The Sixth International Conference on Design Creativity (ICDC2020)
Oulu, Finland, August 26th – 28th 2020
https://doi.org/10.35199/ICDC.2020.02

On the novelty of software products

Alejandra Beghelli[1,2] and Sara Jones[3]

[1]Faculty of Engineering and Sciences, Univ. Adolfo Ibañez, Chile
[2]Department of Computing, Goldsmiths, University of London, UK
[3]Cass Business School, City, University of London, UK

Abstract: The identification of the key factors that make a tangible product creative has been the focus of significant research in the area of design. However, the same effort for software products is almost non-existent. With the ever-increasing importance of software applications in day-to-day life, such endeavour becomes more necessary. In this paper we report on preliminary results that build on our understanding of how we evaluate the creativity of tangible products, to identify key criteria that can be used in assessing the creativity of intangible software products. We argue that this provides a basis for designers to work together with software developers in pushing the boundaries of software innovation.

Keywords: *novelty, creativity, software product*

1. Introduction

The software industry is increasingly impacting our day-to-day life. From entertainment to health applications, we rely on software products to make our lives easier. As a result, the software industry has become very competitive and is constantly searching for creative ideas that push the boundaries of innovation. In this context, design is key in linking creativity with innovation, by "*helping to find practical applications for research insights*" (Design Council, 2015). With this idea in mind, in this paper we make a first attempt to connect knowledge already available in the design area with the development of creative software.

Unlike tangible products, software is abstract and invisible. Hence, the significant progress made in the study of creative tangible products cannot be easily transferred to software products. However, identifying the key criteria that influence the creativity of software products is of high importance if we want to improve software design practice and innovation. Knowing what makes a software product creative will allow software designers to evaluate the creativity of their ideas in a standard and reliable way. As a result, the search for new software solutions will be more efficient and effective.

In this paper, we discuss factors that determine whether a software product will be considered as novel. In our work, we have built on knowledge already available - regarding the novelty of tangible products - to ask a group of software developers and product managers, from a medium-sized software company, about intangible software products, and the factors that make them original, or novel. Results suggest that the factors of new or extra functionality; user interface, interaction or experience; and technology infrastructure are relevant criteria, when assessing the level of novelty of software products.

2. Previous research

2.1. Novelty of tangible products

The creative product is one of the 4P's of the Rhodes creativity model (Rhodes, 1961). In his seminal paper Rhodes states that "*when an idea becomes embodied into **tangible** form*" we talk about a product. For a long time, this notion of a product being a tangible object has dominated the study of product creativity in the design area. It is not surprising, then, that most factors identified as impacting the creativity of products are those related to their physical attributes.

The creativity of products has usually been defined in terms of two aspects, namely: usefulness and novelty. A few studies also include a third aspect: surprise. In this paper, we focus on novelty, and here briefly review the existing literature on this, and also surprise, since surprise can be argued to be a result of novelty, with some factors overlapping (Hoffman et al., 2007).

Regarding novelty, three broad categories exist: historical, societal and psychological novelty (Boden, 2004; Shah, 2003). They refer to ideas that arise for the first time in human history, in a particular society and in a particular individual, respectively. As the software industry is a global one, it thrives on historical novelty. Thus, we will focus on that category.

Different authors have identified several factors that determine whether a product is novel or not, and what level of novelty might be attributed to it. When products differ from the norm, in relation to these factors, novelty arises. These factors can be summarised as follows:

- **Physical appearance**: the product's physical attributes (shape, material, colour, etc.)
- **Physical/Working Principle**: the physical law that governs the product's behaviour, and how it is applied to the product design
- **Interaction**: the behaviour of the product, understood as how it reacts to different inputs
- **Functionality**: the function performed by the product

Shah (2003) presented a generic formula to evaluate the *a-posteriori* novelty of engineering design products. Specific factors are not given; they must be identified in an ad-hoc manner for each product. Further variants have explicitly used: strategy (not for products, but for abstract descriptions of product ideas), physical/working principle and embodiment (Fiorineschi, 2019). Lopez-Mesa and Vidal (2006) evaluated novelty exclusively in terms of how unexpected the physical appearance of the product is. Sarkar & Chakrabarti (2011) proposed a first scale to evaluate the level of novelty of a product based on different combinations of the previous factors and later, Jagtap (2016) proposed a modification to that scale to incorporate more levels.

