4th WDK Workshop on Product Structuring
October 22-23, 1998
Delft University of Technology
Delft, The Netherlands

INDUSTRIAL PRACTICE WITH PRODUCT PLATFORMS IN USA

Per Elgård

Product platforms, architecture

Abstract

This paper is based on visits to 7 US companies. Based on the practices we observed at these visits, combined with findings from literature, it is explained what is understood by a product platform in an industrial context, how platform based product development differs from "one-of-a-kind" projects, and how this requires a different organization.

1. Introduction

Product platforms have been the subject of considerable interest both within academia and industry as a means for supporting product variety and increasing reuse of engineering knowledge. Most of the academic interest has focused on the product architecture, although the industrial experience is that the organizational impact of a platform based approach to product development is also important to consider.

2. Definition of platform

Platform based products are families or generations of products developed from a common product platform. Meyer & Lehnerd [5] define such a platform as a "set of subsystems and interfaces that form a common structure from which a stream of derivative products can be efficiently developed and produced".

The product platform is thus not a product in itself, but rather a foundation of product elements, technologies, knowledge, etc., that provides a "design readiness" from which multiple generations of a product can be developed faster and with more efficient design reuse than without the platform. To illustrate this, one company explained how the platform they were working on was expected to sustain 6 families, with a total of app. 40-50 products, over the next 5-7 years (Figure 1).

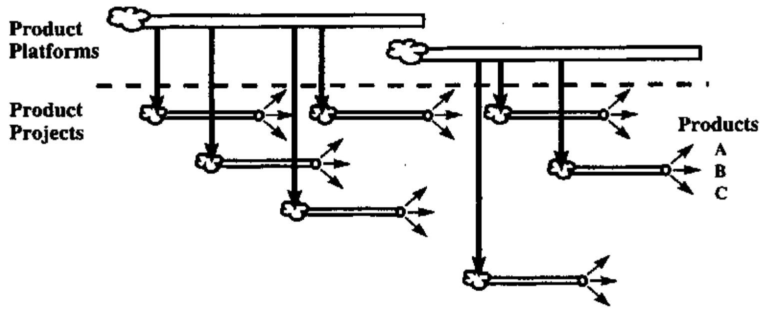

Figure 1: A platform provides a "design readiness" from which multiple generations of a product can be developed.

The design readiness is based on two main elements - the platform architecture and a range of reusable assets.

2.1. Architecture

Ulrich & Eppinger [7] define an architecture as the scheme by which a product's constitutive elements are organized, how functional elements are arranged, how functional elements are allocated to the physical units and how units interact. As Andreasen & Riitahuhta [1] further argue, architecture is more than the arbitrary structures of individually developed products, it is the result of a deliberate design process aimed at utilizing similarities among a range of products.

With reference to the work of Yu, Gonzales-Zugasti & Otto [8], a platform architecture can for example be based on the following three groups of architectures (Table 1):

1. Integral architecture - Single offer, where one product covers all needs with no variety. - Robust offer, where one product automatically recognizes and accommodates demanded variety.
2. Platform architecture - Modular family, where products share components and modules to meet market variety. - Modular generations, where product offerings share modular components in offerings that succeed each other in time. - Consumable platforms, where quickly consumed components are isolated and the isolated module is the platform. - Standards as platforms, where interfaces and format standards are platforms. - Adjustable for purchase platforms, where high variety sub-functions of a product are isolated into replaceable modules.
3. Massively customizable architectures - Fabricate to fit customization, where customers can special order the platform at the exact specifications desired. - Adjust for use customization, where features are provided so the customer can dynamically adjust the product to the demanded variety.

Table 1: Overview of platform architectures, adapted from [8].

Despite considerable academic interest in product architecture, then the definition of a platform architecture was the point where most companies had difficulties and where they saw very few tools and methods available to support the process. "It's like a dartboard game", as one jokingly described the process.

2.2. Reusable Assets

Many companies have tried to increase the commonality between different products by having reuse libraries in the form of preferred parts list, i.e. lists of commonly used parts, developed or introduced in earlier projects and made available for future projects, that designers are encouraged to use rather than introducing new parts. This can be advantageous even when the cost of a preferred part is higher than a new part optimized for a specific product, because one saves the fixed expenses connected with introducing new parts and suppliers. Although this type of general reuse makes perfect sense, then it is also the experience that the potential benefits are relatively modest and that significantly larger benefits can be obtained from a more domain specific reuse, understood as reusable assets based on an underlying architecture [2, 4].

