Portfolio architecture improvements in project-centric organizations (Incl. business cases for architecture improvement)

White Paper Resulting from Architecture Forum Meeting

March 13-14, 2018, Stevens Institute of Technology, Hoboken, NJ, USA

Edited by:

Teun Hendriks, ESI

Gerrit Muller, USN-NISE and ESI

Eirik Hole, Stevens Institute of Technology

Input was provided by the following participants in the Architecture Forum:

Name	Organization
Leandre Adifon	Ingersoll Rand
Igor Gejdos	Roche Diabetes Care
Teun Hendriks	ESI
Eirik Hole	Stevens Institute
Kees Kooijman	Sioux Embedded Systems
Bjørn Victor Larsen	Kongsberg Defense
Jamie McCormack	Thermo Fischer Scientific
Peter Nacken	Canon Production Printing

Name	Organization
Jurgen Nicolai	Bosch Thermotechnik
Paul Romanello	Ingersoll Rand
Martin Simons	Daimler AG
Rolf Siegers	Raytheon
Marnix Tas	Sioux Embedded Systems
Hugo van Leeuwen	Thermo Fischer Scientific
Bart Verdaasdonk	Bosch Thermotechnik
Martin Verheijen	Thermo Fischer Scientific

Published, May 2020

1 Introduction

The strength of a project-centric organization is a strong customer focus. The weakness is a lack of synergy between projects. Depending on the organization, they may have "product architecture", "platform architectures", and architectures at the portfolio level that facilitate synergetic benefits across the organization. We use the term *portfolio* at the higher levels of the organization. *Projects* tend to be customer/problem specific and tactically oriented to deliver. *Programs* on the other hand address classes of customers or problems that have some sort of affinity. A portfolio is the collection of classes of problems and customers at a high strategic level.

The hypothesis is that architectures at the portfolio level will enable an increased potential of reuse of assets, as well as ease of integration. This allows us to move somewhat upward in the System-of-Systems (SoS) type hierarchy from virtual to collaborative to acknowledged to directed (see Figure 1).

Directed - The SoS is centrally managed

Acknowledged - The SoS has recognized objectives, and active cooperation between SoS and constituent systems

Collaborative - The constituent systems and stakeholders cooperate

Virtual - The SoS nature more or less emerge from the constituent systems

J. Dahmann and K. Baldwin. 2008. "Understanding the Current State of US Defense Systems of Systems and the Implications for Systems Engineering." IEEE Systems Conference 2008 in Montreal, 2008

Figure 1: Types of Systems of Systems

We also expect more benefits from synergies inherent in the Portfolio. For instance, we may factor out services to the “cloud” making them available to address a variety of problems and customers.

The question is, will moving towards the portfolio level actually achieve the expected benefits and synergies we expect or hope for? In practice, most architectural synergies at the portfolio level will emerge through deliberate improvements of today's program architectures.

A key question explored in this System Architecture Forum meeting is how architects can identify and make the business case for necessary architecture improvements - especially those that go beyond scope of individual projects and project funds?

The members of the architecting forum discussed this topic, using the following questions:

- What are typical projects, programs, and portfolios do you have in your organization?
- At which level does your organization actively (meaning explicit descriptions that are up-to-date, known, shared, and used) apply architectures?
- What improvements to those architectures are being realized in your organization, especially at the program or portfolio level?
- How did you as an architect identify, motivate, and sell program and/or portfolio architecture improvements that spans more than the immediate project boundaries, especially in project centric organizations?
- How did you construct the business case for those improvements, including increased synergy? How did you identify and sell necessary architecture improvements (especially beyond scope of individual projects and project funds)?
- What level of synergy across projects and programs would be needed to warrant a portfolio architecture at all?
- If your organization has a portfolio architecture:
    - What is the underlying business model?
    - What benefits do customers get from the portfolio architecture?

2 Architecting in the context of large organizations with product portfolios

Large organizations typically develop and sell many different types of products. One SAF member organization sells products ranging from household equipment such as coffee makers, shavers, and vacuum cleaners, to complex medical imaging systems. These products have increasing complexity; the line between software and firmware design gets blurred, the number of variants as the rate of change is increasing exponentially in many markets.