In terms of surprise, Becattini et al. (2017) have proposed the most complete classification to date, of factors affecting the surprise triggered by products. They propose 3 main categories:

- **Mismatch Expectation-Interpretation:** If the interpretation of the intentions of the product does not match the user's expectations, for example if a social habit or routine is infringed.
- **Mismatch Expectation-Behaviour:** If the product behaves or works in an unexpected way, given its function. This aspect was also identified as a factor driving surprise by Rodríguez Ramírez (2014).
- **Mismatch Expectation-Structure:** If the physical appearance of a product triggers surprise, for example through the absence of an expected feature, an unexpected combination of existing features or an unexpected modification of a feature.

2.2. Novelty of software products

A software product is a "*combination of software routines, procedures, modules or objects that provide some functionality*" (Schmidt, 2013). As such, a software product ranges from the implementation of a simple algorithm (e.g. an application that only prints a message on screen) to a complex networked application (e.g. an online multi-player video game).

In the area of software development, work on evaluating the creativity of software products has been rather scarce. A first important effort was presented by Couger and Dengate (1992), who identified factors impacting the usefulness and novelty of software, and used these to evaluate the creativity of 6 products. In terms of usefulness, the identified factors were: cost, return on investment, market niche, performance, new functions and the triplet quality-reliability-product life. In terms of novelty, they were: technology transfer, new approach/method, new algorithm, new technology to compute something perceived as incomputable.

More than 20 years later, a literature review on creativity and information systems was published by Müller and Ulrich (2013). There, 88 papers were studied, but only 5 of them somehow referred to the evaluation of ideas or products, and not specifically to software products.

Since then, we have identified two further relevant pieces of research, as follows. First, Kuzmickaja (2015) evaluated the novelty of ideas for mobile services in terms of two aspects: originality and paradigm shift, as proposed by Dean et al. (2006). For each aspect, a 5-point scale was used as shown in Table 1, with levels of paradigm shift being based on work originally by Nagasundaram and Bostrom (1994).

Table 1. Levels of originality and paradigm shift according to Kuzmickaja (2015)

Level	Originality	Paradigm Shift
1	Common, mundane, boring	Paradigm preserving, no influence to future mobile services
2	Somehow interesting	Slight paradigm stretching, introducing few new elements
3	Interesting, shows some imagination	Moderate paradigm stretching, introducing several new elements
4	Unusual, imaginative	Major paradigm stretching, changing the interactions between mobile service users and mobile technology
5	Surprising, ingenious, not expressed before, rare, unusual	Paradigm breaking or shifting, introducing several new elements and changing the interactions between mobile service users and mobile technology

Finally, Kruger et al. (2017) aimed to evaluate the creativity of the final year projects of information systems students using the Creative Product Assessment Model (CPAM) proposed by Besemer and Treffinger (1981), and contrasted this to the expectations of the IT industry. CPAM evaluates 3 aspects of a product: Novelty, Resolution and Style. In terms of novelty, two factors are identified: originality (how infrequent is the product) and surprise (unexpectedness of some characteristic of the product). CPAM is a general technique and as such, it does not identify specific traits that contribute to the novelty of software products. The paper also focuses on IT industry expectations in terms of creativity and does not present results on the creativity assessment of software products.

The concepts introduced in this section will now be further discussed in section 3 below, in relation to the findings from our study.

3. Factors affecting software product novelty

To develop our understanding of the factors that mean that a software product is considered novel, we ran a session during a recent creativity workshop that aimed to collect information regarding what software products were considered, by the participants, to be novel, or original, and why.

Participants in the session were software developers and software product managers from a medium-sized software company in the business of developing application programming interfaces (APIs). The session was carried out in an open plan space on the same floor of the building where participants normally work. Thus, people were free to join or leave the session at any point if something else had to be attended to, and as a result, numbers of participants varied slightly as the session progressed. The session involved 10 groups, with 3-6 people each. No more than 10% of participants were female. People participated in the session for two main reasons. Firstly, their boss was organising it, so there was an element of compulsion. Secondly, though, the best idea from the workshop as a whole would be awarded a prize and included in the roadmap of the company for the year 2020.

