design RP2 = Base Decisions on Mature Design Input
Design Supports	DS1 = Converge quickly to an initial design solution DS2 = Iterate frequently Design solution	DS1 = Communicate Ranges of solutions DS2 = Delay decisions
Axiomatic Design Matrix	$\begin{pmatrix} RP1 \\ RP2 \end{pmatrix} = \begin{bmatrix} A11 & A12 \\ A21 & A22 \end{bmatrix} \begin{pmatrix} DS1 \\ DS2 \end{pmatrix}$	$\begin{pmatrix} RP1 \\ RP2 \end{pmatrix} = \begin{bmatrix} A11 & 0 \\ 0 & A22 \end{bmatrix} \begin{pmatrix} DS1 \\ DS2 \end{pmatrix}$

To adapt Axiomatic Design to Design Research, a slight change in notation is made. Functional Requirements are substituted by Research Problems (RPs) and Design Parameters become Design Supports (DSs) instead. Analysing PBCE, one can observe that two main research problems are supported by two design supports. Converging quickly to a design solution (DS1) is intended to increase development efficiency (RP1), while the probability of meeting design requirements (RP2) is achieved through frequent iterations of the design solution (DS2). Looking at the Axiomatic Design Matrix, however, it can be observed how there are negative inter-dependencies among the design supports and the research problems. The quick convergence to a design solution (DS1), while on one hand increases efficiency, on the other hand it also decreases the probability of meeting design requirements if the team has started from the “wrong” point in the design space (Sobek et al., 1999). Therefore, DS1 negatively affects RP2. At the same time, frequent design iterations (DS2) may negatively impact the design process efficiency (RP1) if iterations become large modifications (i.e. rework; Maier et al., 2014). In the axiomatic design matrix, these dependencies are visualized as A12 and A21.

PBCE can be seen as an example where the focus is put on symptoms rather than cause (i.e., not focusing enough on the research problems). A different formulation can be made in the case of SBCE. The cause for a poor developmental efficiency is often due to the risk of radical changes if the team has started from the “wrong” point in the design space. This risk can be reduced if a wide range of solutions are communicated (DS1). Another cause affecting the probability of the design meeting the design requirements is that often decisions are made based on a poorly matured knowledge of both the requirements and the design itself (Sobek et al., 1999). This problem can be solved by delaying the decision about a design until a more mature knowledge base has been built (DS2). Looking at the axiomatic design matrix, one can observe how the inter-dependencies among the Design Supports and the Research Problems have been removed (or minimized). Communicating Ranges of solutions (DS1) does not negatively impact the need to base decisions on mature design input (RP2). At the same time, delaying decisions (DS2) does not negatively impact the risk of radical changes in the design (RP1). Rather, it can support it.

3.2 Accounting for the inter-dependencies between research problems and design supports

Table 1 is showing the inter-dependencies among research problems and design supports for SBCE (the off-diagonal elements of the axiomatic matrix, e.g. A12 and A21) as "0". In practice, some inter-dependency can still exist (although reduced compared to PBCE). One possible way to quantify the magnitude of inter-dependencies is to consider the risk of change that one design support can bring to another research problem than the one it is intended to support. Following Clarkson et al., (2004) risk of change can be defined as:

$$Risk\ of\ Change\ =\ Likelihood\ of\ Change\ (L)\ \times\ Impact\ of\ Change\ (I) \quad (3)$$

Where the L-I values are assigned according to a 0-1 scale. Adapted in the Agile DR context, the inter-dependence of a research problem caused by another design support is written as:

$$A_{i,j}\ = (1 - Design\ Support\ Comprehensiveness)\ \times\ (1 -\ Design\ Support\ Timeliness) \quad (4)$$

Where *(1-Design Support Comprehensives)* substitutes the likelihood of change and *(1- Design Support Timeliness)* substitutes the impact of change. The main objective in Agile DR is therefore to reduce inter-dependency by delivering a comprehensive problem-specific design support as early as possible. In practice, however, there is always a trade-off between the two. To comprehend the meaning of these metrics, the PBCE vs. SBCE example is shown again (Table 2).

