Using MVP in Hardware Industry – Reduce “Time to Market” by Taking Risks

Christian Reidl and Daniel Valtiner
Infineon Technologies Austria AG, Villach, Austria
Email: christian.reidl@infineon.com, daniel.valtiner@infineon.com

***Abstract*—The agile manifesto and the resulting change of software project management towards agile approaches did not only change the software world. Branches such as the service or financial sector took over agile approaches and tools. Usage of a “minimum viable product” (MVP) is one approach to accelerate time to market. While software products can be brought to a new version by a recompilation the hardware production has limiting boundary conditions, that cannot be neglected. However, the mindset of an MVP could also help hardware projects like the ones in the chip industry to optimize the development in respect to time to market and address the design gap. The results show how to use MVP in hardware development and the balancing between features and time to market is discussed.**

***Index Terms*—agile, MVP, minimum viable product, chip development, semiconductor industry, design gap, strategic management.**

I. Introduction

For about two decades the term agile development generates the impression to have conquered the software industry. Agile development approaches exist in several shapes. Each of them, in its expression, still pursues the same goal: focusing the customer and providing the development team with the means to focus on the customer’s needs [1]. This needs to be fulfilled even in the case those needs are changing while the product matures.

Some of the approaches like e.g. Scrum, Kanban, Extreme Programming (XP) or Crystal are now the equivalent of Agile Development. To fulfill the customer needs, a fast and close exchange with the customer is in focus to iterate to the desired product [2].

Due to numerous success stories in the software industry, other areas, like the service or the financial sector, have also started to integrate the available approaches into their processes [3], [4]. However, for hardware industry the different conditions have to be considered. Between concept phase and shipping a production phase exists. That phase consumes a not neglectable time.

An iterative exchange with the customer is limited, when each delivery of an interim product version needs many weeks for production. This time is added to concept and development efforts to implement the dedicated customer feedback.

In semiconductor industry additional particularities are present that have to be considered. The production of a semiconductor product can last several weeks or even months due to the manufacturing processes. Additionally product specific lithography masks are needed for the production that could cost up to 20 million Dollars [5] for products using state of the art semiconductor technologies.

Agile approaches have to be adapted to fit to semiconductor industry’s needs. Agile development methodologies cannot be transferred unchanged from software world into the development of semiconductor products.

II. Addressing the Design Gap

One key issue that started the introduction of agile project management has been the inability to cover the high demand of software in the 1990’s. The requirements for the software or the industries that use that software changed faster than it could be created. A high amount of software projects has been canceled or did not meet their goals [6]. In the semiconductor industry a similar phenomenon can be observed called the “design gap”[7]. The design gap describes the inability of the product development to use the possibilities that are provided by technological progress.

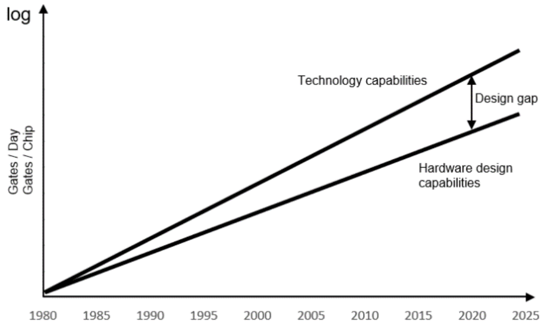

Figure 1. Design gap

The relation of technology and design capabilities including the design gap is shown in Fig. 1. The technological progress is the ability to place more components on the same area of silicon. A new technology node shrinks the components size.

The technological progress regularly doubles the structure on the same area in a constant time span (e.g. 24 month). This technological shrink started in 1960 with a minimal structure size of 50 µm [8] and is actually in 2021 below 5nm. It is still following Moore’s law, that states that it is an exponential behavior as many other topics in the semiconductor industry [9].

Manuscript received February 28, 2021; revised May 20, 2021.

doi: 10.18178/joams.9.4.83-87

The possibility of more structures on the same silicon die size has evolved first in the sector memory applications. Memories are regular structures that can easily be doubled with the only need of an adapted control logic to address all available memory cells.

Doubling the size of products which consists of irregular structures like microcontroller or other logic chips would mean much more than a doubling of the complexity. Approaches from information technology show the connection of different kinds of complexity and their connection to the “design gap”[7].

III. The Sky is the Limit – and Verification

The number of applications based on semiconductor products heavily increased in the last decades. The revenues in the semiconductor market increased from 4 billion US-Dollar per month to nearly ten times as much in 2019 [10]. The need for more performance is considered to increase also in the near futures with the seemingly rise of new technologies like autonomous and electrified car.

