



Elements of a design method – a basis for describing and evaluating design methods

Kilian Gericke ¹, Claudia Eckert ² and Martin Stacey ³

¹Faculty of Mechanical Engineering and Marine Technology, University of Rostock, Rostock, Germany

²School of Engineering and Innovation, The Open University, Milton Keynes, UK

³School of Computer Science and Informatics, De Montfort University, Leicester, UK

Abstract

Method development is at the heart of design research as methods are a formalised way to express knowledge about how aspects of design could or should be done. However, assuring that methods are in fact used in industry has remained a challenge. Industry will only use methods that they can understand and that they feel will give them benefit reliably. To understand the challenges involved in adopting a method, the method needs to be seen in context: it does not exist in isolation but forms a part of an ecosystem of methods for tackling related design problems. A method depends on the knowledge and skills of the practitioners using it: while a description of a method is an artefact that is a formalisation of engineering knowledge, a method in use constitutes a socio-technical system depending on the interaction of human participants with each other as well as with the description of the method, representations of design information and, often, tools for carrying out the method's tasks. This paper argues that crucial factors in the adoption of methods include how well they are described and how convincingly they are evaluated. The description of a method should cover its *core idea*, the *representations* in which design information is described, the *procedure* to be followed, its *intended use*, and the *tools* it uses. The account of a method's intended use should cover its *purpose*, the situations or product types within its *scope*, its *coverage* of kinds of problems within its scope, its *expected benefit* and *conditions* for its use. The different elements need to be evaluated separately as well as the method as an integrated whole. While verification and validation are important for some elements of methods, it is rarely possible to prove the validity of a method. Rather the developers of methods need to gather sufficient evidence that a method will work within a clearly articulated scope. Most design methods do not have binary success criteria, and their usefulness in practice depends as much on simplicity and usability as on the outcomes they produce. Evaluation should focus on how well they work, and how they can be customised and improved.

Key words: design method, design theory

1. Introduction

The development of new means that support designers in their work (e.g., methods, tools, guidelines) is central to engineering design research (Blessing & Chakrabarti 2009). The development of design methods for use in industry has for decades been an important strand of academic research. At the same time, practitioners in companies develop methods themselves to solve recurrent problems or capture

Received 09 September 2021

Revised 07 November 2022

Accepted 09 November 2022

Corresponding author

K. Gericke

kilian.gericke@uni-rostock.de

© The Author(s), 2022. Published by Cambridge University Press. This is an Open Access article, distributed under the terms of the Creative Commons Attribution licence (<http://creativecommons.org/licenses/by/4.0>), which permits unrestricted re-use, distribution and reproduction, provided the original article is properly cited.

Des. Sci., vol. 8, e29

journals.cambridge.org/dsj

DOI: 10.1017/dsj.2022.23

the **Design Society**
a worldwide community

 **CAMBRIDGE**
UNIVERSITY PRESS

best practices. Methods, tools and guidelines seem like simple and straightforward concepts, but on closer inspection they cover a range from rigorous and repeatable procedures to loose sets of recommendations. The terms tool and method are often used interchangeably, and individuals are not consistent with their own uses. However, different terms imply different claims.

The aim of this paper was to aid design researchers or practitioners by clarifying the concept of a method, what needs to be said about a method, and how a method needs to be evaluated. This paper offers definitions for different elements of design support and argues that for a method to be usable each of the elements needs to be well worked out and well described. This paper sees methods as formulations of best practices or principles worked out in research. For a method to be adopted in the industry, it needs to convince industry experts. While much of the evaluation of methods is presented in terms of verification and validation, this paper discusses the evaluation of methods as gathering evidence that methods work in practice. It argues that not only do the elements of a method need to be evaluated separately, but they also need to be evaluated in a way that is appropriate to the nature of the claim they are making.

There are many very well-established methods or method families that engineers are using in many different companies, such as QFD, TRIZ or Six Sigma. These are well described in various textbooks, and engineers are trained through professional training courses when companies introduce them to organisations. Companies often take strategic decisions to adopt these methods, because they hear that other companies have successfully introduced them, as the example of lean manufacturing illustrates, where many companies have been following the well-published example of Toyota (see e.g., Liker & Morgan 2006).

However, new methods are often developed as part of PhD theses or time-limited research projects, and time has run out to test or describe them in detail so they are not developed sufficiently to be used in practice independently of the researchers (Gericke *et al.* 2021). The consequence of this is that the activity of developing a new method is more often an important step in training the next generation of engineering design academics and industry experts (National Academy of Engineering 2004; Eder 2007; Blessing & Chakrabarti 2009; Tomiyama *et al.* 2009; Cross 2018), rather than a direct contribution to industry. But new methods should make a useable and useful contribution to solving practical problems as well as provide a solid intellectual training. This paper is targeted at the developers of new methods.

1.1. Method transfer

The transfer of methods from academia to the industry has been recognised to be problematic for a long time as the industry only uses new design methods occasionally (Araujo *et al.* 1996; Birkhofer *et al.* 2002; Geis *et al.* 2008; Tomiyama *et al.* 2009; Jagtap *et al.* 2014) or in a crisis (Otto 2016). Wallace (2011) gives the following reasons: ‘methods tend to be too complex, abstract and theoretical’; ‘too much effort is needed to implement them’; ‘the immediate benefit is not perceived’; ‘methods do not fit the needs of designers and their working practices’; and ‘little or no training and support are provided’. Shortcomings in method development and transfer fall into three categories according to Jagtap *et al.* (2014):

- (i) *Method development*: insufficient method evaluation, insufficient communication of the value of methods, lack of an understanding of user needs and a discouraging reward system in academia.
- (ii) *Method (attributes)*: user-friendliness, cost and format.
- (iii) *Method use*: attitudes of users, improper use and awareness of design research.

Yet, many companies state that the use of design methods is an integral part of their activities (López-Mesa & Bylund 2010). Many practitioners use alternative names for methods or do not know the names of methods, and are often not even aware that they are using specific methods (Gericke, Kramer, & Roschuni 2016). Direct uptake of new methods published in academic literature is only one way that methods can achieve their intended impact. Students entering the industry and affecting working practices once they are in the right position is a different, powerful path for transferring knowledge about new methods, tools and processes (Gericke *et al.* 2020). This path to transformation is slow and hard to trace but effective. While students might not recognise the full potential of methods as novices, they learn vocabularies and ways of thinking through the methods that they are exposed to and look the methods up when they see a clear need for them. Thus, some of the underlying concepts of design methods affect design practice (Eckert & Clarkson 2005).

1.2. Convincing the user

Daalhuizen (2014) emphasises the method-developers' insufficient understanding of the needs and abilities of method-users (i.e., designers) and of the uses they would put methods to. Gericke *et al.* (2020) highlight that methods are embedded in an ecosystem of different methods used for different parts of the product development process and the success and usefulness of a method is affected by its fit with this ecosystem. Thus, method-developers have to consider that users need to understand, use and accept methods and that they will adapt methods depending on their use context, needs, and abilities, and the method-ecosystem. Methods need to convince prospective users that they provide means to overcome a crisis, improve established practices, or propose an entirely new approach for a perceived challenge. This requires validated claims that are clearly communicated, to manage expectations and avoid frustration resulting in a rejection of methods.

The usual approach in design research is to evaluate methods by applying them in practice. While this can corroborate the method and shows the effect of a successful application, it gives little indication of which aspects of the method are successful and which are not. If a method application disappoints the causes are often not clear. It is therefore important to break methods down into separate elements. This paper argues for separating the core idea, the representation, the procedures and the tools associated with a method, and break these down even further to allow a separate evaluation of the elements as well as an overall evaluation of the method. Only if this is clear can methods be developed further and built on each other.

1.3. The motivation of the paper

This paper is the result of discussions between the authors over the last 10 years as well as various workshops held by the Design Process SIG of the Design Society.

The authors chaired these workshops; they teach methods in engineering design and computer science and have developed themselves methods that have been applied in industry. A workshop at the Design conference at Cavtat, Croatia in 2016 revealed that while many of the Design Society members did teach methods, there was little consensus on what constituted a method and many of the methods listed had not been developed by the engineering design community. A subsequent SIG workshop with industry keynotes confirmed that the industry often uses parts of methods only as long as the methods help in addressing concrete problems. This workshop also included a preliminary discussion of the concepts related to methods, which later led to the definitions of the elements of methods presented in this paper (see Gericke, Eckert, & Stacey 2017). A later workshop focussed on the need for a coherent method ecosystem in which methods build on each other and use coherent terminology (Gericke *et al.* 2020).

1.4. Overview

This paper describes the background to the study of the use of design methods in the industry in Section 2 and argues that methods are an important form of engineering knowledge but that methods in use constitute socio-technical systems. Section 3 explores the concept of methods and sets out a number of constituent parts of methods – *the core idea, the representation, the procedure, the intended use* – which need to come together for a method to be successfully applicable in industry. The decomposition of methods into these elements is expected to facilitate the description, evaluation and adaptation of methods. Section 4 discusses the evaluation of methods, considering how far verification and validation are useful concepts, and when evaluating how well a method works is a better approach. While arguing that a complete rigorous evaluation is seldom possible due to the range of circumstances in which methods would be used, we advocate for an as complete evaluation as possible under the given constraints. Conclusions are drawn in Section 5 for how we should approach developing and introducing methods in industry.