Table 2. Impact of Design Support timeliness and comprehensiveness on the inter-dependencies among research problems and design solutions

	Point-based Concurrent Engineering (PBCE)	Set-based Concurrent Engineering (SBCE)
Design Support Comprehensiveness	DS1, RP2 = 0.1 DS2, RP1 = 0.1	DS1, RP2 = 0.8 DS2, RP1 = 0.8
Design Support Timeliness	DS1, RP2 = 0.6 DS2, RP1 = 0.6	DS1, RP2 = 0.4 DS2, RP1 = 0.4
Axiomatic Design Matrix	$\begin{pmatrix} RP1 \\ RP2 \end{pmatrix} = \begin{bmatrix} X & 0.36 \\ 0.36 & X \end{bmatrix} \begin{pmatrix} DS1 \\ DS2 \end{pmatrix}$	$\begin{pmatrix} RP1 \\ RP2 \end{pmatrix} = \begin{bmatrix} X & 0.12 \\ 0.12 & X \end{bmatrix} \begin{pmatrix} DS1 \\ DS2 \end{pmatrix}$

PBCE is focused on delivering early a design solution (high design support timeliness, 0.6), yet not comprehensive (0.1). SBCE is focused instead on delaying decisions (therefore the timeliness is supposed to be lower than PBCE, 0.4). However, the comprehensiveness is supposed to be higher, since decisions are based on a mature basis and the higher number of solutions delivered provides a "buffer" to avoid the risk of radical changes in the design. Therefore, the comprehensiveness is set to 0.8. The overall inter-dependency value becomes then 0.36 for PBCE, and 0.12 for SBCE.

This simple example is intended to highlight the role of timeliness and comprehensiveness in Agile Design Research, and how the researchers can work with trading-off between the two. The next example will show how timeliness and comprehensiveness play a role in a more complex Design Research set-up.

2.3 Second example: evaluating supplier readiness in highly integrated aero-engine components

This example is inspired by a project conducted in collaboration with a manufacturer of aero-engine components (Panarotto et al., 2022). The main research problem identified by the company was to evaluate supplier readiness in highly integrated aero-engine components. Figure 3 highlights how this problem was divided into two sub-problems to be supported. The two problems identified are *RP1: Decompose hardware-supplier relationships* and *RP2: Compute supplier readiness.* In the project, two researchers were involved in finding a support for these problems. Figure 3 highlights how this design situation impedes the parallel "sprint" of the two researchers. Due to the complexity of the problem, Researcher 1 cannot deliver a comprehensive support early. For example, the studied components are highly integrated products, meaning that all required functions are satisfied by a single, monolithic component – making it more difficult to find a comprehensive decomposition method (e.g., classical methods based on physical decomposition cannot be applied in this context). Therefore, the timeliness and comprehensiveness of Design Support 1 is low, making the support of Research Problem 2 highly dependent on the Design Support 1. At the same time, Researcher 1 is impacted by the computation method identified by Researcher 2. This high inter-dependency (visualized by the red arc in Figure 3) hinders both researchers to make their individual "sprints".