Depending on the architecture, the platform can contain different types of domain specific reusable assets that can be used in more generations of a product. At the companies we visited, the reusable assets would typically be a physical subsystem or a larger software module, e.g. a chemical spreader for instant films, an ASIC for photo scanners, image processing software etc. It was not necessarily reused as-is, but could require some adaptations before it could be used.

A platform does not necessarily contain all elements of a product, but most companies considered it important for a product project to succeed, that all fundamental innovations were made before project launch. This puts the responsibility of investigating new technologies and inventing new ways to implement functions in the hands of the "platform team".

Technologies are rarely mentioned in literature on product platforms, but several of the companies mentioned technologies and innovations as an important part of the platform. At one of the companies, platform thinking was thus a lot more focused on the leverage of common technologies across divisional boundaries than in developing product families. A platform in this context was a core technology, applicable to products in more divisions, e.g. special materials and manufacturing processes.

3. Platforms in the organization

In most literature on product platforms, little attention has been given towards the organizational impacts of a platform based approach to product development. The most important contributions on this aspect seem to originate in software development, where e.g. Jandourek [3] and Karlsson [4] describe how platform development can range from leverage within and between product projects to a formalized architecture and design repository with a separation between platform teams and project teams. We saw different examples of this organizationally separation between platform and product development.

In some cases, the platform responsible would just be a separate group of people in the development department. On a higher level, one company had defined 5 corporate platforms, each managed by a platform director with his own people and linked to R&D. In this case the business units still define what the platform should consist of whereas the platform director defines how and when it is implemented.

The most remarkable separation between platform and product development was probably presented by a company with a 4-level approach (Figure 2). The 4 different levels were - Projects, Reference platforms, Architecture and Technologies.

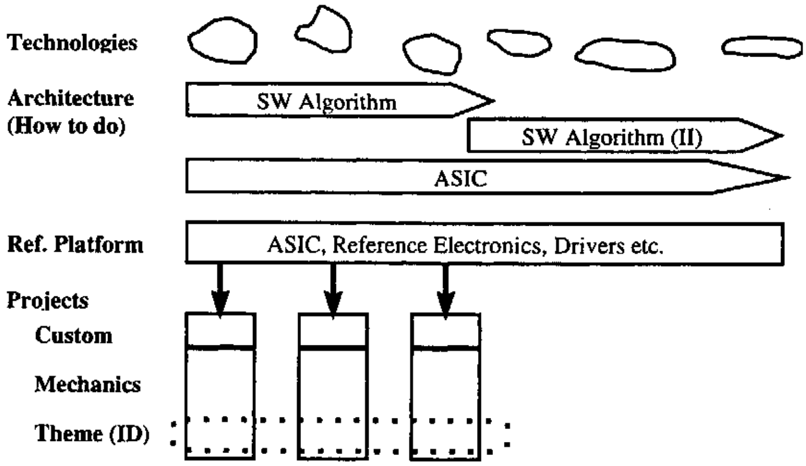

Figure 2: A 4-level platform approach.

Technology development is carried out independent of the rest of the process and mainly takes place at the component level, either at the company or at component suppliers. *Architectures* are abstract, more high level, and not very detailed descriptions of design patterns: e.g. "the philosophy of storage". *Reference platforms* are the solution shelves, where the product projects can pick the specific implementations for the new products, e.g. in the form of ASICs, electronics and software. Reference platforms are described in "Engineering Reference Specifications", detailed descriptions of what is and what is not included in the platform. To achieve a very fast product development cycle time, *product projects* would then build on whatever platform was available, although with some room for customization. If a specific solution was not available in time for implementation in a product design, the previous generation solution will normally be used. This may result in higher cost and less margin, but "the consequence of missing a market window is much more painful".

This need for product projects to adhere rather strict to the available platform was also mentioned by others. There is a tradeoff between the need to make a platform reusable and avoid constraints. It therefore requires discipline to adhere to the platform, but is necessary in order to avoid one-offs.

One way of improving product teams' ability to utilize a platform was to let some of the platform people participate in the product projects, not only to facilitate the adaptation of the platform but also to provide feedback from the downstream activities to the platform team.