Similar behavior is visible at the smartphone market. The strive for small gadgets with long battery endurance and high performance depicts an incessant competition.

Nevertheless continuously adding features leads to additional verification effort. That might be a bottle neck for many developments. The effect of increasing complexity can be illustrated using a simple example. Assuming to have 9 binary settings (9 bits) 512 combinations of that 9 settings are possible (2^9=512). The development has to implement 9 settings. If those settings can be combined unrestrictedly 512 combinations have to be verified (2^9). By adding a tenth option, the amount of possible verification cases doubles to 1024 (2^{10}).

Verification effort constantly increases compared to the implementation [11]. The verification methodologies at the chip development have been developed to self-testing regression simulations. Otherwise it might be already challenging for verification to handle the actual efforts at all.

The verification effort could be addressed by restricting the use cases. Even with the same amount of feature settings but restrictions in the use cases, that allows the possible combinations of the settings, the complexity could be made manageable. A proper definition of limited use cases according settings leads to a reduction of unnecessary complexity.

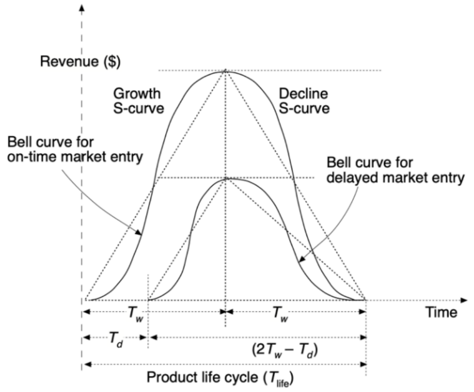
Revenue ($)
Growth S-curve
Decline S-curve
Bell curve for on-time market entry
Bell curve for delayed market entry
Time
T_w
T_w
T_d
$(2T_w - T_d)$
Product life cycle (T_{life})

Figure 2. Time to market - revenue within product life cycle.

IV. Focus “Time to Market”

Higher development effort results in a longer time until the market can be entered. In a highly competitive environment time to market is essential for economic success.

Biren Prasad discussed a possible model how to quantify time to market in his article “Analysis of pricing strategies for new product introduction” [12]. In this article he as well assesses the cost of missing the right point in time.

The model consists of several assumptions e.g. all products are fulfilling the needs and no competitor attacks the market by pricing strategies. Further the model is based on the assumption of a market window that shows the maximum possible revenue for all competitors at the same point of time. The point is called $T_{W.}$ and is located in the middle of the product life cycle (T_{Life}). Each competitor shows the same growth rate α. Therefore, the maximal possible revenue is $\alpha T_{W.}$

Fig. 2 shows the revenue curve for the competitor with the on-time market entry together with the curve of a competitor with delayed market entry. The delay when the late competitor enters the market is named T_d. Assuming that the growth S-curve is symmetrical it can be narrowed by straight lines that combine to triangles. The overall revenue is the integrated area beneath the lines.

Assuming the same growth rate the first noticeable point is, that the competitor with the later market entry cannot reach maximum peak revenue anymore until the downturn starts.

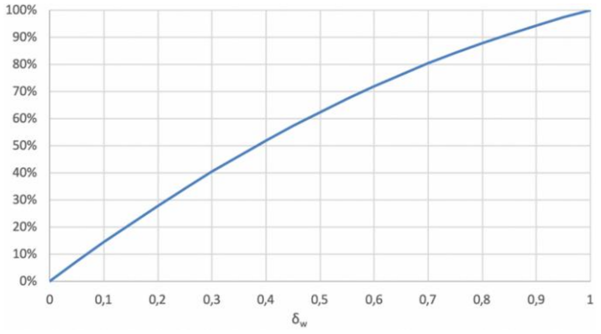

Figure 3. R_{loss} in relation to delayed market entry.

The resulting loss of revenue in percentage can be quantified by the difference between the areas of the triangles divided by the area of the on-time triangle.

$$R_{loss} = \frac{R_{early} - R_{delayed}}{R_{early}} = \frac{[(3T_w - T_d)T_d]}{2(T_w)^2} \tag{1}$$

This is the formula delivered by Brian Prasad [12]. Through replacing T_d by $\delta_W T_W$ formula 1 can be simplified to formula 3 by using formula 2.

$$T_d = \delta_w T_w \tag{2}$$

$$R_{loss} = \frac{3}{2}\delta_w - \frac{1}{2}\delta_w^2 \tag{3}$$

The factor δ_W has a range from 0 to 1 where 0 means that there is no delay in the market entry by the competitor the loss of revenue is 0%. At δ_W=1 the competitor entry is on the market window peak.