The workshop session of interest here focused on the novelty of software products, and lasted approximately 1 hour. During the first part of the session, we asked groups to sort 10 tangible products from least to most original. The participants did not at this stage receive any training on what is considered a novel product. Before starting the sorting task, participants were shown the physical products, the function of each was explained to them and then the products were left on a table for the participants to interact with them, if they so required. The participants were then given pictures of the products to produce a row of them from least to most novel.

After 10 minutes, the participants were asked to finish this task. We then presented to them a simplified version of the method proposed by Sarkar and Chakrabarti (2011) for evaluating the novelty of physical products. We selected this method not only because it lists explicitly the factors affecting the novelty of a product, but also because it assigns an *a-priori* importance to each factor. In this way, participants not trained in product novelty could easily follow the method. The simplified method was presented in the form of the flowchart shown on Figure 1. The next 10 minutes were spent reviewing how well participants' sorting matched the categories proposed in the flowchart, and applying the flowchart to a further 4 tangible products shown on a projection screen.

Once participants were familiar with this method, we asked them to make a list of software products that they considered to be novel at the time they were launched. The software products identified by the participants are listed in Table 2, in alphabetical order of the category they belong to. The categories are based on the 28 categories used in the App Store (Apple, 2020).

Note that because the App Store covers mobile applications, software products associated to a specific hardware platform could not be categorised using this list. In that case, the software product is included in the category "Other". Notice also that this list is not exhaustive: several categories are not covered. For example, from the App Store, software products in categories such as Books, Food & Drink, Kids, Lifestyle, Magazines & Newspapers, News, Video, Travel were not mentioned. If the categorization of another source repository was used (e.g. Source Forge, see Table 2 in Linares-Vásquez et al., (2014)) the same phenomenon would be observed: several categories would not be included. Thus, as we discuss further below, this exercise must be considered as a preliminary approach to developing our understanding of the characteristics that make a software product original, that should ideally be extended by further studies.

Once the groups of participants had created their lists of software products that they considered to be novel at time of launching, we asked them to identify what factors they thought made these software products novel. To do so, we asked them to use as inspiration the factors proposed by Sarkar and Chakrabarti (2011). The factors identified by the groups of participants were:

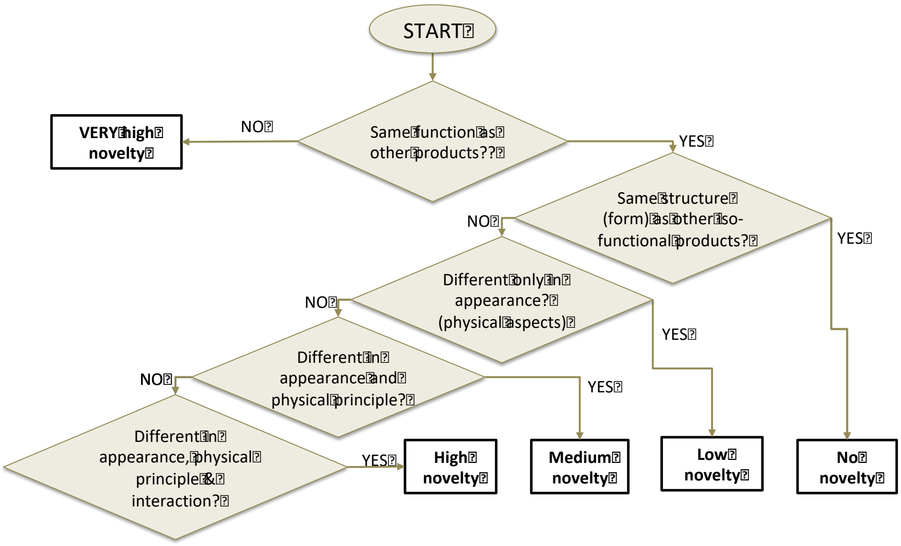
START
Same function as other products??
NO
VERY high novelty
YES
Same structure (form) as other iso-functional products?
NO
YES
Different only in appearance? (physical aspects)
NO
YES
Different in appearance and physical principle?
NO
YES
Different in appearance, physical principle & interaction?
YES
High novelty
Medium novelty
Low novelty
No novelty

Figure 1. Simplified flowchart for classifying the level of novelty of physical products

Table 2. Software products considered by participants to be novel at the time they were launched