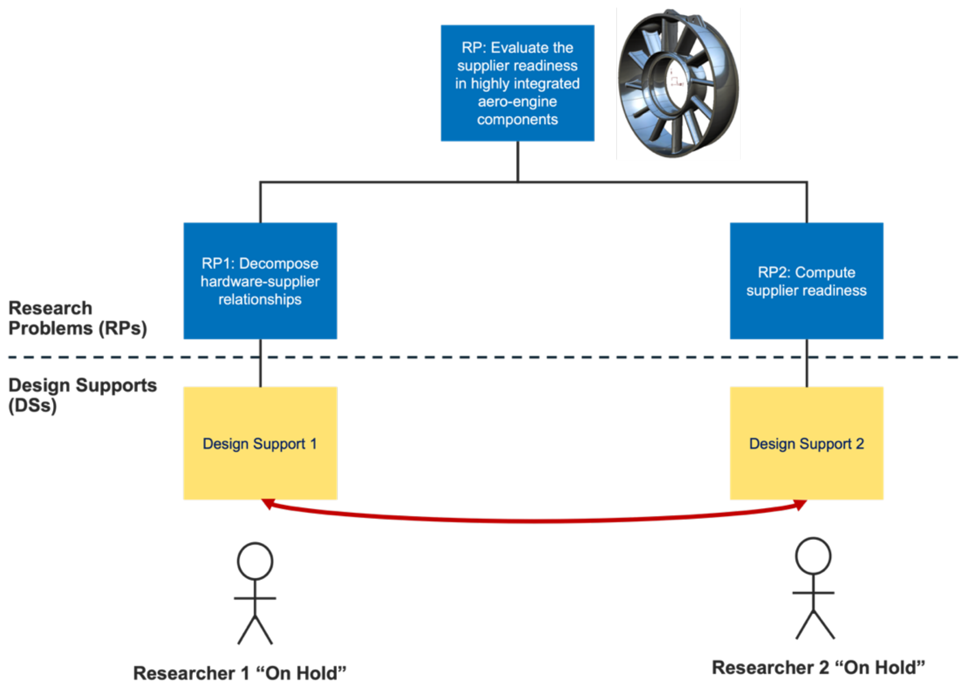

Figure 3. The impact of Design support Dependency on the ability to conduct an Agile Research Sprint

By applying Axiomatic Design principles and trading off between timeliness and comprehensiveness (Equation 4), a more independent problem-support architecture can be defined (Figure 4).

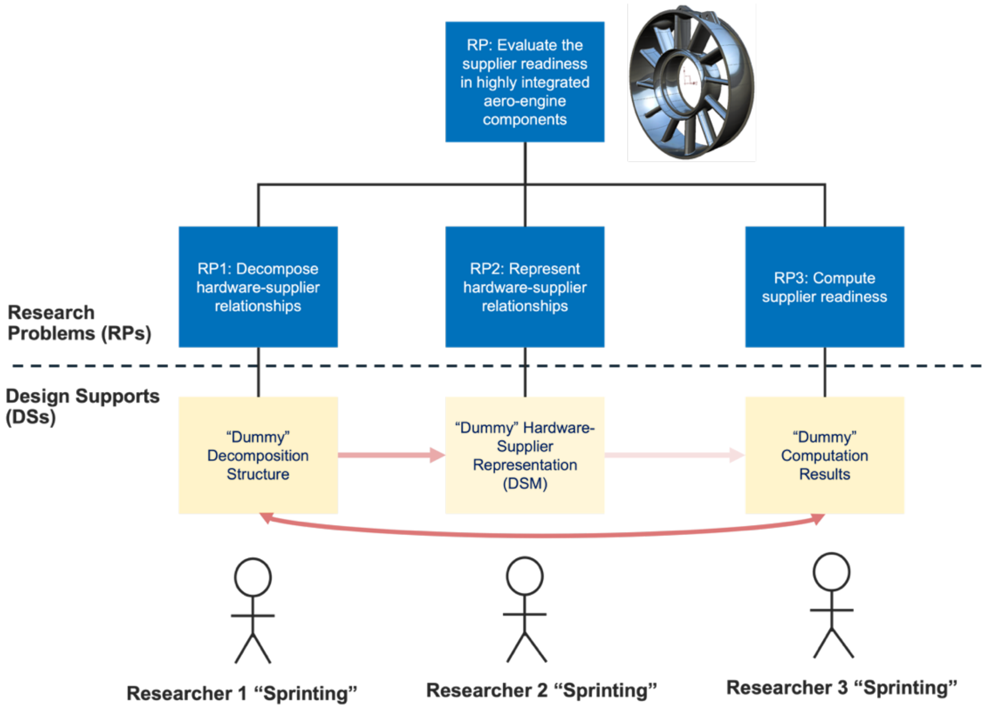

Figure 4. New Agile Design Research Architectures after Applying the Independence Axiom

As suggested by Suh (1995) the Research Problems are further decomposed. It is decided to create a new problem "*RP2: Represent hardware-supplier relationships*" in between the problem of decomposing and the problem of computing. Also, a new researcher is tasked to work on this problem.