2. Methods as a vital element of engineering design research

‘If you call it, “It’s a Good Idea To Do”, I like it very much; if you call it a “Method”, I like it but I’m beginning to get turned off; if you call it a “Methodology”, I just don’t want to talk about it’ (Alexander 1971).

To make sense of what a method contributes to designing an artefact – or could contribute to designing an artefact – we need a clear view of what it is. What the method is different for different kinds of methods, and often the effective use of a method can be limited by an insufficiently sophisticated view of what it offers. In this section, we outline some important aspects of the relationship between methods and design processes, that influence what we need to know about it.

Much of the confusion with respect to design methods comes from disagreements about the relationships between methods, tools, and other means intended to support designers. These are exacerbated by differences in connotation in different languages and disagreements about the scope of these terms, and changes in their meanings over time.

2.1. Historical perspective on the use of the term method and methodology

The use of the terms ‘design method’ and ‘design methodology’ has changed over the years. During the design method movement in the 1960s apparently little distinction was made between method and methodology, while the term ‘the design method’ referred to the overall process of producing a design, considered as a whole.

‘... the process of design: a process the pattern of which is the same whether it deals with the design of a new oil refinery, the construction of a cathedral, or the writing of Dante’s *Divine Comedy*. (...) This pattern of work, whether conscious or unconscious, is the design method. The design method is a way of solving certain classes of problem’ (Gregory 1966).

Over the last 40 years the word ‘methodology’ has acquired a distinct meaning in software design referring to the specification of an overarching approach to producing an artefact that includes different activities and how they are sequenced. More elaborate and prescriptive methodologies specify what outputs should be produced by the different activities and how these should be described; less commonly they can specify the methods that should be used to perform the activities. Engineering is also adopting this conceptualisation; however, so far, no consensus has been reached on how formal, detailed and prescriptive an approach needs to be to qualify as a methodology.

Multiple definitions for method and methodology exist, which reflect the perspectives of different design researchers, as illustrated in Heymann’s (2005) overview of the evolution of German design research. The leading design theorists at the different German engineering schools differed in their understanding of engineering design as art or as science, which is reflected in the methodologies and methods they proposed. Those, like Rodenacker or Hubka, who treated design research as a scientific discipline similar to other disciplines such as physics, were strict methodologists striving for precise and general scientific findings about how to do design, while pragmatic researchers like Redtenbacher in the 19th century and Leyer in the 20th argued that designing is not a scientific activity. In between, many design researchers, like Pahl and Beitz, took a flexible stance on methods and methodologies, with a pragmatic interpretation of methodological contributions.

These differing perspectives are reflected in the interpretation of how methods ought to be used: as strict recipes that must be applied without modification (strict perspective) or as recommendations that can be adapted if required (pragmatic perspective). Hubka (1982) defines a design methodology as ‘General theory of the procedures for the solving of design process [...]. Idealised conditions are usually assumed for the factors [...] influencing the design process and the model is intended to be valid for all types of design problem [...]’. Contrary to this understanding of design methodology being a theory, Pahl and Beitz propose a methodology based on successful strategies observed in industry as a means to train students and as guidance to designers. They define a design methodology as ‘a concrete plan of action for the design of technical systems (...). It includes plans of action that link working steps and design phases according to content and organisation’ (Pahl et al. 2007).

Views also differ on how prescriptive a way to do something needs to be to qualify as a method. Cross (2008) includes all observable ways of working, which

can be ‘procedures, techniques, aids, or “tools” for designing’ in the context of product development, as design methods. French (1999) even includes ‘ideas, approaches, techniques, or aids but he distinguishes them by maturity and breadth of applicability. Pahl et al. (2007) define a method as a ‘systematic procedure with the intention to reach a specific goal’. Daalhuizen (2014) narrows this definition to ‘Methods are means to help designers achieve desired change as efficiently and effectively as possible’. However, demonstrating that a method is as effective and efficient as possible is rarely possible.

The term ‘tool’ also causes confusion. For example, software programs are clearly tools, but are sometimes also referred to as methods. Birkhofer *et al.* (2002) clarify that tools are working aids, that is, means that support the application of a method.

2.2. Methods are part of ecosystems

Methods cannot really be seen in isolation. They are used in conjunction with other methods for tackling different parts of the design process, from which they receive information or feed information to. Successful development requires that the information content of the results produced by the different methods fit the requirements of the methods they provide inputs for. Gericke *et al.* (2020) argue that successful combinations of methods should be seen as *ecosystems*. Different methods for different parts of the process do not merely coexist but occupy interlocking niches in an environment. This environment is formed by the various design tasks that an organisation carries out as well as by the methods themselves.

For methods to work successfully they need to function in the ecosystem of methods used in the design process. This requires the methods used for different parts of the same design process to employ compatible and overlapping sets of concepts, as well as representations and terminology, so the outputs produced by one method really can be used as the inputs required by the next. This requires a certain alignment of perspectives, ways of looking at design problems and formulating solutions, that embody compatible sets of assumptions about how to carry out design (Gericke *et al.* 2020). Drawing on Andreasen (2003), Daalhuizen and Cash (2021) refer to the way of thinking about the problem and its potential solution that underlies a method as a *mindset*. Bucciarelli (1994) terms the sets of terms and concepts that members of particular professional communities think with an *object world*.

Many methodologies recommend a consistent set of methods and thus could be considered as method ecosystems (such as Pugh 1991; Roozenburg & Eekels 1995; French 1999; Andreasen & Hein 2000; Frey & Dym 2006; Pahl *et al.* 2007; Ulrich & Eppinger 2008; Ullman 2010; Vajna 2014); and others in mechanical engineering. These methodologies are consistent throughout the individual methods, because they typically come from a particular worldview or school of thought, or are often developed by a single research group. However, other methodologies provide ways of identifying and organising tasks without recommending specific ways or methods for executing these tasks, such as the V-model in engineering design, and Rapid Application Development (Martin 1991), in software development).

Combinations of notational formalisms for different types of information play an important role in some design disciplines, notably software development. They function as representation ecosystems; they inform design mindsets and shape

design activities and choices of methods. Most famously UML (see Fowler 2004) was consciously developed as a representation ecosystem for the Unified Process (Jacobson, Booch, & Rumbaugh 1999), but is much more widely used. SSADM (see Goodland & Slater 1995) is an older, now outmoded example of a software development methodology built on a set of notations that function as a representation ecosystem.

2.3. Methods as a form of engineering knowledge

Methods can also be seen by a formalisation of knowledge gained through engineering practice. The design methods in widespread use are grounded in a combination of extensive design practice and theoretical analysis. Different kinds of engineering knowledge, especially the explicitly codified how-to knowledge incorporated in methods, present distinct challenges for how we can justify the claims we want to make about truth or accuracy, so we can treat them as knowledge. Habermas (1984) argued that (explicitly stated) assertions need to be justified with *good* reasons; reasons are accepted as good reasons when a consensus within a discourse is reached on their acceptability. Different fields use different criteria. Different types of knowledge also require different criteria.

Vincenti (1990) divides engineering knowledge into fundamental design concepts, criteria and specifications, theoretical tools, quantitative data, practical considerations, and design instrumentalities. While Vincenti provides a fairly comprehensive map of types of engineering knowledge, he says little about explicitly described or formalised methods, and it is not obvious where they fit. Our view is that they should be seen as their own category, distinct from the knowledge of how to carry out design activities that Vincenti refers to as design instrumentalities, covering skills, problem-solving strategies, ways of thinking about design problems, and capacities for making judgements. When designers apply design methods, they often employ theoretical tools, and use criteria and specifications as well as quantitative data. The deployment of fundamental design concepts is often an important element of design methods.

The different elements of Vincenti's classification of knowledge have a different epistemic status. Fundamental design concepts are mostly tried and tested elements of the designs, such as machine elements, that designers can rely on working, as well as more abstract solution principles. Theoretical tools are based on established scientific knowledge, even though they are often only used by engineers and not by scientists. Quantitative data are usually objectively measured, even though what and how it is measured is often governed by complex socio-technical processes. Criteria and specifications and practical considerations arise from the specific problem context and are also subject to human interpretations. Design instrumentalities are rather more mixed; Vincenti uses the term as a catch-all for everything engineers know about how to design. While some of the how-to knowledge is well tested and explicitly communicated, other elements are tacit. Many designing activities governed by the application of explicit methods involve perception and action skills that people cannot fully explain and need to learn experientially (Ryle 1946, 1949). Elements of what engineers know about how to design are explicit prescriptions, such as process models or test procedures, which the engineers need to adapt to specific situations. Vincenti does not talk much explicitly about methods.

Vermaas (2016a) points out that many methods are formalised descriptions of expert behaviour augmented with tools and representations. The elicitation and the negotiation of the understanding of methods is a social process, that this profoundly influenced by the knowledge and perspective of both the expert and the researcher. As this paper will argue, methods are heterogenous. However, as methods often incorporate the experiences of one or more people they can be thought of as a form of testimonial knowledge, where a judgement about the origin of a statement affects its credibility and the credibility of the statement cannot be solely derived from the statement itself. The consequence of this is that unless methods are established through long and widespread use, they need accounts of their provenance, so that users have a basis for assessing the claims that are made for them, as well as accounts of what those claims *are*.