Architecture can help then to increase speed to market across products and product lines by promoting reuse, increasing synergy, with the aim to shorten the development and test time and effort. For systems with long lifetimes, architecture can improve the coexistence of legacy and new products in marketplace,

Nonetheless ambiguity exists when we include Portfolio and Architecture in one sentence. This could be because of the Product vs Project centricity in organizations. In many organizations, the Portfolio level is not clearly defined. What is even architecting at the portfolio level? How and when to seek synergies? What does the business case look like to warrant architecting at the portfolio level? To answer these questions, we look first to how organizations manage product and project portfolios across their business using portfolio management.

2.1 Portfolio management

Large organizations increasingly turn to Portfolio Management to manage their product innovation. Portfolio management is the decision-making process around which programs and projects should be executed, based on their alignment with the goals and objectives of the organization [Van Kessel et. al., 2015].

While project management is focused on delivering tangible results, portfolio management is focused on the strategic decision-making process around which programs and projects should be executed. This is to a large extent based on the alignment of the programs and project with the goals and objectives of the organization. Program management is the intermediate layer focused on the delivery of business benefits. The objectives for portfolio, program, and project management are summarized in Table 1.

Table 1: Definitions and Key objectives for (project) Portfolio, Program, and Project [Van Kessel et. al., 2015].

Level	Definition	Key objective
Portfolio	a group of programs and/or projects managed in a coordinated way to support business strategy and to deliver benefits in line with strategic objectives	Portfolio management is focused on doing the right things.
Program	a set of interrelated projects managed in a coordinated way to attain the business objectives and benefits	Program management is focused on realizing the benefits.
Project	a temporary endeavor to create a unique product, service or result	Project management is focused on doing the things right

Portfolio management aims to strengthen the strategic alignment of programs and projects to prevent initiatives being undertaken that do not support the enterprise strategy. It aims to enhance the overall economic value of the portfolio to improve the return on investment, i.e. on the tangible business benefits of programs and projects. Further, portfolio management aims to enhance executive decision-making on programs and projects based on company specific criteria [Van Kessel et. al., 2015].

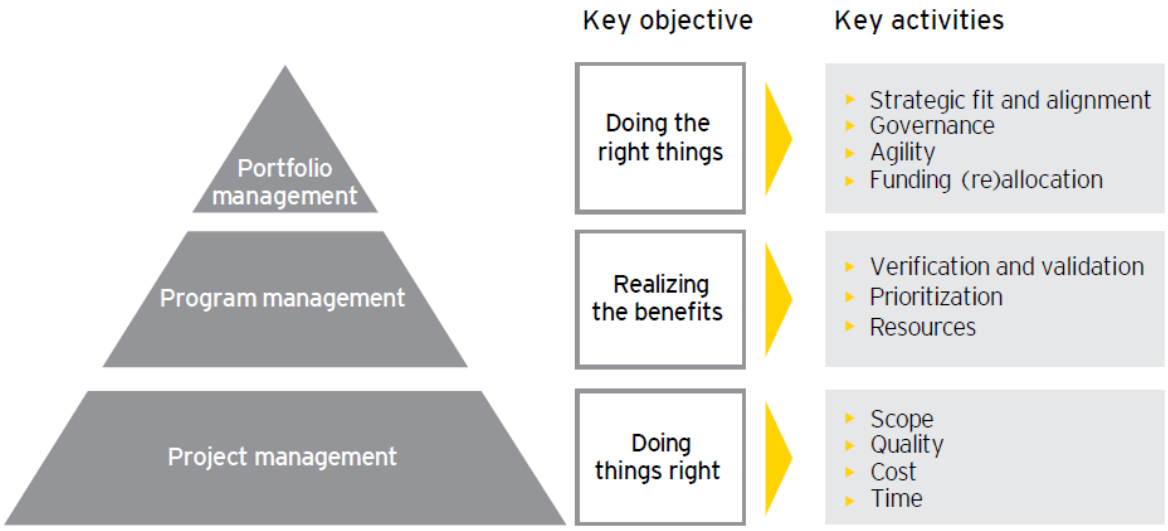

Figure 2: Portfolio, program and project management objectives and activities

At the portfolio level, organizations strive to align funding with strategic objectives. Programs coordinate and manage a set of interrelated projects to realise the targeted

business benefits. Projects will have to be scheduled based on a thorough evaluation and prioritization given the current capacity, to maximize value and ensure delivery, maintain, and monitor portfolio health and financial performance (see Figure 2). The key questions addressed by portfolio management are the following:

- Portfolio mix: Is the funding aligned with strategic objectives?
- Demand and capacity: Are the prioritization and sequencing of projects right, given current capacity?
- Portfolio value: To what extent did the portfolio so far achieve the organization's strategic objectives?
- Portfolio health: Are current programs/projects progressing as planned?
- Financial management: How effective are program /project budgets managed?