Now, the three researchers are asked for early delivery of their design support, focusing mainly on the part of the support that is going to be the interface with the other supports. This, for example, can take the form of "dummy" datasets that represent an "idealized" version of the main output data coming from the "to be" supports. This allows to increase the timeliness of the support. Regarding comprehensiveness, the addition of *RP2* makes its "dummy" design support (e.g. a DSM) quite comprehensive already. The problem of representing relationships (of any form) has been a research focus for a long time (e.g., research in DSM dates back to the 50s). Therefore, the "dummy" hardware-supplier relationship can already be rather comprehensive (DSM has been demonstrated to be flexible to several applications). This allows to reduce the dependency among research problems (visualized in Figure 4 with arcs with different levels of transparency) and the researchers can therefore engage in their individual “sprints” keeping the momentum. However, some of these relationships can still be very dependent (for example among RP1 and RP3). This would require a new loop of axiomatic reformulation to be conducted.

4 A framework to use DSM as a Planning Support to Refine the “Design Problems Backlog” in Agile Design Research

The previous examples have highlighted how the independence axiom can be used to reformulate research problems, creating a “design problems backlog”. Figure 5 offers a more general framework for how the design problems backlog can be refined.

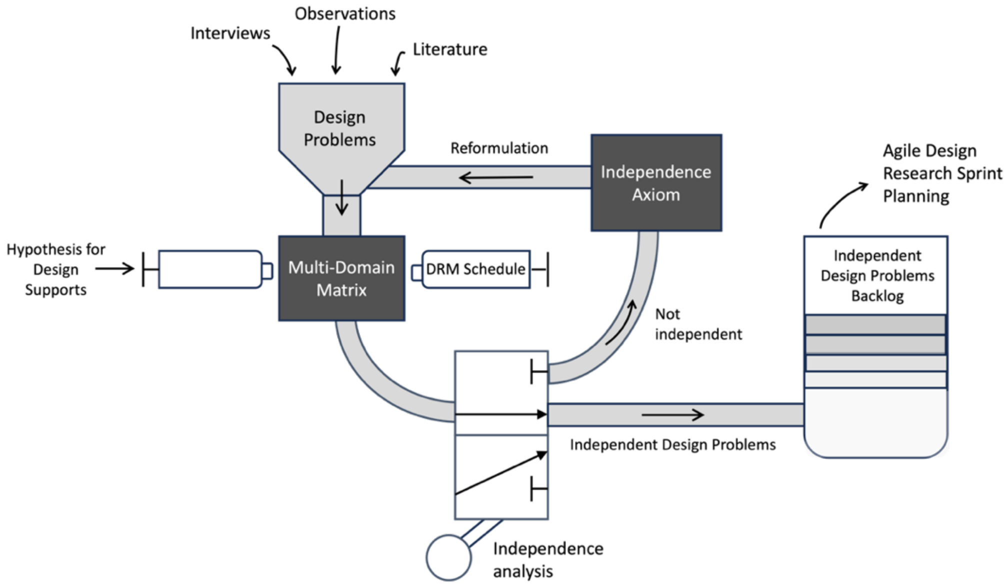

Figure 5. Framework for Design Problems Backlog refinement

The framework starts by collecting design problems from (among others) interviews, observations and literature. Afterwards, a Multi-Domain Matrix (MDM; Browning, 2015) is created by combining the design problems with initial hypotheses for design supports (e.g., the “dummy” supports in Figure 4) and the DRM schedule. An example of such MDM is shown in Figure 6, based on the example provided in Section 2.3.