2.4. Methods as socio-technical constructs

Engineering design researchers are familiar with engineering products being elements of socio-technical systems comprising humans as well as mechanical and electronic systems and software, such as air transportation systems, water supply networks or healthcare systems (de Weck, Ross, & Magee 2011). 'They are ensembles of technical artifacts embedded in society, connected with natural ecosystems, functioning within regulatory frameworks and markets, and exhibiting a high degree of complexity and dynamics that are not fully understood' (Siddiqi & Collins 2017). In the sense of this characterisation, methods are also socio-technical in nature.

A description of a method is an artefact in its own right. It is an embodiment of engineering knowledge, available for engineers to draw on and employ more or less exactly. But a method in use is a socio-technical system: the engineers applying the method are as much part of the system as the descriptions of the method, or the diagrams, models and documents they read and construct. What happens depends on the interaction of all the elements of the system: engineers interpreting the description of the method, using software systems, reading diagrams, passing on documents to their colleagues. Communication in design processes includes communication *through* documents such as requirements specifications and design drawings, but also conversations between colleagues exchanging information informally that may never be recorded in any formal documentation (Bucciarelli 1994; Henderson 1998). Design meetings often involve a fluid mixture of speaking, sketching and gesturing where each channel would be unintelligible in isolation (Bly 1988; Tang 1989, 1991; Minnemann 1991; Neilson & Lee 1994).

Viewing the application of the method as an enactment of the socio-technical system, and the engineers as participants in the enactment, focuses our attention on the crucial role of the engineering knowledge the participants possess and how they combine it with the knowledge embodied in the method. While experts in particular specialisms have knowledge that overlaps, giving them shared competences, everyone's knowledge and experience is idiosyncratic and much of it is tacit. Not only are all design problems unique, so are the components of the socio-technical systems enacting the methods. It is important to maintain the dual perspective of a method as a means to carry out a task, and a method as a socio-technical system generated by the actions of its participants.

Many methods incorporate and make explicit a lot of technical information, but they need to work in the social context in which they are used. How well a method works depends not only on the method itself but on the context in which it is deployed. What the engineering problem is and how well it fits the assumptions behind the method influences whether the method is appropriate. The knowledge of the individual engineers plays an important role in exactly how the method is used. And the social organisation of the design team and design process influences how the method is enacted as a socio-technical system. Many methods are explained and tested through small-scale problems, and applying them to complex real applications can pose problems. If the developers have to learn new skills or familiarise themselves with new models or data structures, it adds to the implementation effort.

It is hardly possible to predict or prescribe fully how a method is used. The users operate in changeable environments and use and adapt methods in ways that seem useful to them at the time. The interaction design maxim that you cannot design a user experience, only design *for* a user experience, applies to design methods.

Dorst (2008) has criticised engineering research for concentrating on the activities required to carry out a task and therefore focusing on efficiency and effectiveness, while neglecting the object, the actor and the context in which these activities are carried out. We see descriptions of methods, as artefacts, as powerful ways to share knowledge across and between organisations that researchers and educators can facilitate. We see methods in use as socio-technical constructs, so that the actors and the context become an integral part of design methods where the actor is considered in the procedure as well as the context and the product and its intended uses. This highlights the need to be clear about the assumptions the method embodies about the human resources it requires, what its users can do and how they think, as well as about the type of problem it will be applied to.

3. What do we need to say about a design method?

The terms method, tool, guideline or thinking tool are sometimes used interchangeably or with overlapping meanings. Greater clarity would benefit various audiences in:

- (i) Selecting suitable methods and using methods appropriately.
- (ii) Aiding clear communication among design researchers.
- (iii) Clarifying the claims made for the method thus enabling a better understanding of failed or successful method application.
- (iv) Enabling the assessment of the appropriateness of the evaluation approach.

Different abstractions of methods exist in the literature (e.g., Jones 1970; Zanker 1999; Lindemann 2009), often intended to facilitate organising methods for different purposes such as method selection or comparison (Birkhofer 2008). Other authors, such as Zier, Bohn, & Birkhöfer (2012), organise the many different methods with the aim of making teaching them easier, by decomposing methods into elementary methods, that is, fundamental information entities and basic operations. However, these efforts often focus on single views or elements of methods, such as the procedural aspect of methods or their purposes, neglecting other important elements of methods. They also ignore the embedding of methods

in the socio-technical context as well as dependencies between the inputs and outputs of methods in method eco-systems.

The following section builds on prior work (Gericke *et al.* 2017) and introduces a definition of the elements of methods as well as an account of the description of a method required to support the introduction of methods into industry. Daalhuizen and Cash (2021) propose a similar abstraction of methods and their relation to use, context and design outcomes with the intention of predicting a method's performance.

3.1. Core vocabulary

Blessing and Chakrabarti (2009) apply the term 'design support' to the entirety of means that are intended to support designers in their activities. They draw the following conceptual distinctions between different categories of design support:

- (i) Design approach/methodology
- (ii) Design methods (different classes of methods distinguished depending on their primary purpose, for example, methods for analysing objectives and establishing requirements, methods for evaluating and selecting support proposals)
- (iii) Design guidelines (including rules, principles and heuristics)
- (iv) Design tools (including hardware and software)

Building on Blessing and Chakrabarti (2009), Gericke *et al.* (2017) clarify some of the terminology drawing on relevant literature from engineering design and computer science (see Table 1) and contemporary usage in teaching and industry.

A design methodology connects methods, guidelines and tools, each of which can exist individually, through an organised process of design activities, and the use of the methods and tools. Tools aid or enable the application of methods and guidelines, and the organisation and performance of the process.

3.2. A proposal for describing design methods

No method is equally applicable to all design situations. It is therefore very important to be clear how and when a method is intended to be used. In practice it might be difficult to state exactly when a method does or does not work; however, from a usability perspective, it is better to underclaim rather than overclaim the scope of a method. As this section will argue, methods have different elements, and these also have potentially different scopes. This decomposition of what a method constitutes is expected to facilitate a clear description of a method and the evaluation of methods.

Methods are usually developed with a particular use or application area in mind. For example, a method might be developed for early stages of the development process in the automotive industry. This is the intended use from the viewpoint of the developer of the method, which the developer can evaluate and comment on with confidence. However, the method might in fact be applicable more widely. For example, the same method might be useful in later phases of the process, or in the aerospace industry. Therefore, the actual scope of a method might well be much larger than the scope that is validated as part of the intended use. Hence, we propose to consider including a description of the intended use of a

Table 1. Examples of different types of design support from Gericke *et al.* (2017)

Term	Explanation
Design methodology	In design, a clearly and explicitly articulated approach to producing designs for a class of systems, that specifies in more or less detail the activities to be carried out, the relationship and sequencing of the activities, the methods to be used for particular activities, the information artefacts to be produced by the activities and used as inputs to other activities, as well as (tacitly or explicitly) the paradigm for thinking about the design problem and the priorities given to particular decisions or aspects of the design or ways of thinking about the design
Design process	In design, the phrase <i>design process</i> has two different and equally well-established meanings, of which the first constitutes a type of design support. (a) A formally specified sequence of activities to be carried out in developing a particular design, or a class of designs, which will often be an application or customisation of a methodology to a particular problem (b) The actual sequence of activities carried out in the development of a design, which may correspond more or less well to any formally specified process
Design method	A specification of how a specified result is to be achieved. This may include specifications of how information is to be shown, what information is to be used as input to the method, what tools are to be used, what actions are to be performed and how, and how a task should be decomposed and how actions should be sequenced
Design guideline	In design, a statement of what to do when, or what should be the case under particular circumstances. A guideline should only be violated for a good reason, with a careful consideration of the consequences
Design standards	In mature areas of design, standards are a binding set of prescriptive steps that need to be followed and – unlike guidelines – can be audited
tool	An object, artefact, software or other means that is used to perform some action (for example to produce new design information). Tools might be based on particular methods, guidelines, processes or approaches, or can be generic environments that can be used in conjunction with many methods

method when describing a method. In the following, guidance for describing both is provided. The level of detail in the description should allow practitioners to easily find relevant information. This means the focus should not be on obvious information but on the information that is hidden or necessary and on underlying assumptions relevant to the method application. An abstract might be a cost-effective and useful means to assist method users when exploring new methods.

The intended use of a method

A method is a kind of promise: follow the outlined steps and you should be able to get a certain kind of result. Behind these prescriptions are theoretical claims. However, these are rarely made explicit in the description of methods. For users to form their own opinions on the suitability of a method they would they require the following information:

Purpose of a method: What is the method intended to achieve? A design method is supposed to achieve something: it is supposed to generate new information about

the design. Beyond that, it is hard to generalise; many design methods are for advancing the design by making new design decisions and constructing new parts of the design (synthesis methods), while others are for generating additional information *about* the problem or the current state of the design (analysis methods); some methods are intended to produce formal, exact, trustworthy results in a particular format, while others are intended to produce qualitative or approximate results or just insights into the problem. In order to judge whether a method can meet a need you have, you need to know what outcomes it is intended to produce. A description of the ostensible information content of the outputs may not be enough; making the purpose clear may require an account of how exact or approximate the results are, and how complete they are, as well as (if relevant) how they can be validated.