The appendix (page 20) lists sample metrics to measure such portfolio management aspects.

How does architecture fit with such portfolio management objectives? How can architecture contribute to the answers for these portfolio questions? To examine what role architecture can play at the portfolio level, the forum discussed a case for a Common Imaging Platform.

3 Case Study: a Common Imaging Platform at the product portfolio level

A SAF member presented a case around a common imaging platform. The organization produces several large industrial systems as product lines, where in each product line the imaging capability played a central role in the product functionality

The organization had three business lines, each supporting various product lines and product variations. All three markets/business lines moved to graphic arts. The overlap between the product lines became apparent, especially in the SW domain. This caused both duplication of effort as well as some confusion for the customers.

However, the organization was project-centric, with one project delivering one product (i.e. a product/market combination). Project size ranged from single digit person-year size to three-digit person-year size for a complete product overhaul. Projects were organized in a matrix with the disciplines co-located . The *project axis* had to deliver successful product in time, the *functional axis* had responsibility for technology, platforms, and synergy.

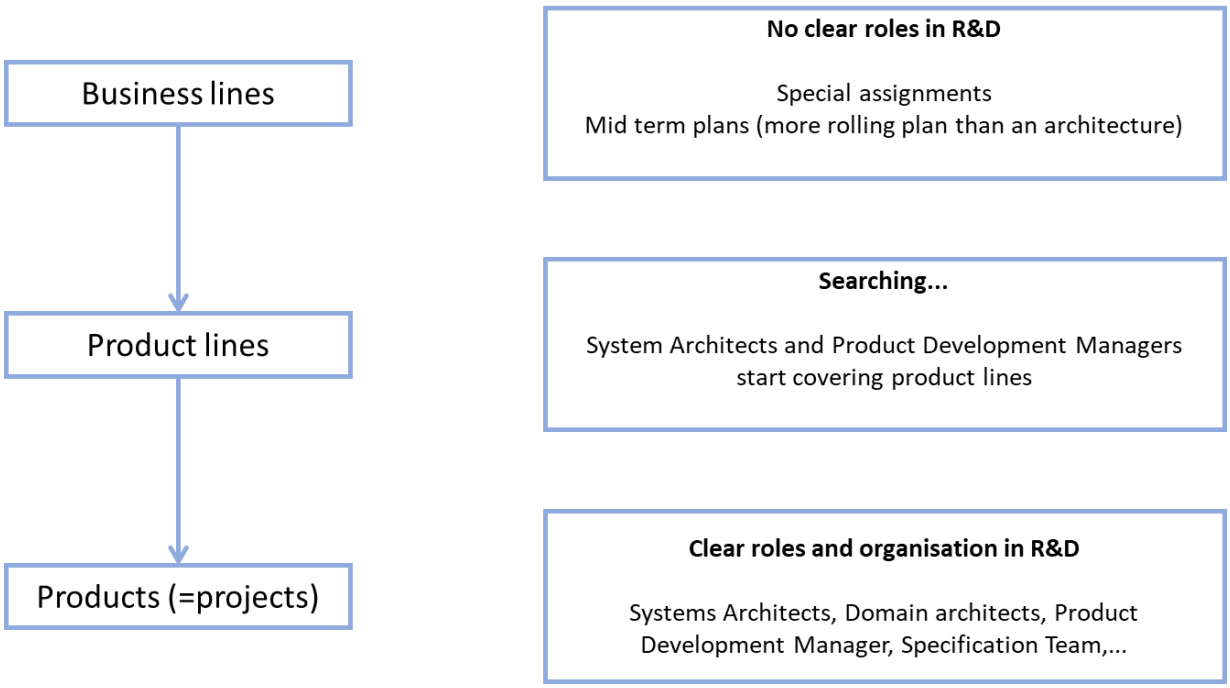

Figure 3: The structure of the organization and definition of roles

The project reporting structure went all the way up to the corporate board. On the other hand, the functional reporting on architecture synergy and portfolio development was more about influencing. The necessary functional roles were only partly defined in the organization (see Figure 3). At the project level they were defined, but at the level of the product/business line level they had only started to formalize functional roles. At the corporate level, no roles were defined; rather arranged through special assignments and mid-term plans.