CATEGORY	APPLICATIONS
Business	Google Suite (sheets, docs, etc), Google Auth, Google Search; Swift Key, Docu Sign; Amazon Web Services (AWS)
Developer Tools	Tensor Flow
Education	Anki
Entertainment	Snapchat filters
Finance	Monzo, Revolut, Bitcoin, Transfer Wise, Pay Pal
Games	Playstation, Wii Golf, Pokemon Go
Graphics & Design	Deep Fakes
Health & Fitness	Head Space, CAT scan
Music	Shazam, Spotify
Navigation	Google Maps, Uber, Waze
Shopping	Sizer, Airbnb, Wetherspoons
Social Networking	Snapchat, Tik Tok, Tinder, Twitter
Other	Oculus Rift, Leap Motion, Alexa, Siri, WWW

- **New Functionality**: If the product performs a task that no other product has performed before. In the sense that Sarkar & Chakrabarti (2011) define this aspect, it would correspond to a software product that performs a task that no other product, either software or not, has performed before. An example of this case is Deep Fakes. In the sense that Couger and Dengate (1992) define this aspect, it would correspond to a software product that makes computable something that was not computable before. An example of this case is Waze. This factor is in line with the findings of Rose and Forneaux (2016), where a survey of 236 papers showed that new software functionality was identified as the most common form of software innovation.
- **Extra Functionality**: If the product adds a new function to what similar products already offer. An example of this is the filters of Snapchat. Multi-functional products are not considered in the classification of Sarkar & Chakrabarti (2011), where the focus is on single functional products, since this is the most common situation with tangible products. However, software products are more prone than tangible products to be multi-functional.
- **User Interface**: If the product offers a new interface with respect to similar products. For example, the World Wide Web at its launching time. This aspect might be equivalent to the Physical Appearance factor of Sarkar & Chakrabarti (2011). In a software product the change of interface can be as simple as changing the distribution of elements on a screen, or something more radical, like adding sensing hardware, as Leap Motion did. In that case, the change of interface comes hand in hand with a change in the interaction with the user (as below).
- **User Interaction**: If the product offers a new way of interacting with the user. Examples of this are the virtual assistants Alexa or Siri, or the hand tracking system Leap Motion.
- **User Experience**: If the product changes the user experience offered by similar products. Examples of this are Uber and Oculus Rift. This is an aspect not considered in the evaluation of novelty given in Sarkar & Chakrabarti (2011). Usually, a new User Experience is the result of significant changes in User Interface and Interaction. According to Kruger et al. (2017) this aspect was mentioned by 2 out of 9 managers, when asked what important aspects of a software product should be included in an assessment of its creativity.
- **Technology infrastructure:** If the product offers a completely new technology infrastructure to perform usual functions plus perhaps a few functionalities resulting from the new technology. Examples of this are Amazon Web Services (cloud computing) or Google Suite (online applications). This aspect might be related to the factor Physical Principle identified by Sarkar & Chakrabarti (2011).

Others factors mentioned by our participants, that are related to usefulness rather than novelty, and are therefore beyond the scope of this paper, were performance (if the product offers improved performance), and cost (if the product offers operational savings with respect to similar products).

4. Software novelty evaluation

Based on the factors identified by the participants in our creativity workshop, as listed in the previous section, and using the flowchart of Sarkar & Chakrabarti (2011) as inspiration, we propose the flowchart in Figure 2 as an approach to evaluating the novelty of a software product.

Using the flowchart in Figure 2, the following are examples of software products in the different categories:

- **Low novelty** [NI: New Interface]: Operating systems of mobile phones that tend to be different just in terms of the interface they offer to the user belong to this level.
- **Low-Medium novelty** [NIEF: New Interface + Extra Functionality]: Several browsers belong to this category. Most of them only change the interface they present to the user and add one or few features to make navigation easier.
- **Medium novelty** [NIEFT: New Interface + Extra Functionality + New Technology Infrastructure]: Amazon Web Services and Google Suite can be classified as software products of medium novelty, given that their main source of novelty is the technology infrastructure they use to offer their services. This new infrastructure allows them to modify the interface and add some extra functionality as well.
- **Medium-High novelty** [NIEFTIN: New Interface + Extra Functionality + New Technology Infrastructure + New Interaction]: Wetherspoons' pubs allow ordering your food in the pub and paying for it without leaving your table. In that way, their app offers a new interface to go through the menu and order the food, the extra functionality of paying in the same app and as a result, a new interaction of the user with the food service.