Scope of a method: What situation or product type is a method intended for? Methods typically arise from particular case studies or sectors, but this is not always stated. The scope that the researcher can be confident about might be limited, leaving the burden of transferring it elsewhere with the user. Many methods overclaim; for example, many method descriptions make claims in terms of “design” or “engineering” in general, while others are targeted at particular sectors like automotive or aerospace. Similarly, it is often not made clear whether methods are intended for use in original design, incremental design, or all design situations. Underlying this might be tacit assumptions about the similarity between different industry sectors, or what the activities are in the design processes, which domain experts might not share.

The coverage within the scope: Is the method applicable to all problems in the scope or only to some? There might be particular situations where methods are not applicable or required information might not be available. For example, a method targeted at aerospace might only be useful for incremental problems in the early stages.

Benefit expected from a method: What benefit can be expected from using the method? The utility of a method can also vary from providing some helpful insights to completely solving a problem. This varies with both the situation in which the method is used and the skills and attention of the user and the level of completeness with which it is applied.

Conditions for using a method: What knowledge and information is needed for using the method? Many methods assume that their users have particular skills and background knowledge, as well as particular items of input information. While it may well not be cost-effective to describe this in detail, making some key assumptions explicit may be essential to avoid failures in use when tacit success conditions are not met.

A description of the intended use should inform the user about the suitability of the method for a particular design task in a specific context. Where necessary, it should include a description of the conditions for successful use of the method, including what knowledge and skills the users need.

The elements of a design method

The description of the intended use for a method is complementary to the description of the method itself, which comprises the *core idea* of the method, the *representation* in which design information is described, and the *procedure* (see

Table 2. Explanation of design method terms

Term	Explanation
Core idea	The basic principle, technique or theory that the method employs for generating new information that constitutes the output of the method
Representation	Many methods specify or require a notation or way to create a representation of the new information the method produces, that fits the concepts employed by the method and is integral to how the method works. In some cases, the representations may be informal and flexible, in others they are precisely defined formalisms
Procedure	The set of actions or activities required to apply a method, for enabling the user of the method to do something more easily or with a sufficient guarantee of correctness
Tool	An object or artefact that is used to perform some action (in our case, to produce new design information). For design methods, documents or software that direct and structure designers' thoughts and actions, or facilitate constructing representations, or perform computations
Intended use	The purpose and scope of a method, the coverage within the scope, the expected benefit from using the method, and the conditions for using a method

Table 2). Core idea, representation and procedure build on each other (see Figure 1) and form the method, thus the method description should provide the necessary information about each element of the method as well as information about any tool implementation of the method if available or required. A method might have dedicated tools, shared tools with other methods, or use generic tools (see Gericke *et al.* 2017).

A description of the procedure, focusing on the sequence of actions and their completeness, is an essential part of a description of a method. The method description should provide, besides explanations of each element of the method, information about possible adaptations of representations and procedures that allow the method's use in different contexts, as well as information about the required rigour in the application of the method. Some elements of a method might allow adaptation while other elements, for example, those required for or related to compliance, should not be modified. For example, the same method could use alternative representations, such as using graphs instead of matrices. Method users should be informed about such options and limits of adaptation. How much the description of the method needs to say about the expected mindset with which users of the method think about the problem and the form taken by potential

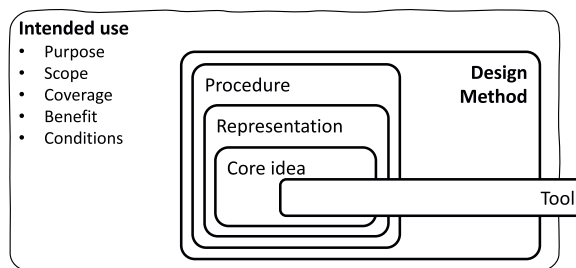


Figure 1. Elements of a method, based on Gericke *et al.* (2017).

solutions will depend on context, particularly whether the method is part of a coherent methodology or fits into an established method ecosystem.

4. Validation of design methods

Engineers are very familiar with the need to verify and validate their products or services before they can be released. Verification is establishing that something conforms to its specification; validation is establishing that something is fit for purpose. The modelling and simulation community is concerned with validity of models and sees model validation as ‘the process of determining the degree to which a model is an accurate representation of the real world from the perspective of the intended uses of the model’ (AIAA 1998); see also (Sargent 2013).

In engineering, verification and validation are hugely important in the context of quality control for physical systems, and rigorous processes are required to carry out systematic testing of physical systems (Tahera *et al.* 2019). The engineering research community brings this mindset also to the tools and methods it develops. In software development, rigorous testing of systems is also essential. However, in the development of interactive systems, a separate process of evaluating the interaction between the system and its users is also needed.

This section starts by looking at the literature on the validation of methods in the engineering design literature, but argues that in practice it can be more helpful to think in terms of evaluation, that is, gathering evidence that methods are or are not useful, and critically assessing how well a method works and under which circumstances.

4.1. Validation of methods in the design research literature

According to Blessing and Chakrabarti (2009) ‘design research has two related objectives: the formulation and validation of models and theories about the phenomenon of design ... and the development and validation of support founded on these models and theories, in order to improve design practice, including education, and its outcomes’.

An analysis of papers published in the proceedings of the International Conference on Engineering Design (ICED) up to 2003 by Cantamessa (2003) revealed that 50% of design research was concerned with the development of new tools to support design processes or activities. The rest covered empirical, experimental and implementation studies as well as research on design theory and education. Many research papers do not even attempt to validate their findings (Blessing & Chakrabarti 2009). A study of the 78 papers published in *Research in Engineering Design* from 2006 to 2010 revealed that nearly half the papers made no attempt to validate the research, and only four included an experimental validation while some were using case studies (Barth, Caillaud, & Rose 2011). A more recent analysis of studies aiming at design method validation reports a lack of success evaluation. A possible explanation given by the authors is that aside from the high effort required by such studies strategies for the transition of methods to practice are missing (Eisenmann *et al.* 2021).

One fundamental division is between a logical verification, which checks the consistency and completeness of the steps, and a form of verification by acceptance,

which involves an application in practice by experienced designers and their acceptance of the methods and their underlying theories (Guba & Lincoln 1989; Buur 1990). Validation by acceptance is usually based on case studies. These require a well-thought-out strategy for data collection, analysis and interpretation, so that the metrics for performance are trustworthy indicators of success – that is, they have construct validity (Yin 2014).

Pedersen *et al.* (2000) developed these ideas further and proposed the validation square consisting of four quadrants, which combine the internal consistency of the research and its application to a target context along two dimensions, giving: *Theoretical structural validity*; *Empirical structural validity*; *Theoretical performance validity*; and *Empirical performance validity*. The Design Research Methodology (DRM) by Blessing and Chakrabarti (2009) proposes validation steps along the stages of the research and advocates validating the developed design support during a second descriptive study.

However, potential improvement can be due to a large number of different factors, one of which is the Hawthorne effect, that is, improvement due to the fact that the process is observed, and its participants feel valued. In an industrial setting, identical situations rarely occur, because either the product has changed or learning from the past design effort has occurred.

Hazelrigg (2003) points out, in the context of decision-making tools in engineering design, that while it is possible to show that a method is not valid through results, validation can be done only mathematically, and only through validation of the procedures. Validation therefore requires a logical framework laying out the steps in a procedure, which themselves must be valid. They must be rational, self-consistent and derivable from a self-consistent set of axioms, otherwise, the method might have inherent contradictions. Hazelrigg's argument only makes sense for problems to which mathematical decision theory is directly applicable, but it suggests that rigorous validation of methods is otherwise impossible.

While Hazelrigg argues for a mathematical validation, Frey and Dym (2006) advocate a scientific paradigm based on an analogy to medicine. Controlled field experiments can be seen as the equivalent of clinical trials, studies of industrial practice as material experiments, and lab experiments as the equivalent of in vitro experiments, while detailed simulations can be seen as analogous to animal models. Many other authors also advocate testing methods experimentally. Kroll and Weisbrod (2020) suggest that methods should be evaluated in terms of applicability (ease of use, ease of teaching, ease of understanding, ease of following procedures) and effectiveness, where the criteria depend on the purpose of the methods, for example in their case supporting creativity. However, experimental evaluation requires careful design of the experiments, in terms of understanding the scientific and practice concerns, doing the appropriate theoretical framing, defining the scope, intended generalisability, sampling scheme and sample size and strategy (Cash *et al.* 2022). However, it is unlikely that in the development of methods it is possible to obtain large enough sample sizes of experienced practitioners to get statistically significant results. Many researchers therefore resort to student groups, which can provide useful insights into aspects of method development, such as the quality of the method description, but students ultimately lack the situated knowledge that practitioners would bring. This can be effective for methods that build on general human capabilities, but can be problematic for highly context-dependent activities, as it is often the case with engineering methods. Some

universities have therefore created LifeLabs, where students can operate under conditions closely resembling industrial practice, to test methods (Albers *et al.* 2018). Statistical measurements of outcomes may still be unobtainable or uninformative. Experiments can nonetheless play a valuable role as artificial case studies for analyses of what happened in the application of the method and why.

There is largely a consensus that the evaluation of design research, and by implication method development, needs to be split into different elements. Eckert, Clarkson and Stacey (2003) divided research validation by the phases that design research goes through from empirical studies and understanding the problem, to theory development, to the development of tools and methods, and to their introduction. They suggest that the findings of each of these phases need to be evaluated separately according to the standards of the disciplines they draw on. They advocate validating the understanding of the problem separately from the approach to address it, and also making the theory or model-building component explicit. This enables the researchers to identify clearly both the scope of their research and the occurrence of potential failures.