The consequent power balance towards the project axis caused project objectives to dominate decisions. Thus, project timing, and cultural differences between the various business lines made each line develop its own solutions. The rationale for that varied from "maintain control over own destiny", need to maintain a certain "cost level" up to "we can do it quicker ourselves" supplemented by the "not-invented-here" syndrome. In this case, the architecture conceded to the organization.

Then, at one point, a top-down corporate decision was made to have an "one organization" approach to imaging across the business lines. This required the business lines to merge their imaging assets and achieve a common imaging platform as basis for its generic

development and evolution to serve the product line needs. This transition posed issues: the new product portfolio should be ready in four years, however until then the existing products would need upgrades as well. The “one organization” approach across business lines necessitated generic development however.

3.1 Generic development

Generic development of shared assets requires a shared understanding of commonalities and differences across business lines to achieve synergy. However, achieving *effective* synergy by means of generic development is also a complicating factor both organizational and technical [Muller, 2018]. Often, the generic assets rather become a liability than an enabler for customer value, e.g. when rapid change is made difficult by high change effort and cost with resulting complex configuration management to match all supported products.

Figure 4 shows drivers for Generic Developments and derived requirements for Shared Assets development, i.e. development beyond the scope of single projects and products. The first driver (Customer value) is extrovert: does the product have value for the customer and is the customer willing to buy the product? The second driver (Internal Benefits) is introvert, it is the normal economic constraint for a company.

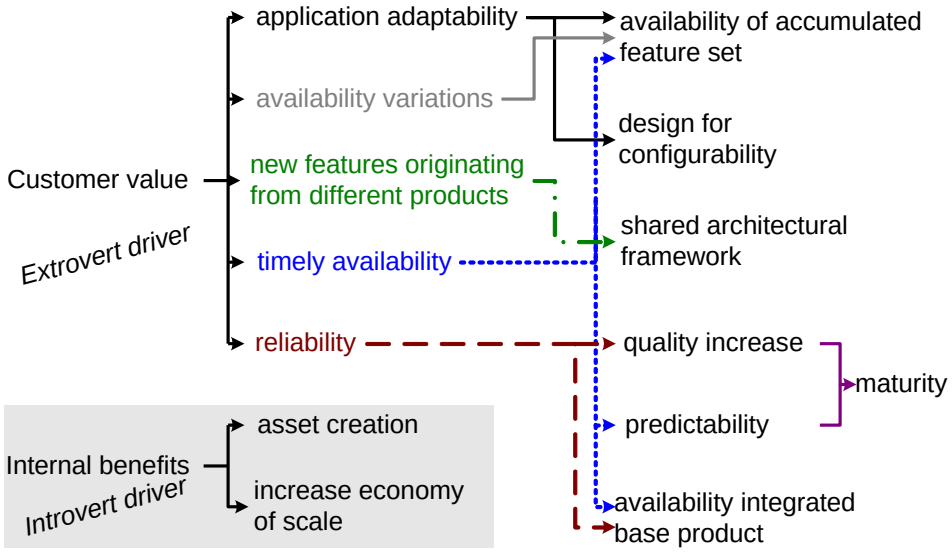

Figure 4: Drivers for generic development [Muller, 2018]

These drivers clarify that architecture at the portfolio level is only a means to facilitate synergy e.g. through the re-use of features and assets across product lines. Such synergy

only makes sense however if it does create value for the customer base, e.g. provides new features at less cost, or with less lead-time. So, a shared architecture framework at the portfolio level (a 'portfolio architecture') must simplify use of and re-use of features.

3.2 Generic development and re-use at the portfolio level

A transition to a common imaging platform firstly requires convergence of existing assets in the product lines. The main challenges are to overcome the 'not invented here', 'cost more and takes longer' arguments, and skepticism whether synergy is truly effective ("synergy makes it cheaper, right?"). Here, use comes before re-use. Re-use of existing modules and assets may help when people see that it makes sense and supports common business value.

Achieving re-use across product lines and platforms is more difficult; funding must be assured. A high-end product range may for example fund new developments to be re-used in the mid-end and low-end product ranges. Business lines may accept less margin on the mid-range product line to keep development for the low-range product line viable. Architectural alignment should not be enforced over product deadlines (this invariantly goes awry when project pressure increases with request for a "short term" solutions). Rather re-use module accommodation should be prepared through interface management and architectural preparation. Cross-licensing of predeveloped modules to recipient platforms provides incentives for the business line management to support development and re-use.