As visible in Fig. 3, it is a relation containing a quadratic component. The loss of revenue at a market entry at the peak is 100%. This means, no revenue earnings are expected anymore, when the time window of the growing market is missed.

It is even more severe, since it a nonlinear function and thus the loss caused by the delay is higher than one would assume. Missing 50% of T_W does not lead to only miss 50% of total revenue but even 62.5%. Missing 25% of the growing market leads to revenue loss of already 34%.

A possible explanation for the complete loss of any expected revenue when entering the market at the end of the growing market is that at in the declining market no new sales contracts are concluded anymore. The already existing contracts are fulfilled till up to the end of the product life. At the declining market, the customers already choose the next generation of that type of product.

V. Minimum Viable Product

One of the methods connected to agile development is the usage of a "minimal viable product" (MVP) [13]. It is known for example at mentions by Eric Ries [14] that especially focuses startups to speeding up the market entry.

The MVP approach questions if products are filled with a lot of functions and features right from the beginning. Especially start-up companies might not have the financial capabilities to do a long development until they can enter the market [14].

Ries' assumption is that it does not make sense to delay the market entry by adding functionality the customer might not need.

The focus of an MVP is to enter the market as soon as possible with a version that contains just enough functionality that the customer can use it. This starts the learning cycles to get feedback from the market and the customer.

By entering the market, the build-measure-learn cycle is started with the expectation that fast deliverance and also fast customer feedback will lead to an optimized product. It targets to contain exactly what the customer needs. There may be several iterations until the final product is figured out.

The *build-measure-learn cycle* is not only important for the developers. Also, the customer itself might not know all requirements right from the beginning. Early deliverables also help him to understand his own needs.

The MVP approach does not mean that only products with reduced features are developed but a reduced feature set is the base for the further development cycles. They are used to reach the optimized mature product. By considering inputs from the learning cycles the product is improved according to the customer feedback until it contains the desired functionality.

This approach shows advantages like:

- Fast market entry
- Fast customer feedback
- Avoiding cost and delay by developing unused features

Especially, avoiding the development of unused features is speeding up the development process. An examination of the Standish Group implies, that up to 50% of a feature set of today's software products are rarely or not used at all [15]. In the expression of agile manifesto, this is "waste"[16].

It may be that some features are implemented at the beginning of the development iteration that are not needed by the customer. The early feedback will stop that development in an early phase. The customer feedback prevents that the unused feature is more detailed, finalized and verified.

VI. MVP in Semiconductor Projects

Before going into the explanation of how an MVP could look like in the semiconductor hardware industry, assumptions regarding the development setups are done.

The developments are executed within projects having a project owner, project manager and a project team. The team is set up with experts from different technical disciplines out of a combined pool. By project and pipeline planning the team members are allocated to the projects accordingly.

According to the desired functionality different experts might be needed from application-, concept-, design-, layout-, verification- and test engineering. To create a system on chip (SoC) that should provide a solution for the customers wishes different blocks and modules are combined into one big circuit. This circuit is transferred to a layout that is the basis for the mask set that is needed for production. The mask set is unique for each design and each design step. Single modules may be tested in extra test chips separated from the SoC to verify them separate in silicon.

At a first view it does not seem be economically to create several iterations of a circuit to find out the customer needs. Having several iterations with development and following production this approach might last years until the final requirements are collected and the final product is created. In the software world a new version "only" consists of a recompilation of adapted source code.

Additionally, the cost of a mask set for several productions will sum up to an expensive amount.

Neither from time perspective nor from cost perspective chip production of intermediate versions to derive the final customer requirements is a reasonable way in this context.

Intermediate versions could be delivered with help of software models. The customer can use that model to implement the module into his software environment for system evaluations. Basic requirements can be checked and approved or changed by the customer in quick iterations.

To check a new product in an already existing system, hardware solutions like FPGA (Field Programmable Gate Array) boards can help. With this a functional description can be synthesized into a programmable board like firmware on a microcontroller. The board behaves like the final hardware that is targeted to be produced. The planned circuit can be emulated with that boards directly in the customers system. Adaptions can be done with new synthesis of code.

Therefore, the creation of FPGA-Solution might be an adequate way of close interaction with the customer for a learning phase for both partners. The developer as well as the customer can check very fast the characteristics of the planned product. Nevertheless, also this is only a first step within the product development to focus the final product requirements.