In the company where the platform was based on shared technologies rather than physical assets, the organizational implementation was completely different from the before mentioned approach, with focus on networking rather than product architecture. We saw different means of supporting cross divisional use of these core technologies. First of all, the divisional research labs of the company were all located together on the same campus at the headquarter of the company. The interpersonal networks were further supported through internal technical forums and conferences, both at engineer and management levels. A last point was to give research labs names that actually described their activities, thus making it easier for other employees to identify them.

4. Phase plans

The different organizational implementations of product platforms were also reflected in the way phase plans were used to support platforms. It was typical that in the companies where platforms were implemented in the organization, they were also reflected in the phase plans.

Two different approaches were used. One was based on a traditional phase plan, but with a platform definition phase prior to the normal delivery process. The other described "defining the platform" and "developing products based on the platform" as separate activities.

At one company the latter approach was implemented as illustrated in Figure 3. Larger projects with an uncertain demand for variety would be based on a platform approach whereas smaller and simpler projects could be developed directly.

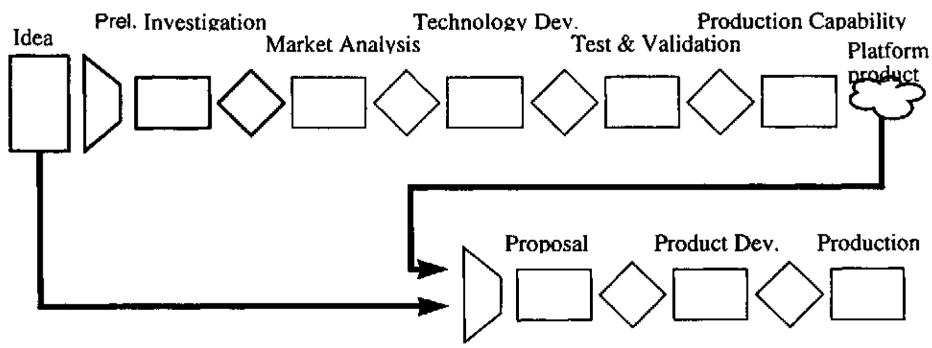

Figure 3: A 2-level phase plan.

In large projects, the platform phase in this company was used to develop and rigorously test a generic product, i.e. the platform. In the product projects, they could then adapt the generic product to specific requirements in response to orders, with a much lower risk and a significantly shorter development time than without the platform.

5. Evolution of a platform

Sanderson & Uzumeri [6] describe the technology life cycle of most products as being divided in two phases (Figure 4). The starting point for a life cycle is a technological discontinuity, a shift in paradigms, that renders existing technologies and competencies obsolete because of the appearance a new technology. Some recent examples are the shift from analogue records to compact discs and the shift from typewriters to computer based word processors.

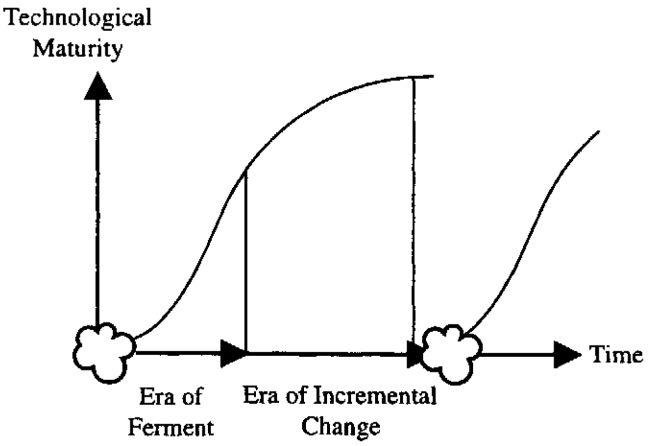

Figure 4 Technology Life Cycles. After [6].

The first period after such a shift, the *Era of Ferment*, is characterized by a large variety in solutions and significant technological progress while exploring different ways to apply the new technology. At the end of this period, competing technologies and solutions are progressively eliminated until only a few dominant designs remain.

These dominant designs provide the basis for the second period, the *Era of Incremental Design*, during which the design effort is focusing on optimization and incremental change rather than searching for technological breakthroughs. Sanderson & Uzumeri [6] give an example on this evolution of a dominant design in portable computers. The first portable computers entered the market around 1981 and during the next ten years the market saw a large variety in the physical design of these computers. But then around 1990 the notebook design (an A4 sized computer) appeared and today still completely dominates the market.