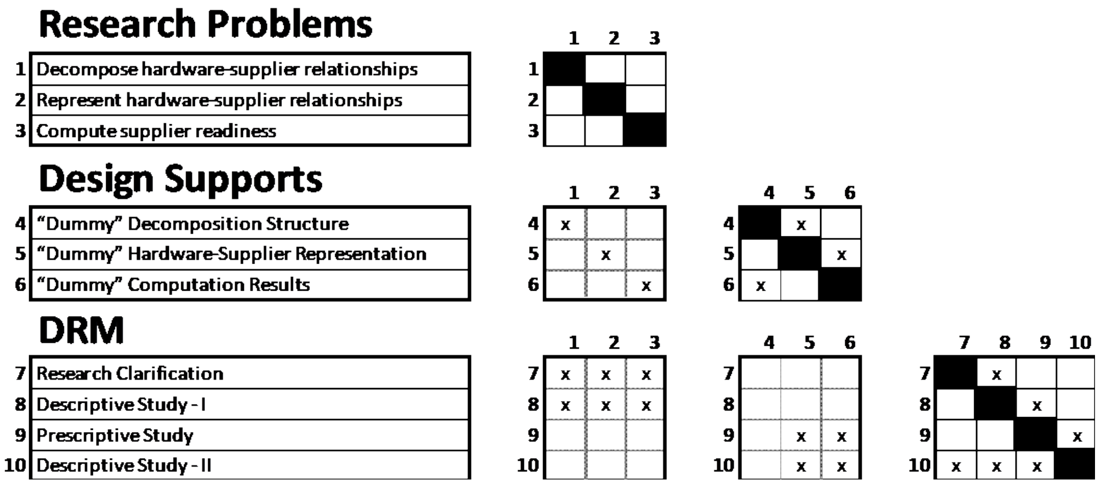

Figure 6. Multi-Domain Matrix among Research Problems, Design Supports and DRM

The MDM combines the Axiomatic Design matrix among Research Problems and Design Supports (rows and columns 1 to 6), as well as their relationships with DRM (rows 7 to 10 and columns 1 to 6). For simplicity, it is assumed that the Research Clarification and Descriptive Study – I stages are connected to the Research Problems, while the Prescriptive Study and the Descriptive Study-II are related to the Design Supports. The MDM allows also to identify relationships among the DRM stages (for example, it is assumed a series connection among the DRM stages, rows and columns 7 to 10) as well as possible risks of loopbacks among the stages. For example, here, it is assumed that if the Descriptive Study-II is not valid (row 10), all the other stages will be impacted. The impact of these loopbacks on the overall research efficiency is highly affected by the axiomatic independency among the research problems and the supports. To assess such independence, an *independence analysis* (symbolized by a two-way valve in the framework) is conducted. Various methods can be used to conduct such independence analysis, for example, the change propagation method (CPM, Clarkson et al., 2004). Here, the metrics introduced in Equation 4 (based on *Design Support Comprehensiveness* and *Design Support Timeliness*) are combined with assumed Likelihood – Impact (L-I) risk values for the different DRM stages. Such analysis for two research architectures reported in Section 2.3 is presented in Figure 7.