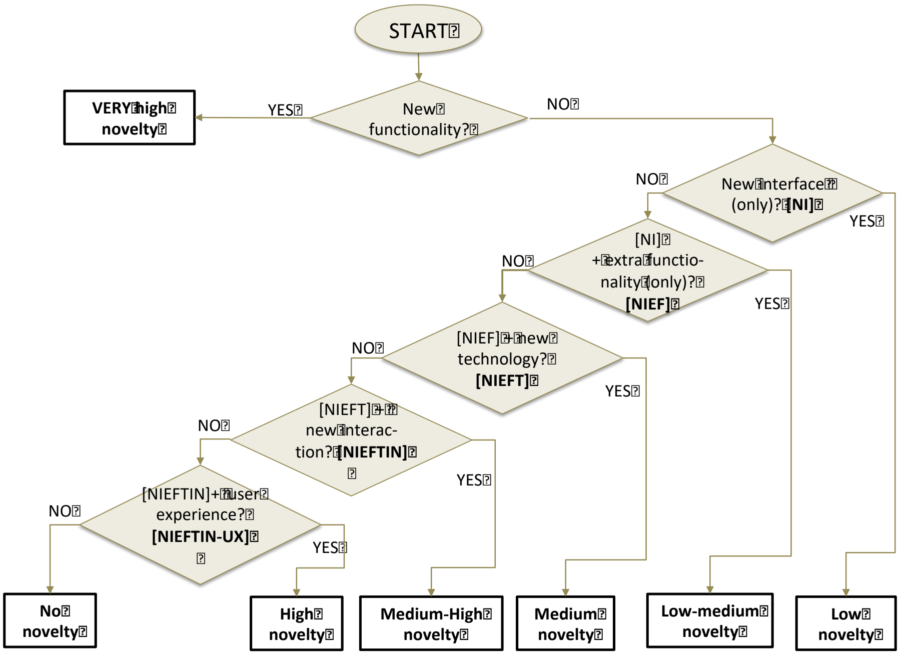
START
New functionality?
YES
NO
VERY high novelty
New interface (only)? [NI]
NO
YES
[NI] + extra functio-nality (only)? [NIEF]
NO
YES
[NIEF] + new technology? [NIEFT]
NO
YES
[NIEFT] + new interac-tion? [NIEFTIN]
NO
YES
[NIEFTIN] + user experience? [NIEFTIN-UX]
NO
YES
No novelty
High novelty
Medium-High novelty
Medium novelty
Low-medium novelty
Low novelty

Figure 2. Flowchart for classifying the level of novelty of software products

- **High novelty** [NIEFTIN-UX: New Interface + Extra Functionality + New Technology Infrastructure + New Interaction + New User Experience]: Uber was a software product that completely changed the user experience of taxi users. The new interface with the taxi service

was the mobile phone screen (instead of a phone call), with the extra functionalities of knowing exactly when the car would arrive, how much the trip would cost and what route should be the shorter one. To enable this, the new technology of geolocation was included.
- **Very high novelty** [NF: New Functionality]: Deep Fakes and Pokemon Go can be considered in this category. One example of the former is "Synthetizing Obama", launched in 2017, that depicts former US president Obama realistically mouthing the words of a different audio track. The latter was the first video game to allow capturing Pokémon in real locations.

Notice how these factors might be related to those used by Sarkar and Chakrabarti (2011): functionality has the same meaning in both flowcharts; interface might be related to the physical appearance; user interaction/experience can be connected to the product interaction; and new technology infrastructure to a new physical principle being used in achieving the function of a physical product.

5. Conclusions and Future Work

In this paper we have identified key criteria used in assessing the novelty of software products, and proposed a new approach to using these criteria, in assessing levels of novelty in such products. Using the work of Sarkar and Chakrabarti (2011) as inspiration, we asked a group of software industry workers to identify what aspects of a software product make it novel. Based on the collected information, we built a flowchart that allows the classification of a software product into one of 7 levels of novelty using the dimensions of new or extra functionality; user interface, interaction or experience; and technology infrastructure. We argue that this approach may provide a basis for designers to work together with software developers in using a better understanding of what constitutes novelty, and hence creativity, in software products, to push the boundaries of software innovation. Using the criteria we have identified here, software innovation challenges can be better addressed by focusing efforts on the aspects that most impact the perception of novelty in software products. The main limitations of this work are the focus on only one previous work on product novelty (Sarkar & Chakrabarti, 2011); the limited number of original software products used to identify the possible factors affecting software novelty; and the lack of empirical validation of the proposed flowchart. Further work should focus on overcoming these limitations.