The requirements definition of the product will now indirectly define everything that is also relevant for time to market. The number of features defines the amount of effort for concept engineering, design engineering, the teams for pre-silicon (before production) and post-silicon (after production) verification.

Increasing number of features within the products will extend the duration of the single project within the product pipeline. The project owner of the following project has to wait until his project can start. To get the most revenue the number of features of the planned product might be increased to address several applications.

This again leads to longer development time and the next project owner has to wait for his project. He or she might increase the functionality of his project, too.

This feedback loop leads to steadily increasing features and therefore also increasing development efforts like resources and time. In that scenario, the maximized number of features is in focus instead of time to market. Such approaches can be seen more as unfounded assumptions than realistic planning. Features might be included with the expectation that it might be beneficial in future.

The result of that feature focused projects is shown at the left side of Fig. 4. The target application should be covered, since it was basis of the development project. Some near applications might also be usable. Other applications cannot be addressed, because certain sub features have not been implemented despite additional effort.

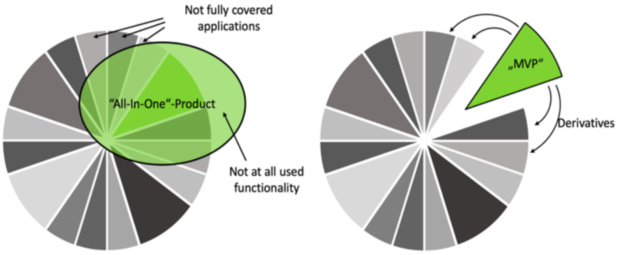

Figure 4. Full feature product vs. MVP

In addition, there is an amount of effort that had been spent for the implementation of features, that are not usable at any of the target applications. Since the target application is delayed till the whole project is implemented, the loss of revenue due to the delayed market entry has to be added to the cost of this type of development approach.

At the right side of Fig. 4 the approach with "MVP" is shown as it could be used in the hardware development. The approach is to focus on what is really needed and not on features that might be useful. The development of the tailored product with minimal extensions has to be focused first. It can be a development with optimized effort in respect to time to market.

As in software development, this does not have to be the end of the development. The existing product can be a basis for the next targeted applications that can be created with minimized additional effort. A development team to create a derivative can be minimized and is less effort than starting a completely new product.

The first product can be used to collect the requirements for other neighboring applications. As stated, the learning phases are needed by the developer as well as the potential customer. By delivering a product that is already close to a dedicated version, the customer can identify the needed changes. However, he already has a close experience of the final product.

Comparing the usage of MVP in software and hardware development would lead to following statements:

MVP-Software: *Don't waste time to develop features you don't know whether they are needed or not.*
MVP-Hardware: *Don't waste time to develop features you know that they are not needed.*

In software development an MVP is used to find out the final customer requirements. In hardware development MVP could be used to focus only the already known customer requirements.

VII. Face the Risk

In many product definitions the cost of delayed market entry might not be considered systematically enough. It is compared with the risk of losing opportunities by additional features. In such an environment for business decisions, using the MVP approach may not be straight forward. Decisions of what will be a required feature of the product are of high importance. The same importance applies for the decision-making process of what will be implemented.

Limiting the feature set is a challenging task. Applications might only differ in small parts of features or even only in sub features. The line between a feature that is worth to be implemented and a feature that endangers the time to market is thin. The cost of mask sets for an extra derivative might be an additional argument to simply take an additional feature to the actual development.

Finally, there is no guarantee to make the product fit to the market requirements of the next decades by adding a lot of features. Attempts might endanger time to market for todays needed features.

To conduct an MVP project a clear risk assessment has to be done in order to make clear decisions between missing opportunities by missing features or by missing time to market.

In case of a decision towards an MVP approach, the platform approach should be considered right from the beginning to ensure, a product design and its team is already prepared for new derivatives.

VIII. Conclusion

The paper presents a possible way of applying the approach of a minimum viable product coming from the software industry to the semiconductor industry. The transfer of the learning cycles has to be considered in different approaches like software models or FPGA solutions since the particularities of the chip production or hardware development in general in respect to time and cost are limiting factors for repeating iterations.

Focusing target applications saves development resources that are already limited in respect to the design gap. Additionally, the shortened development time is an important step towards a fast market entry. The potential revenue loss of late market entry is severe as the shown model shows.

Fast development of the target application to optimize the time to market followed by derivatives in a platform approach is the basic principle of MVP in hardware development.

Finally, the price for the fast market entry is a thoroughly executed risk assessments that decides between missing opportunities and securing the market entry.