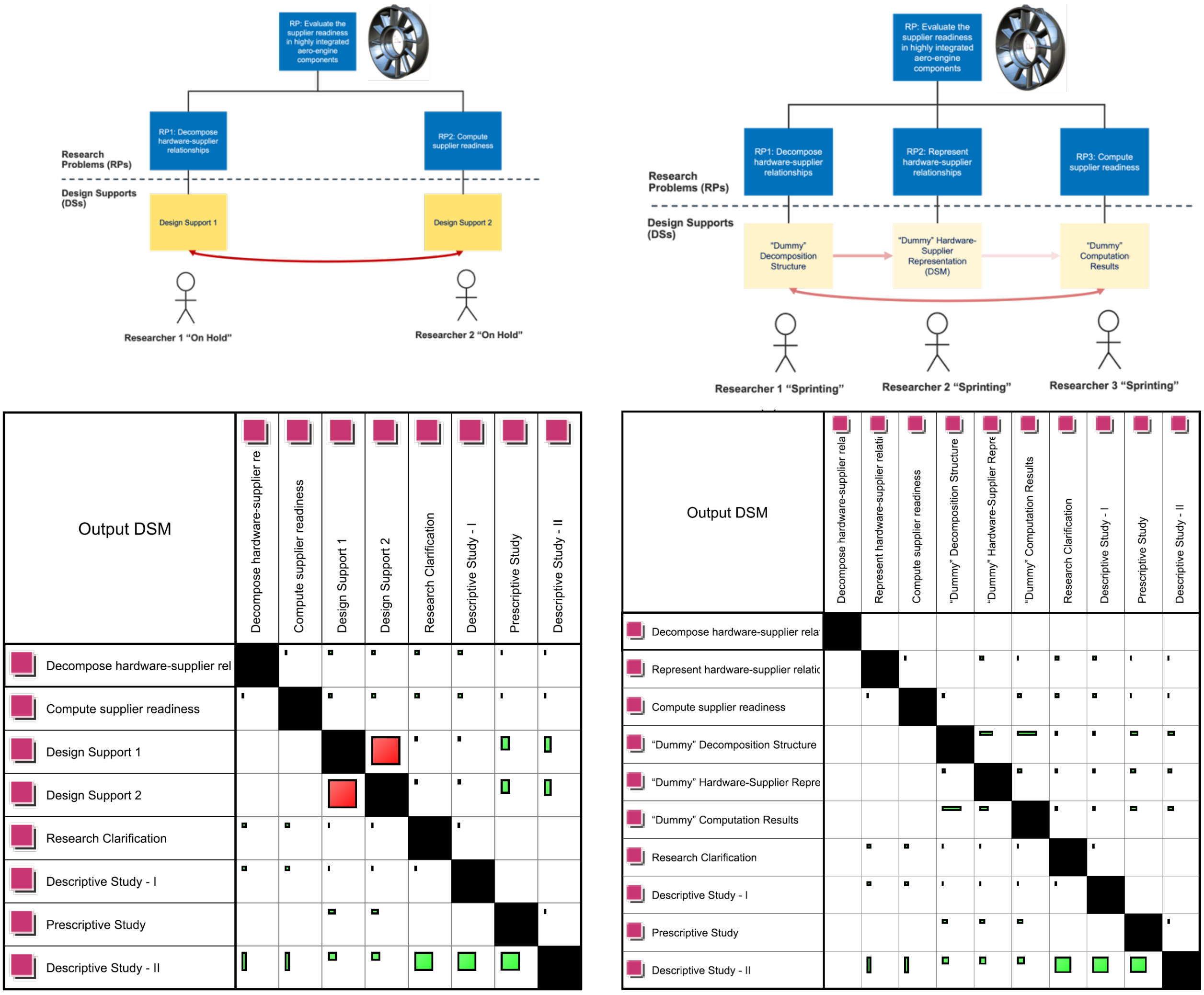

Figure 7. The independence Analysis (through CPM) conducted for the first and the second problem-support architecture

In the figure, CPM has run for both architectures and the mini-graphs show the combined risk of change (where a bigger square means more risk).

The figure highlights how in the first problem-support architecture there is a high dependency among the two research problems and the two design supports (the Axiomatic Design Matrix). This dependency impedes the ability of the two researchers to parallelize work and to conduct a whole DRM "loop" individually and separately. For example, if the Descriptive Study-II for Design Support 2 reveals to be invalid it means that the support needs to be reworked (a whole DRM loop needs to be run again for Design Support 2. However, this profoundly affects Design Support 1 as well. If Researcher 1 has started with an invalid input regarding Design Support 2, it means that also Design Support 1 needs to be reworked. This risk is less pronounced for the second problem-support architecture. There is still a risk that if the Descriptive Study II results are found to be invalid for one specific support, it must be reworked (last row of the matrix). However, this risk is confined to this specific Design Support itself, and minimally impacts the others (this is visualized in the rows belonging to the "dummy" design supports).

Once the independence is considered sufficient, the design problems "backlog" is created and stored (for example in a kanban-style list-making application). The researcher can now run an Agile Design Research sprint for only one problem at a time. This can be performed "safely" as any uncertainty encountered in the research process is going to affect only that specific problem (and not the others). One such uncertainty can be an unexpectedly long Prescriptive Study, due to the fact that developing the design support for one problem requires trial and error, because a new technique needs to be learned (e.g. programming). If independence is ensured, the work on another research problem is not affected, and the Agile Research Sprint for another problem can be planned and performed in the meanwhile.