References

Apple, (2020). https://developer.apple.com/app-store/categories/. Last accessed 4th January 2020

Becattini, N., Borgianni, Y., Cascini, G. and Rotini, F. (2017). Surprise and design creativity: investigating the drivers of unexpectedness. International Journal of Design Creativity and Innovation, 5(1), 29-47

Besemer, S. And Treffinger, D. (1981) Analysis of creative products: Review and Synthesis. Journal of Creative Behaviour, 15(3), 158-178

Boden, M. A. (2004). The Creative Mind: Myths and Mechanisms. Routledge, Taylor & Francis Group.

Couger, J., Dengate, G. (1992) Measurement of creativity of I.S. products. *Proceedings of 25th Hawaii International Conference on Systems Sciences*, Kauai, USA

Dean, D., J. Hender, T. Rodgers, and E. Santanen (2006) "Identifying Quality, Novel, and Creative Ideas: Constructs and Scales for Idea Evaluation", *Journal of the Association for Information Systems*, (7)10, 646–699.

Design Council (2015), Innovation by Design. Available at https://www.designcouncil.org.uk/sites/default/files/asset/document/innovation-by-design.pdf. Last accessed 4th January 2019

Fiorineschi, L., Frillici, F., Rottini, F. (2019). Orienting through the variants of the Shah's a posteriori novelty metric", International Conference on Engineering Design, ICED 2019, Delft, The Netherlands

Hoffmann, O., Cropley, D., Cropley, A., Nguyen, L., and Swatman, P. (2007). Creativity, requirements and perspectives. *Australasian Journal of Information Systems*, 13, 159–175.

Jagtap, S. (2016) Assessing Design Creativity: Refinements to the novelty assessment method. *In Proceedings of DESIGN 2016*, Dubrovnik, Croatia

Kruger A., Matthee M., Turpin M. (2017) Information Systems as Creative Products: What Are Industry's Expectations?. *Communications in Computer and Information Science*. Liebenberg J., Gruner S. (eds) ICT Education. SACLA 2017, vol 730. Springer, Cham

Kuzmickaja, I., Wang, X., Graziotin, D., Dodero, G. and Abrahamsson, P. (2015) In Need of Creative Mobile Service Ideas? Forget Adults and Ask Young Children. *SAGE Open*, 5(3), 1-13

Linares-Vásquez, M., McMillan, C., Poshyvanyk, P. and Grechanik, M. (2014). On using machine learning to automatically classify software applications into domain categories. *Empirical Software Engineering*, 19(3), 582-618

Lopez-Mesa, B. and Vidal, R. Novelty metrics in engineering design experiments. *Proceedings DESIGN 2006, the 9th International Design Conference*. Dubrovnik, Croatia, 2006.

Müller, S. And Ulrich, F. (2013) Creativity and Information Systems in an Hypercompetitive Environment: A literature review. Communications of the Association for Information Systems, 32(7), 175-200

Nagasundaram, M. and Bostrom, R. (1994) The Structuring of Creative Processes Using GSS: A Framework for Research. *Journal of Management Information Systems*, 11:3, 87-114

Rodríguez Ramírez,E.R.(2014). Industrial design strategies for eliciting surprise. *Design Studies* 35(3) 273–297.

Rhodes, M. (1961). An Analysis of Creativity. *The Phi Delta Kappan*, (42)7, 305–310.

Rose, J., and Furneaux, B. (2016). Innovation Drivers and Outputs for Software Firms: Literature Review and Concept Development. *Advances in Software Engineering*, 2016, 1–25.

Sarkar, P. and Chakrabarti, A., 2011. Assessing design creativity. *Design studies*, 32(4), pp.348-383.

Satzinger, J. W., M. J. Garfield, and M. Nagasundaram (1999). The Creative Process: The Effects of Group Memory on Individual Idea Generation. *Journal of Management Information Systems* (15) 4, pp. 143-160.

Schmidt, R. (2013). Software Engineering. Architecture-driven software development. Morgan Kaufmann.

Shah, J.J. and Vargas-Hernández, N (2003). Metrics for measuring idea effectiveness. *Design Studies* (24), 111-134