4.2. What does success and failure mean for a method?

From the perspective of the creator, a method fails if it is never applied in a practical situation. Many methods that are published fail at this first hurdle. Conversely, we would assume that a method is successful if it provides the expected result and achieves better or faster results than would be achieved without the method. This can also be challenging for methods provided by researchers, because before the users become fluent in using the method, learning the method can slow them down. They will only try using the method again if they see enough promise in it. Many design methods are applied by the researchers themselves or with the help of the researchers and are abandoned without them. Therefore, we can think of a method being successful if it is used repeatedly and independently. However, it can also happen that a method can fail to produce its intended results, but is accepted in practice because it produces useful enough results even though it is limited or unsound.

However, if a method is not successful, the reason may be that any one of the elements has not been successful. A typical failure mode of a method would be that the core idea is sound, but the description of the method is not good enough for independent users to apply the method. Many publications about methods describe only the core idea of the method, rather than the full process. Alternatively, methods might not scale up to larger problems, not because the idea would not apply, but because the representations used in conjunction with the core idea do not work on large-scale problems.

Even with a good description, a method could fail if it is applied in an inappropriate context, in other words, if the problem was out of scope. Alternatively, the proposed method might just not be well thought out.

In industry, many of the more complex methods are only used in parts and not for their intended purpose. For example, a company only might carry out the first few steps of a QFD analysis to get insights or to develop a good visualisation of their problem, such as a house of quality in QFD; and do not proceed to carry out the later stages of the analysis. However, this does not mean that the method has failed

in its application, rather the method has become something that the users own and customise.

4.3. Expectations on validation

Methods are proposed and interpreted with different degrees of prescription from procedures that need to get followed exactly to loose collections of heuristics. In the strictest sense of a 'foolproof' recipe, it becomes necessary to prescribe each of the steps precisely, unambiguously and in sufficient detail to assure that the method can be followed by everybody with a clearly articulated level of experience. If components of a method can be interpreted as recommendations rather than essential parts of the process, then it is left up to the users to decide whether they want to adopt all of them or only some. Components can vary in importance, and tools and representations may be substituted for others within a relatively strict procedure; for example, a function model might be developed using model-based systems engineering (MBSE) rather than a network of function boxes.

What the method is intended to produce can range from possibly useful insights into a problem, to identification of requirements, or evaluations of particular characteristics of a design, or generation of potentially useful solution fragments, to rigorously justified design solutions. This is tightly bound to the theoretical grounding of the method and the claims made for the outputs of the method, as well as to the strictness of the procedures. It is also tightly related to what constitutes success for the application of a method, and how the method can be validated or demonstrated to have succeeded.

What would constitute an appropriate validation depends on the claim behind the method. At one end of the spectrum some methods or elements of methods are based on mathematical calculations, which can be proved to work under given and specified conditions. This aspect of validation is described by Hazelrigg (2003). Other elements of methods are heuristics, rules of thumb that indicate which results can be expected under given circumstances. Heuristics are usually assumed to be valid unless they are contradicted by results that have not been expected or cannot be explained. In practice, the validation of engineering methods is a corroboration through example application, as suggested by Blessing and Chakrabarti (2009). However, corroboration only shows that something can work, but not that it must work. An application example also tells very little about the situations for which a method or an element of a method would work.

Engineers in industry know that methods can rarely be fully validated. They can be persuaded to try out a method, if they can see that it is applied in other companies. In practice, they look for recommendations or endorsements (Gericke *et al.* 2016). Some highly successful methods, like Value Stream Mapping, are based on formalised expert behaviour, as Vermaas (2016a) points out. The reputation of the organisation in which it has been pioneered therefore serves as a proxy for method validation. This looking for successful applications in well-known companies published by well-known academics also explains how the uptake of methods can behave like fashions, where many companies take up a method at the same time and then also discard it at the same time.

4.4. Evaluation versus validation

As we have seen, validation of methods is both conceptually problematic and practically difficult to do. However, what can be done more straightforwardly is evaluation. Verification is about whether you've built the system right. Validation is about whether you've built the right system. Both of these assume that the system is finished, and both assume binary criteria for success. Evaluation is about assessing how well the system works. Success is not binary but a matter of degree. As Krug (2000) points out in the context of usability: 'The point of testing isn't to prove or disprove anything. It's to inform your judgement'.

Evaluation is an integral part of many kinds of system development, and can as easily be directed to looking for ways to make improvements as assessing whether a product is good enough. Testing to evaluate how well prototypes work to guide redesign and further development is an integral part of iterative software development methodologies (see for instance Martin 1991).

Software developers carry out verification and validation but usually refer to it as testing. Terminology and procedures vary, but software development processes regularly include unit testing (for verifying that everything works to spec), usability testing, and acceptance testing (close enough to performing a validation but seldom called that). Some software development processes include formal verification methods for proving the correctness of algorithms. Otherwise, when software developers do use the term, it is also often when talking about methods (Zelkowitz & Wallace 1997), rather than when talking about whether software systems do what their users need them to do. However, verification and validation are important concepts in requirements engineering, where processes are needed for ensuring that the users' real requirements have been understood. Figuring out what a software system needs to do to meet its users' real needs is notoriously hard, and for interactive systems used by human users, relying on the design specification to be the right specification (so treating verification as sufficient for validation) is unlikely to be sufficient for more than ensuring the developers get paid by the clients, so usability testing and acceptance testing will still be required.

While evaluation and validation on the surface appear to be closely related, they are in fact based on different assumptions as illustrated in Table 3.

Any procedure or part of one can be evaluated, by trying to use it and seeing what happens and how well it works, preferably on a real-life problem. However, validation requires conditions to be met that are frequently not met, or cannot be met, by tests of design methods. These include:

- (i) That the range of possible inputs to the procedure can be specified, or can be specified for a defined set of situations, so that it is possible to see that the method works when it is supposed to, or fails in the anticipated ways when the inputs are not those needed for success.
- (ii) That what constitutes success is well enough defined to see whether you have it or you do not.
- (iii) That the procedure itself is well-enough defined to see whether the procedure is being followed and what the procedure contributes to the outcome.
- (iv) That the skills and knowledge required to use the procedure successfully are sufficiently well defined to anticipate whether someone can successfully use it, that is, that the procedure does not depend on tacit knowledge that is so

Table 3. Comparison of validation and evaluation

Validation	Evaluation
Aim is demonstrating that an endpoint has been reached	Aim is to establish to how well a system is working
Success or failure is binary, or a matter of meeting a threshold	Success or failure is a matter of degree
Development has an endpoint, beyond which no further changes are made	Development is ongoing and improvement can go on
Concern with efficacy: whether the system works	Concern with both efficacy and efficiency: how well the system works in terms of cost, effort, etc.
Understanding the quality of the result	Understanding problems with the process with the view to improve it
If extreme and typical cases work, then the whole system will work	Performance should be assessed in real use or simulations of real use

idiosyncratic and poorly understood that you do not know how to train people to have it.

These conditions can only be fully met by methods that are algorithmic procedures for taking defined inputs and calculating well-defined numerical results, such as Finite Element Analysis or the mathematical decision-making procedures discussed by Hazelrigg (2003). Such methods are an important part of engineering practice, especially now since much of what design teams do is analysis rather than ‘synthesis’, and validation is crucial because we are going to depend on the results.

However, these conditions can only be partially met by a lot of design methods. In these cases, validation is not a very helpful concept – or possibly an actively unhelpful one. We should focus on evaluating how well the method works, including evaluating how easy people find it to use as well as how useful the outcomes are. This is especially the case when we cannot define success because it is not a binary or threshold value, when ‘helpful’ is a good outcome, or when partial results from applying part of the method are all we actually need, or when real-world usefulness depends as much on usability and cost-effectiveness as on the results. Methods in this category include procedures for structuring the design process, requirements gathering methods, and ideation methods.

Ancillary components of methods such as software tools can be verified and validated: to ensure algorithms are correct and the tools perform to specification; representations of design information have the information content they need and are sufficiently complete; and checklists are complete. How far procedures can be validated depends on how formally and exactly they can be specified, which depends on what they are for.

As we have argued above, design methods in use are enacted by participants in socio-technical processes, and depend on both the skills and social interactions of the participants. Evaluating methods by observing the method being used or by assessing applications retrospectively should include the human element: the relationship between individual tasks and the skills, motivation and behaviour of

the participants; and relationship between the demands of the procedure and the physical and social organisation of the design team and how the participants interact.

4.5. Implications for evaluation of methods

The elements of the method (the core idea, the representation, the procedure, the intended use and the tool) can and need to be evaluated separately and with different methods. Several different representations and tools can be used around the same core idea; therefore, it is important to learn which one is the most appropriate under particular circumstances. For example, Design Structure Matrices and directed graphs showing dependency networks are isomorphic (Eppinger & Browning 2012) but have different usability issues (Keller, Eckert, & Clarkson 2006). Similarly, different processes could be constructed around the same core ideas. For example, a dependency analysis method could include the modelling process or focus on the analysis. The description of a method, like all descriptions, can be done in many different ways. A method is successful and used in industry only if each of these elements work to a greater or lesser extent. For example, a description can be more or less easy to follow; what matters is that it covers the steps and the assumptions underlying them. Dissatisfaction or failure can occur at each of these steps. A flaw in the core idea of the method might be the end of the development of that particular idea or require significant additional work. The other elements affect the usability of the method and can be fixed and improved following the interactive systems paradigm of evaluation.