Therefore, it is crucial to limit the product development to the features that are already needed and are backed with real market opportunities. Securing and improving market positions depends on the ability to deliver the needed product in time.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

CR conducted the research; drafted the manuscript and designed the figures and wrote the paper. All authors discussed the results and commented on the manuscript.

References

[1] M. Kropp and A. Meier, “Agile success factors - A qualitative study about what makes agile projects successful,” 2015.

[2] J. Kisielnicki and A. M. Misiak, “Effectiveness of agile compared to waterfall implementation methods in IT projects: Analysis based on business intelligence projects,” *Found. Manag.*, vol. 9, no. 1, pp. 273–286, Oct. 2017.

[3] D. Beerbaum, “Applying agile methodology to regulatory compliance projects in the financial industry: A case study research,” *J. Appl. Res. Rev.*, Jan. 2020.

[4] S. Lamberth-Cocca and T. Meiren, “Towards a reference model for agile new service development using the example of E-mobility service systems,” in *Procedia CIRP*, vol. 64, pp. 259–264, 2017.

[5] K. Flamm, “Measuring moore’s law: Evidence from price, cost, and quality indexes,” National Bureau of Economic Research, Cambridge, MA, w24553, Apr. 2018.

[6] H. Mulder, *The Chaos Report*. 1994.

[7] B. Menhorn and F. Slomka, “Confirming the design gap,” in *Advances in Computational Science, Engineering and Information Technology*, vol. 225, D. Nagamalai, A. Kumar, and A. Annamalai, Eds. Heidelberg: Springer International Publishing, 2013, pp. 281–292.

[8] C. Mack, “Fifty years of moore’s law,” *Semicond. Manuf. IEEE Trans. On*, vol. 24, pp. 202–207, June 2011.

[9] G. E. Moore, “Cramming more components onto integrated circuits,” vol. 38, no. 8, p. 6, 1965.

[10] ‘2023 - Mikroelektronik - Trendanalyse bis 2023.pdf’. Accessed: Aug. 09, 2020. [Online]. Available: https://www.zvei.org/fileadmin/user_upload/Presse_und_Medien/Publikationen/2019/April/Mikroelektronik_Trendanalyse_bis_2023/ZVEI_Mikroelektronik-Trendanalyse-bis-2023.pdf.

[11] B. Bailey, “Is verification falling behind?” *Semiconductor Engineering*, Dec. 21, 2017. [Online]. Available: https://semiengineering.com/is-verification-falling-behind/ (accessed Jan. 22, 2021).

[12] B. Prasad, “Analysis of pricing strategies for new product introduction,” *Pricing Strategy Pract.*, vol. 5, no. 4, pp. 132–141, Dec. 1997.

[13] V. Lenarduzzi and D. Taibi, “MVP explained: A systematic mapping study on the definitions of minimal viable product,” in *Proc. 2016 42th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, Limassol, Cyprus, Aug. 2016, pp. 112–119.

[14] E. Ries, *The Lean Startup: How Today ́s Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses*, 1. Aufl. USA: Crown Business, 2011.

[15] The Standish Group International, Inc., ‘Exceeding Value’, 2014.

[16] K. Beck *et al.*, “Manifesto for agile software development,” *Manifesto for Agile Software Development*, 2001. [Online]. Available: http://agilemanifesto.org/iso/de/manifesto.html (accessed Sep. 08, 2020).

Christian Reidl was born in 1976 in Passau, Germany. He studied electrical engineering at the Friedrich-Alexander University of Erlangen-Nurnberg starting from 1998 focusing microelectronics. He finalized University 2003 as a degreed engineer. He accomplished a master of business administration in General Management in October 2017. Actually, he is working on a PhD thesis in International Management focusing the inclusion of agile processes and methodologies into the development processes in the product development of semiconductor products.

In 2003 he started as a design engineer at Infineon Technologies Austria to contribute at the development of integrated magnetic sensors. 2013 he changed into a management position and since them he leads design groups as a director for product development in Villach.

Daniel Valtiner was born in Villach, Austria, in 1986. He studied a Systems Engineering bachelor program at the Carinthia University of Applied Sciences. Further he finished a master degree in Information Technology, followed by an MBA in General Management at the University of Klagenfurt in 2017. Valtiner worked 9 years in refractory industries (RHI AG) followed by a further 10 years in semiconductor industry at Infineon Austria, where he filled different positions in R&D as well as manufacturing. He is currently working as a senior manager. Besides that, Valtiner gives different lectures at the Carinthia University of Applied Sciences and currently works on his PhD thesis in International Management.