5 Conclusion

The examples shown in this paper highlight the benefit of using DSMs combined with the 'independence axiom' suggested by Suh (1995) to control a limited amount of interfaces between design problems. Ensuring independence among research problems and design supports can enable an Agile Design Research approach, where a whole DRM "loop" is applied to each of the individual problems one at a time. In this context, the independence analysis is a fundamental step to ensure that the research problems backlog is independent. For example, one useful exercise - to perform regularly among supervisors and PhD students - could be to assess whether the design problem is axiomatically independent of others (using the MDM and CPM analysis). If the problems are dependent the "valve" activates (Figure 6) and PhD students and supervisors can engage in reformulating the problems and supports until independence is achieved.

Agile DR supports the idea that usefulness in design research can be achieved by a continuous and iterative process of gradually building confidence (Seepersad et al., 2006). Pedersen et al., (2000), for instance, argue that this gradual way of confidence building, derived from the relativistic school of epistemology, is necessary because of the often subjective nature of engineering design (making the logical empiricist approach unsuitable). The independent loops suggested in agile research are therefore nothing but a form of heuristic where idealizations or imprecise representations are made, to eventually 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). In practice, for instance, it is common for a design engineer to make suitable but imprecise assumptions when there is a lack of information. The assumption lets them move forward with the design, which may be optimized later when accurate information is available.

However, there may be operational challenges connected to the application of the independence axiom (with or without DSM) to research problems. There may be cases in which the research problems are indeed coupled. The same challenges have been encountered when applying agile in complex projects where complete independence is difficult to achieve (Drutchas and Eppinger, 2023). For these reasons, the Scaled Agile Framework (SAFe; Conboy and Carroll, 2019) has been proposed to handle interdependencies when implementing agile (so if problems are not independent, interfaces can still be managed). Recent research has started to investigate possible means to apply DSMs to assist in the application of Scaled Agile in such projects (e.g.; Narayanan, et al., 2021). One area of future research is to look into the ways in which Scaled Agile principles can be adapted for an Agile Design Research methodology for cases in which full independence is not achievable.

Acknowledgements

The project leading to this paper has received funding from the Clean Sky 2 Joint Undertaking under the European Union's Horizon 2020 research and innovation programme. Grant agreement No 887174. Also, 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

Antonsson, E. K., 1987. Development and testing of hypotheses in engineering design research. Journal of Mechanisms, Transmissions, and Automation in Design, 109(2), 153-154. Doi: 10.1115/1.3267429

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., et al., 2001. "Manifesto for Agile software development," [Online]. Available: https://agilemanifesto.org/

Blessing, L. T. M., Chakrabarti, A., & Wallace, K. M., 1992. Some issues in engineering design research. In OU/SERC Design Methods Workshop, The Open University, Milton Keynes.

Blessing, L.T. and Chakrabarti, A., 2009. DRM: A design research methodology (pp. 13-42). Springer London. Doi: 10.1007/978-1-84882-587-1_2

Browning, T. R., 2015. Design structure matrix extensions and innovations: a survey and new opportunities. IEEE Transactions on engineering management, 63(1), 27-52. Doi: 10.1109/TEM.2015.2491283

Cantamessa, M., 2003. An empirical perspective upon design research. Journal of Engineering Design, 14(1), pp.1-15. Doi: 10.1080/0954482031000078126

Cash, P., Isaksson, O., Maier, A. and Summers, J., 2022. Sampling in design research: Eight key considerations. Design studies, 78, p.101077. doi: 10.1016/j.destud.2021.101077

Clarkson, P. J., Simons, C., & Eckert, C., 2004. Predicting change propagation in complex design. J. Mech. Des., 126(5), 788-797. Doi: 10.1115/1.1765117

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

Conboy, K., & Carroll, N. (2019). Implementing large-scale agile frameworks: challenges and recommendations. IEEE software, 36(2), 44-50.

Cross, N., 2018. Developing design as a discipline. Journal of Engineering Design, 29(12), pp.691-708. Doi: 10.1080/09544828.2018.1537481

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. Doi: 10.1016/j.jss.2012.02.033

Drutchas, J. F., & Eppinger, S. (2023). ADJUSTING SCALED AGILE FOR SYSTEMS ENGINEERING. Proceedings of the Design Society, 3, 475-484.