Another important factor in the evaluation of a method is its scope. As argued in Section 3.2, methods have an intended scope, that is, they are often developed with a specific problem or class of problems in mind. The characteristics of that problem will influence many of the decisions made regarding the methods. For example, a particular representation might be picked because a group of target users is already familiar with it. In an evaluation based on one example application, it is very difficult to make claims beyond the particular example. One way to widen the scope that can be claimed is by analysing the problems that are addressed carefully and mapping the elements of the methods to these problems. This increases the confidence that similar problems with similar characteristics would also be addressed by the method.

Product validation is often based on worst-case scenarios, under which a product still has to operate safely, as well as typical use cases. Theoretically, this would also be possible for methods, but in practice defining the scenarios could be very difficult. However, an evaluation with multiple examples could increase the confidence in a method as would a characterisation of the problem situation or application context.

4.6. Method evaluation in the context of Design Research Methodology frameworks

The DRM put forward by Blessing and Chakrabarti (2009) sets method development or method improvement explicitly as a goal of design research. The analysis of the current situation and the needs for intervention are established during the first phase of DRM, the *Research Clarification (RC)*. This includes a

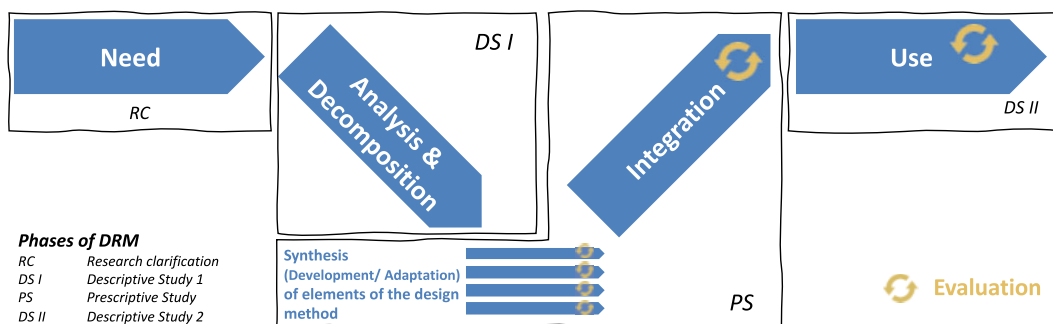


Figure 2. Method development in DRM.

literature review. In the second phase, the *Descriptive Study (DS) I*, the design situation or design activity targeted by the development of a new design method will be analysed in more detail. This includes the definition of the design context that is within the scope of the analysis and what particular aspect of design support needs improvement (decomposition). Figure 2 uses the analogy to the V-model which is widely used in product development to illustrate this (Qureshi, Gericke, & Blessing 2013). It emphasises the decomposition of a deliverable (here, a method) into elements, the separate development and evaluation of the elements, and the evaluation of the integrated deliverable. During the *Prescriptive Study (PS)*, the elements of a method are developed iteratively. This might happen concurrently or sequentially depending on the novelty of the elements and the interdependencies between them, for example, between procedures and representations. Each element requires evaluation before the integration of the individual elements into the actual method, using different methods according to the underlying questions and degree of newness. The PS stage will conclude with an evaluation of the functionality and consistency of the developed method including eventual tool implementations. The usability and usefulness are evaluated during the final stage – *Descriptive Study (DS) II* of DRM.

The spiral of applied research (Eckert *et al.* 2003) acknowledges the centrality of method development to design research while rejecting the view that research is or should be always motivated by solving practical problems (see Figure 3). It splits the design research process in a slightly different way, highlighting the iterative nature of all aspects of it including method development. It divides research leading to the development of methods into four main activities (which in practice are often mixed).

- (i) Empirical research on how designing is done (which may or may not be motivated by awareness of a problem and may or may not uncover one).
- (ii) The development of the theoretical understanding of design (which may or may not inform the core idea of a method).
- (iii) The development of methods and tools with core ideas derived from research (including procedures, descriptions and representations).
- (iv) Introduction into industry, including adaptation to the context and analysis of performance in real use.

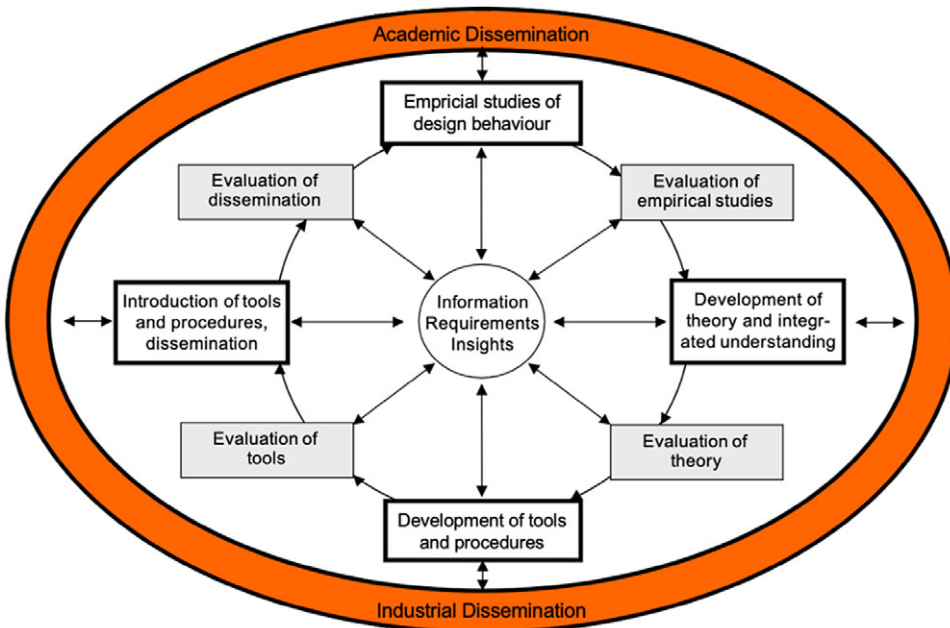


Figure 3. Method development viewed in terms of the spiral of applied research.

At any point in the process, it might become clear that either problems arise and more empirical studies, theory development or tool development are necessary, or conversely that concepts, processes or tools already exist and can be adapted. Eckert *et al.* (2003) advocate evaluating the different types of research findings at different points in the spiral separately, using the techniques and standards of the academic disciplines the research is grounded in. The final evaluation of a method after its introduction in industry (corresponding to the DSII in DRM) is included in the spiral as ‘evaluation of dissemination’, the last step leading back to the beginning.

Vermaas (2016b) reminds us that many methods are in practice evaluated by expert approval, that is, experts use methods and comment on their success or more often are shown methods and invited to comment. Both the experts and the researchers make assumptions about the nature of the problem and tacit skills in problem understanding that are not articulated. Both map the methods to their own understanding and experiences. It is a critical element of method evaluation to make the assumptions and perspectives explicit.

4.7. The test of time

The actual merit or scope of methods only becomes clear over time. Widely used and successful methods, such as TRIZ or QFD, often have multiple publications associated with them by the original developers of the method, but also by other researchers and practitioners as well as articles by users in the trade press. Through these different channels a picture of the purpose, scope, coverage, benefit, and conditions of the methods emerges. However, it is difficult to get a sense for when a method does not work, as failures of application are rarely published. In situations

where a method did not work well or bring benefit, the causes are also not clear, as it might not be obvious whether the method has been correctly or fully deployed. For example, DSMs are widely used (Eppinger & Browning 2012), but some applications just use a basic DSM as a convenient representation of relationships, whereas others make use of algorithms to analyse and modify the relationships between the elements or use dedicated DSM tools. DSMs are isomorphic with directed graphs, and for finding paths graphs are far better suited than matrix structures (Keller *et al.* 2006, provide a rare example of a comparative evaluation).

Successful methods are often used within particular communities whose members share their experiences and develop the method further, and can become the focus of communities of practice (see Wenger 1998), for example, life cycle assessment (LCA). If the elements of a method are clearly identified and evaluated, it is also possible for a community of practice to exchange views about aspects of a method that could be improved. Users might not want to use a method in its entirety and a well-designed and described method allows the users to select the elements that benefit them. For example, many users only pick up on the House of Quality, of the many elements offered by QFD, and very few carry out a complete QFD analysis.

Methods are often created for a particular context and purpose which the users need to understand. Once a method is well established it will be used in many different contexts, for which it might be less well suited. Overclaiming scope can seriously limit uptake by industry. If the method is initially used in an inappropriate context, it is unlikely to ever take off. It is worth noting that different elements of a method have different scopes. For example, matrix representations have been used for application outside DSMs and sometimes graphs are more suitable representations for the results of DSM analyses. For a research perspective, it is always advisable to underclaim the scope of a method and report what one can be confident of. It is easier to expand the claimed scope, coverage and benefit of a method supported by empirical evidence afterwards.

5. Conclusion

When is a method successfully adopted by industry? This is a question that we can and should interpret in two ways: what are the conditions for successful use, including being applied to the right problems; and what constitutes success, which depends on what the method is for.