To accommodate this evolution, several of the visited companies noted that a successful platform should be dynamic and able to change in response to changes in customer requirements and technological evolutions. Especially technological changes were considered important for the evolution of a platform. Different technologies have different lifespan, and platforms can even include "not yet developed" technologies. In this case the architecture should be modularized around the object that will see the change in technology.

One could have expected that the inherent dynamic nature of a platform would also be reflected in an iterative type of phase plan, but the actual phase plans for platforms we saw appeared to be linear in the same way as for normal projects. Despite this apparent linearity of the phase plans, the platforms were actually dynamic in most of the cases, either through yearly platform revisions or in response to projects in a way described as a "chicken and egg game".

The most dynamic platform was in the company that had divided activities in projects, reference platforms, architecture and technologies. They used normal "linear" phase plans for projects, whereas the platform itself was not described in a phase plan but was rather continuously updated. The actual update was done by 3 – 4 people (including the chief technical officer) at a technical committee meeting held once a month and with only one part of the architecture to discuss at each meeting. This continuos update of different parts of the architecture was also reflected in the reference platform, where the company made a point of developing different elements of the platform out of sync with each other. In this way, they avoided having too many new and unproven solutions in the same product, thus reducing the risk of failures.

6. Conclusion

When comparing the practices at the visited companies with literature on product platforms, it can be concluded that there is good agreement on the definition of a platform as a design readiness, based on an architecture and a range of reusable assets, from which a stream of products can be developed. In practice, however, the definition of reusable assets seems to be broader than in most literature, as several companies also consider abstract assets, like technologies and innovations, as part of the platform.

In addition to the content of the platform, an important aspect of product platforms seems to be their practical implementation in an organization. Most of the visited companies have separated platform and product development organizationally, as well as they use different phase plans for platform and product development. An important reason for this separation is probably the inherent linear nature of product development versus a more dynamic nature of a platform, creating a need for a more iterative process.

The main benefits of a platform approach to product development are seen as a shortened product development cycle and a lower risk in projects, both leading to a better responsiveness.

The main challenge for the companies seems to be the architecture definition, where they see very few tools and methods available to support the process, especially the trade-off between the need to make a platform reusable and avoid constraints.

7. Acknowledgements

Several people have been involved in this work. First of all, I would like to thank the persons at the companies we visited for generously sharing their experiences on product platforms. I would also like to thank my two colleagues, Niels Gade and Frands Voss, with whom I made the visits and afterwards had several good discussions with, and the Royal Danish Consulate General in Los Angeles, Melissa Cefkin at IRL and Cathy Meyers at Danfoss Drives for their help in arranging the visits.

8. References

[1] Andreasen, M.M. & Riitahuhta, A. *"What is Modularization?"*, Konstruktionsdag 1997, DTU, pp. 1-12, 1997, (In Danish)

[2] Cornwell, P.C. *"HP Domain Analysis: Producing Useful Models for Reusable Software"*, Hewlett-Packard Journal, Article 5, pp. 1-10, August 1996.

[3] Jandourek, E. *"A Model for Platform Development"*, Hewlett-Packard Journal, pp. 1-18, August 1996.

[4] Karlsson, E.A. (Ed.) *"Software Reuse: A Holistic Approach"*, John Wiley & Sons, ISBN: 0471958190, 1995.

[5] Meyer, M. & Lehnerd, A., *"The Power of Product Platforms"*, The Free Press, 1997.

[6] Sanderson, S.W. & Uzumeri, M. *"The Innovation Imperative - Strategies for Managing Product Models and Families"*, Irwin, ISBN: 078631009X, 1997.

[7] Ulrich, K. & Eppinger, S. *"Product Design and Development"*, McGraw Hill, ISBN: 0070658110, 1995.

[8] Yu, J.S., Gonzales-Zugasti, J.P., Otto, K.N. *"Product Architecture Definition Based Upon Customer Demands"*, Proceedings of the 1998 ASME Design Theory and Methodology Conference, Atlanta, GA, 1998.

Per Elgård
Danfoss A/S
Building E33
DK-6430 Nordborg
Denmark
alt.
DTU
IKS
Building 358
DK-2800 Lyngby
Denmark
e-mail: elgaard@danfoss.com