Duffy, Alex H B and O'Donnell, F J; Mortensen, Niels Henrik and Sigurjonsson, Johannes, eds., 1999 *A design research approach.* In: Critical Enthusiasm. Norges teknisk-naturvitenskapelige universitet, Lisbon, Portugal, pp. 33-40.

Eckert, C.M., Stacey, M.K. and Clarkson, P.J., 2003. The spiral of applied research: A methodological view on integrated design research. In: Proceedings of the 14th International Conference on Engineering Design (ICED'03), 19-21 Aug 2003, Stockholm, Sweden.

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. doi: 10.1017/dsj.2020.21

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. doi: 10.1115/1.4034088

Horváth, I., 2007. Comparison of three methodological approaches of design research. In DS 42: Proceedings of ICED 2007, the 16th International Conference on Engineering Design, Paris, France, 28.-31.07. 2007 (pp. 361-362).

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. Doi: 10.1007/978-3-319-19449-3_24

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.

Lenart, M., & Pasztor, A., 2002. Constructing Design Worlds: Changing paradigms. In Artificial Intelligence in Design'02 (pp. 65-88). Springer Netherlands. Doi: 10.1007/978-94-017-0795-4_4

Mabrok, M. A., Efatmaneshnik, M., & Ryan, M. J., 2015. Integrating nonfunctional requirements into axiomatic design methodology. IEEE Systems Journal, 11(4), 2204-2214. Doi: 10.1109/JSYST.2015.2462073

Maier, J. F., Wynn, D. C., Biedermann, W., Lindemann, U., & Clarkson, P. J. (2014). Simulating progressive iteration, rework and change propagation to prioritise design tasks. Research in Engineering Design, 25, 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. Doi: 10.1016/j.respol.2015.11.009

Mumford, E., 2001. Advice for an action researcher. Information Technology & People, 14(1), pp.12-27. Emerald. Doi: 10.1108/09593840110384753

Narayanan, N., Joglekar, N., & Eppinger, S. (2021). Improving Scaled Agile with Multi-Domain Matrix. The Design Society.

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. doi: 10.1115/1.4033654

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.

Panarotto, M., Kipouros, T., Brahma, A., Isaksson, O., Strandh Tholin, O., & Clarkson, J. (2022). Using DSMs in functionally driven explorative design experiments–an automation approach. In DS 121: Proceedings of the 24th International DSM Conference (DSM 2022), Eindhoven, The Netherlands, October, 11-13, 2022 (pp. 68-77).

Pedersen, K., Emblemsvåg, J., Bailey, R., Allen, J. K., & Mistree, F., 2000. Validating design methods and research: the validation square. In IDETC and IEC (Vol. 35142, pp. 379-390). ASME. Doi: 10.1115/DETC2000/DTM-14579

Seepersad, C. C., Pedersen, K., Emblemsvåg, J., Bailey, R., Allen, J. K., & Mistree, F., 2006. The validation square: how does one verify and validate a design method. Decision making in engineering design, 303-314. Doi: 10.1115/1.802469.ch25

Sobek II, D. K., Ward, A. C., & Liker, J. K., 1999. Toyota's principles of set-based concurrent engineering. MIT Sloan Management Review.

Suh, N. P., 1995. Axiomatic Design of Mechanical Systems. ASME. ASME. J. Mech. Des. June 1995; 117(B): 2–10. Doi: 10.1115/1.2836467.

Vermaas, P.E., 2013. The coexistence of engineering meanings of function: four responses and their methodological implications. AI EDAM, 27(3), pp.191-202. Doi: 10.1017/S0890060413000206

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. doi: 10.1142/S0219877014400057

Contact: Massimo Panarotto, Chalmers University of Technology, Department of Industrial and Materials Science, Campus Johanneberg: Chalmersplatsen 4, SE-412 96, Tel: +46 031-7726785, E-mail: massimo.panarotto@chalmers.se