In this paper, we have argued that descriptions of design methods are encapsulations of design knowledge. Engineering knowledge evolves through time. Adaptations of methods – including cherry-picking components of methods – are not a sign of a failed implementation but a sign of continuous evolution of methods, allowing their application beyond the original intended use. As methods are developed with a limited set of applications in mind their adaptation is a matter of course. An adaptation of a method is thus a source of new knowledge. Expanding our understanding of method use requires analyses of adaptations and customisations of methods to understand why the adaptations were needed and how well they worked in practice. This view on method uptake does not mean that we have to care less about the quality of our results. Practitioners that are seeking advice from academic literature as well as other design researchers expect research results that are carefully evaluated before being published.

The validation of design methods is very different to the validation of engineering products designed by the potential application of methods, where companies take great care and effort to assure that a product works under multiple use and misuse conditions. However, we can also learn from product evaluation approaches. Some different analogies offer ways forward. One is testing components (in engineering) or unit testing (in software): the deliverable (in our case a method) is decomposed into elements that are evaluated individually before being evaluated as an integrated whole. Another is testing tools for users: methods, whether or not they are supported by software tools, have much in common with interactive systems that need to be evaluated to find and correct usability problems and assess how well real users can use them in practice, as well as whether they can perform the tasks they were designed for. We view design methods in use as socio-technical systems that include the engineers applying the methods as well as the tools they use, guidelines they follow and documents they produce. We can look at how well the application of methods works in terms of how all the human and inanimate components interact, which may be influenced by the social organisation of the design process as well as the skills of the engineers.

The analysis of the components of design methods proposed in this paper provides a structure that guides decomposition of methods into their elements and their subsequent evaluation and refinement or adaptation. The evaluation of individual elements helps to trace possible problems, identified during separate evaluations, back to the element that causes the problem. Thus, methods can be continuously improved, and design researchers and practitioners can learn from mistakes and are enabled to adapt a method or its elements to their particular contextual needs and method-ecosystem. The decomposition of a method into its elements and the detailed description of all elements will facilitate the exchange within and across communities of practice and will enable benchmarking of methods for similar design contexts.

A clear communication of the intended use of a method – its purpose, scope, coverage, benefit and conditions – as a complementary element to the core idea, representation, procedure, and tool will ease evaluation efforts and help avoid overclaiming the method's contribution. A more precise claim will facilitate the context-sensitive selection of methods as well as the management of expectations of prospective method users, and ultimately support a more successful adoption of the method in design practice. It will facilitate an understanding of the method's place in a wider ecosystem of methods for different situations and aspects of design problems. An articulation of the intended use will ease evaluation, as the target audience, required rigour, allowed adaptations and expected benefits would be clearly defined. Evaluation against more clearly defined objectives would rely less on interpretation.

References

- AIAA. 1998 Guide: guide for the verification and validation of computational fluid dynamics simulations (AIAA G-077-1998(2002)). In *Guide for the Verification and Validation of Computational Fluid Dynamics Simulations (AIAA G-077-1998(2002))*. American Institute of Aeronautics and Astronautics, Inc.; doi:[10.2514/4.472855.001](https://doi.org/10.2514/4.472855.001).

- Albers, A., Walter, B., Wilmsen, M. & Bursac, N.** 2018 Live-labs as real-world validation environments for design methods. In *DS 92: Proceedings of the DESIGN 2018 15th International Design Conference*, pp. 13–24; doi:[10.21278/idc.2018.0303](https://doi.org/10.21278/idc.2018.0303).
- Alexander, C.** 1971 The state of art in design methodology. In *Design Method Group Newsletter*, pp. 3–7. Kluwer Academic Publishers.
- Andreasen, M. M.** 2003 Improving design methods' usability by a mindset approach. In *Human Behaviour in Design* (ed. U. Lindemann), pp. 209–218. Springer.
- Andreasen, M. M. & Hein, L.** 2000 *Integrated Product Development*. Institute for Product Development, Technical University of Denmark.
- Araujo, C. S., Benedetto-Neto, H., Campello, A. C., Segre, F. M. & Wright, I. C.** 1996 The utilization of product development methods: a survey of UK industry. *Journal of Engineering Design* 7 (3), 265–277; doi:[10.1080/09544829608907940](https://doi.org/10.1080/09544829608907940).
- Barth A., Caillaud E. & Rose, B.** 2011 How to validate research in engineering design? In *Proceedings of the 18th International Conference on Engineering Design (ICED'11), Vol. 2: Design Theory and Research Methodology*. The Design Society.
- Birkhofer, H.** 2008 Tidy up design methods - an approach using elementary design methods. In *International Design Conference - Design 2008, Dubrovnik, Croatia*, pp. 1–8. The Design Society.
- Birkhofer, H., Kloberdanz, H., Berger, B. & Sauer, T.** 2002 Cleaning up design methods - describing methods completely and standardised. *Proceedings of DESIGN 2002, the 7th International Design Conference* 30, 17–22.
- Blessing, L. & Chakrabarti, A.** 2009 *DRM, A Design Research Methodology*. Springer.
- Bly, S.A.** 1988 A use of drawing surfaces in different collaborative settings. In *Proceedings of Computer Supported Cooperative Work '88*, pp. 250–256. ACM Press.
- Bucciarelli, L. L.** 1994 *Designing Engineers*. MIT Press.
- Buur, J.** 1990 *A Theoretical Approach to Mechatronics Design*. Technical University of Denmark.
- Cantamessa, M.** 2003 An empirical perspective upon design research. *Journal of Engineering Design* 14 (1), 1–15; doi:[10.1080/0954482031000078126](https://doi.org/10.1080/0954482031000078126).
- Cash, P., Isaksson, O., Maier, A. M. & Summers, J. D.** 2022 Sampling in design research: eight key considerations. *Design Studies* 78, 101077; doi:[10.1016/j.destud.2021.101077](https://doi.org/10.1016/j.destud.2021.101077).
- Cross, N.** 2008 *Engineering Design Methods* (3rd ed.). John Wiley.
- Cross, N.** 2018 Developing design as a discipline. *Journal of Engineering Design* 29 (12), 691–708; doi:[10.1080/09544828.2018.1537481](https://doi.org/10.1080/09544828.2018.1537481).
- Daalhuizen, J.** 2014 *Method Usage in Design – How Methods Function as Mental Tools for Designers*. Technical University of Delft.
- Daalhuizen, J. & Cash, P.** 2021 Method content theory: towards a new understanding of methods in design. *Design Studies* 75, 101018; doi:[10.1016/j.destud.2021.101018](https://doi.org/10.1016/j.destud.2021.101018).
- de Weck, O. L., Ross, D. & Magee, C. L.** 2011 *Engineering Systems: Meeting Human Needs in a Complex Technological World*. MIT Press.
- Dorst, K.** 2008 Design research: a revolution-waiting-to-happen. *Design Studies* 29 (1), 4–11; doi:[10.1016/j.destud.2007.12.001](https://doi.org/10.1016/j.destud.2007.12.001).
- Eckert, C. M. & Clarkson, P. J.** 2005 The reality of design. In *Design Process Improvement a Review of Current Practice* (ed. P. J. Clarkson & C. M. Eckert), pp. 1–29. Springer.
- Eckert, C. M., Clarkson, P. J. & Stacey, M. K.** 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*, pp. 245–246. The Design Society.

- Eder, W. E. 2007 Design engineering - not just applied science. In *Proceedings of CDEN/ CCEE*, p. 11. University of Winnipeg.
- Eisenmann, M., Grauberger, P., Üreten, S., Krause, D. & Matthiese, S. 2021 „design method validation – an investigation of the current practice in design research. *Journal of Engineering Design* 32 (11), 621–645; doi:[10.101080/09544828.2021.1950655](https://doi.org/10.101080/09544828.2021.1950655).
- Eppinger, S. D. & Browning, T. R. 2012 *Design Structure Matrix Methods and Applications*. MIT Press.
- Fowler, M. (2004): *UML Distilled: A Brief Guide to the Standard Object Modeling Language*. Addison-Wesley Professional, third edition.
- French, M. J. 1999 *Conceptual Design for Engineers* (3rd ed.). Springer.
- Frey, D. D. & Dym, C. L. 2006 Validation of design methods: lessons from medicine. *Research in Engineering Design* 17 (1), 45–57; doi:[10.1007/s00163-006-0016-4](https://doi.org/10.1007/s00163-006-0016-4).
- Geis, C., Bierhals, R., Schuster, I., Badke-Schaub, P. & Birkhofer, H. (2008) Methods in practice - a study on requirements for development and transfer of design methods. In *International Design Conference - DESIGN 2008* (ed. D. Marjanovic, M. Storga, N. Pavkovic & N. Bojcetic), pp. 369–376. The Design Society.
- Gericke, K., Adolphy, S., Qureshi, A. J., Blessing, L. & Stark, R. 2021 Opening up DESIGN methodology. In *The Future of Transdisciplinary Design*, pp. 3–14. Springer; doi:[10.1007/978-3-319-06382-9_1](https://doi.org/10.1007/978-3-319-06382-9_1).
- Gericke, K., Eckert, C.M., Campean, F., Clarkson, P.J., Flening, E., Isaksson, O., Kipouros, T., Kokkolaras, M., Köhler, C., Panarotto, M. & Wilmsen, M. 2020 Supporting designers: moving from method menagerie to method ecosystem. *Design Science* 6, e21; doi:[10.1017/dsj.2020.21](https://doi.org/10.1017/dsj.2020.21).
- Gericke, K., Eckert, C. & Stacey, M. 2017 What do we need to say about a design method? In *Proceedings of the 21st International Conference on Engineering Design, ICED 17*. The Design Society.
- Gericke, K., Kramer, J. & Roschuni, C. 2016 An exploratory study of the discovery and selection of design methods in practice. *Journal of Mechanical Design* 138 (10), 101–109; doi:[10.1115/1.4034088](https://doi.org/10.1115/1.4034088).
- Goodland, M. & Slater, C. 1995 *SSADM Version 4: A Practical Approach*. Mc-Graw-Hill.
- Gregory, S. A. 1966 Design and the Design Method. In *The Design Method* (ed. S. A. Gregory), pp. 3–11. Butterworth.
- Guba, E. G. & Lincoln, Y. S. 1989 *Fourth Generation Evaluation*. SAGE Publications.
- Habermas, J. 1984 *Theory of Communicative Action, Vols 1 and 2* (trans. Thomas McCarthy). Polity Press.
- Hazelrigg, G. A. 2003 Validation of engineering design alternative selection methods. *Engineering Optimization* 35 (2), 103–120; doi:[10.1080/0305215031000097059](https://doi.org/10.1080/0305215031000097059).
- Henderson, K. 1998 *On Line and on Paper*. MIT Press.
- Heymann, M. 2005 *Kunst Und Wissenschaft in der Technik Des 20. Jahrhunderts: Zur Geschichte der Konstruktionswissenschaft*. Chronos Verlag.
- Hubka, V. 1982 *Principles of Engineering Design*. Butterworth Scientific.
- Jacobson, I., Booch, G. & Rumbaugh, J. 1999 *The Unified Software Development Process*. Addison-Wesley.
- Jagtap, S., Warell, A., Hiort, V., Motte, D. & Larsson, A. 2014 Design methods and factors influencing their uptake in product development companies: a review. In *Proceedings of International Design Conference - Design 2014*, pp. 231–240. The Design Society.
- Jones, J. C. 1970 *Design Methods: Seeds of Human Futures*. Wiley Interscience

- Keller, R., Eckert, C. M. & Clarkson, P. J.** 2006 Matrices or node-link diagrams: Which visual representation is better for Visualising connectivity models? *Information Visualization* 5 (1), 62–76; doi:[10.1057/palgrave.ivs.9500116](https://doi.org/10.1057/palgrave.ivs.9500116).
- Kroll, E. & Weisbrod, G.** 2020 Testing and evaluating the applicability and effectiveness of the new idea-configuration-evaluation (ICE) method of conceptual design. *Research in Engineering Design* 31 (1), 103–122; doi:[10.1007/s00163-019-00324-6](https://doi.org/10.1007/s00163-019-00324-6).
- Krug, S.** 2000 *Don't Make me Think*. New Riders.
- Liker, J. K. & Morgan, J. M.** 2006 The Toyota way in services: The case of lean product development. *Academy of Management Perspectives* 20 (2), 5–20.
- Lindemann, U.** 2009 *Methodische Entwicklung Technischer Produkte: Methoden Flexibel Und Situationsgerecht Anwenden*. Springer (VDI-Buch).
- López-Mesa, B. & Bylund, N.** 2010 A study of the use of concept selection methods from inside a company. *Research in Engineering Design* 22 (1), 7–27; doi:[10.1007/s00163-010-0093-2](https://doi.org/10.1007/s00163-010-0093-2).
- Martin, J.** 1991 *Rapid Application Development*. Macmillan.
- Minnemann, S. L.** 1991 *The Social Construction of a Technical Reality: Empirical Studies of Group Engineering Design Practice*. Xerox Palo Alto Research Center Report SSL-91-22.
- National Academy of Engineering.** 2004 *The Engineer of 2020*. National Academies Press; doi:[10.17226/10999](https://doi.org/10.17226/10999).
- Neilson, I. & Lee, J.** 1994 Conversations with graphics: Implications for the design of natural language/graphics interfaces. *International Journal of Human-Computer Studies* 40, 509–541.
- Otto, K.** (2016) Industry Adoption of Design Methods: Where is the Crisis? [keynote presentation], International Design Conference - DESIGN 2016, Cavtat, Croatia online document <https://designfactory.aalto.fi/2016/04/kevin-otto-to-give-keynote-address-at-design-2016/> Accessed: 1.6.2021.
- Pahl, G., Beitz, W., Feldhusen, J. & Grote, K.-H.** 2007 *Engineering Design* (3rd ed., ed. K. Wallace & L. T. M. Blessing). Springer; doi:[10.1007/978-1-84628-319-2](https://doi.org/10.1007/978-1-84628-319-2).
- Pedersen, K., Emlensvag, J., Bailey, R., Allen, J. K. & Mistree, F.** 2000 Validating design methods & research: the validation square. In *Proceedings of DETC'00 2000 ASME Design Engineering Technical Conferences*, pp. 1–12. ASME.
- Pugh, S.** 1991 *Total Design: Integrated Methods for Successful Product Engineering*. Addison-Wesley.
- Qureshi, A. J., Gericke, K. & Blessing, L.** 2013 Design process commonalities in trans-disciplinary design. In *Proceedings of 19th International Conference on Engineering Design, ICED'13*. The Design Society.
- Rozenburg, N. F. M. & Eekels, J.** 1995 *Product design: fundamentals and methods*. In *Product Development: Planning, Designing, Engineering*. John Wiley & Sons.
- Ryle, G.** 1946 Knowing how and knowing that. *Proceedings of the Aristotelian Society* 56, 212–225.
- Ryle, G.** 1949 *The Concept of Mind*. Hutchinson.
- Sargent, R. G.** 2013 Verification and validation of simulation models. *Journal of Simulation* 7 (1), 12–24; doi:[10.1057/jos.2012.20](https://doi.org/10.1057/jos.2012.20).
- Siddiqi, A. & Collins, R. D.** 2017 Sociotechnical systems and sustainability: Current and future perspectives for inclusive development. *Current Opinion in Environmental Sustainability* 24, 7–13; doi:[10.1016/j.cosust.2017.01.006](https://doi.org/10.1016/j.cosust.2017.01.006).
- Tahera, K., Wynn, D. C., Earl, C. & Eckert, C. M.** 2019 Testing in the incremental design and development of complex products. *Research in Engineering Design* 30 (2), 291–316; doi:[10.1007/s00163-018-0295-6](https://doi.org/10.1007/s00163-018-0295-6).

- Tang, J. C.** 1989 *Listing, Drawing, and Gesturing in Design: A Study of the Use of Shared Workspaces by Design Teams*. Xerox Palo Alto Research Center Report SSL-89-3.
- Tang, J. C.** 1991 Findings from observational studies of collaborative work. *International Journal of Man-Machine Studies* **34**, 143–160.
- Tomiyama, T., Gu, P., Jin, Y., Lutters, D., Kind, C. & Kimura, F.** 2009 Design methodologies: industrial and educational applications. *CIRP Annals - Manufacturing Technology* **58** (2), 543–565; doi:[10.1016/j.cirp.2009.09.003](https://doi.org/10.1016/j.cirp.2009.09.003).
- Ullman, D. G.** 2010 *The Mechanical Design Process* (4th ed.). McGraw-Hill.
- Ulrich, K. T. & Eppinger, S. D.** 2008 *Product Design and Development* (4th ed.). McGraw-Hill.
- Vajna, S.** 2014 *Integrated Design Engineering - Ein interdisziplinäres Modell für Die Ganzheitliche Produktentwicklung*. Springer.
- Vermaas, P.** 2016a A logical critique of the expert position in design research: beyond expert justification of design methods and towards empirical validation. *Design Science* **2**, e7; doi:[10.1017/dsj.2016.6](https://doi.org/10.1017/dsj.2016.6).
- Vermaas, P. E.** 2016b Towards precedence that justifies the knowledge claims of design methods. *The Design Journal* **19** (2), 195–204; doi:[10.1080/14606925.2016.1129144](https://doi.org/10.1080/14606925.2016.1129144).
- Vincenti, W. G.** 1990 *What Engineers Know and How They Know it*. Johns Hopkins University Press.
- Wallace, K.** 2011 Transferring design methods into practice. In *The Future of Design Methodology* (ed. H. Birkhofer), pp. 239–248. Springer-Verlag London Limited.
- Wenger, E.** 1998 *Communities of Practice: Learning, Meaning and Identity*. Cambridge University Press.
- Yin, R. K.** 2014 *Case Study Research Design and Methods* (5th ed.). Sage.
- Zanker, W.** 1999 Situative Anpassung Und Neukombination von Entwicklungsmethoden (Doctoral-Thesis). Technical University Munich, Shaker Verlag, Munich.
- Zelkowitz, M. V. & Wallace, D.** 1997 Experimental validation in software engineering. *Information and Software Technology* **39** (11), 735–743; doi:[10.1016/S0950-5849\(97\)00025-6](https://doi.org/10.1016/S0950-5849(97)00025-6).
- Zier, S., Bohn, A. & Birkhöfer, H.** 2012 Methods: using the genome approach for creating complete and adaptable method descriptions. In *Proceedings of the 12th International Design Conference DESIGN 2012*, pp. 1215–1224. The Design Society.