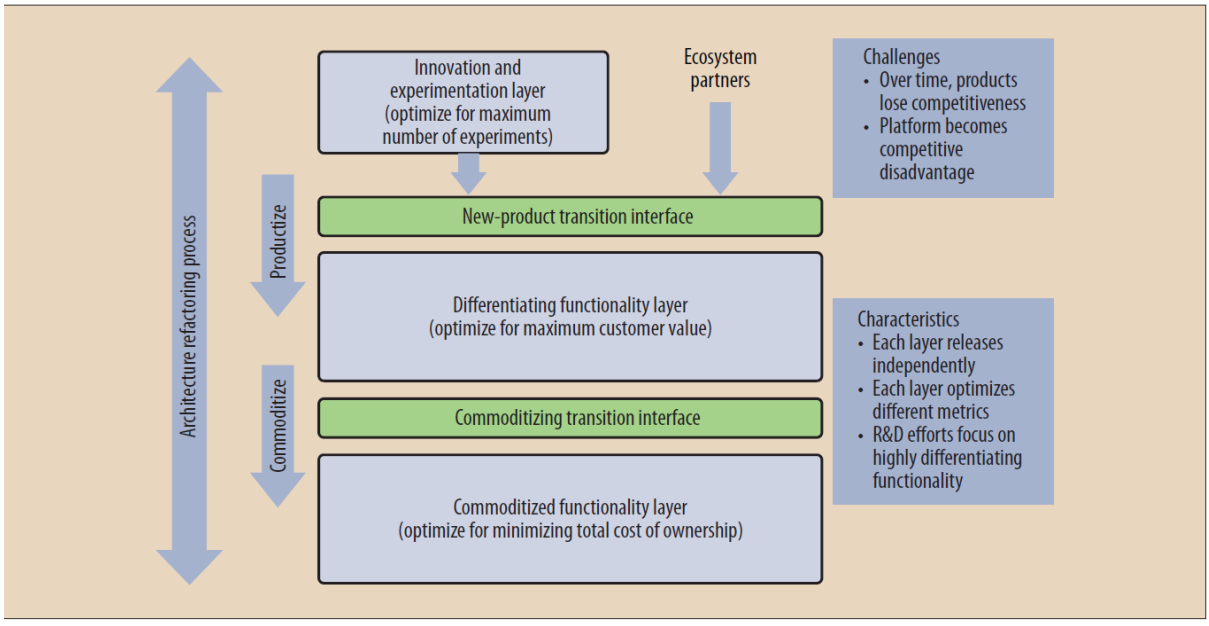

Figure 5: Layered functionality with different business objectives & development strategies [Bosch 2013]

An understanding of commonalities and differences across product lines is input for a development roadmap. Partial roadmaps can drive architectural improvements, the overall roadmap is a tool for portfolio management. A key aspect of a portfolio roadmap is the distinction of separate functionality layers/modules for: commoditized functionality ('base functionality'), differentiating functionality ('core functionality') and innovative / experimental functionality ('key functionality'), see also Figure 5. These functionalities are differentiated as follows [Bosch, 2013]:

- Commoditized ('base') functionality represents functionality that over time has become so integral to a system it no longer adds real value.
- differentiating ('core') functionality offers newer, more specialized advantages and clearly has customer value (
- Innovative / experimental ('key') functionality is functionality under various stages of development which does not currently add value but has potential to do so.

Key argument of Bosch [Bosch, 2013] is to keep these functionalities separate, and where possible to migrate prior 'core' functionality (that has become a commodity) to the commoditized layer as well, to apply a cost-minimizing development life-cycle strategy. In addition to facilitating the flow of functionality from the innovation and experimentation layer to the differentiating layer and from the differentiating layer to the commoditized layer, this requires a proactive refactoring of a system's architecture to ensure that its structure does not deteriorate over time.

With respect to product and product portfolio development strategy, most R&D investment should target the differentiating functionality. This means that the Commoditized ('base') functionality should be consolidated, and, where possible, replaced by COTS and open source alternatives, to minimize maintenance and effort. Doing this across product lines and targeting synergy provides for economy of scale.

3.3 The role of architecture at the portfolio level

However, what does a portfolio architecture look like? How do we communicate regarding 'portfolio' when the stakeholders have a more diverse background? At the system level, system architectures can be constructed and described. Corresponding architecture methods are available to describe platform architectures, including methods to use these to derive individual systems and their architectures.

At the portfolio level, no templates are available to describe portfolio architectures. At this portfolio level, SAF members indicated that they rather explore synergies across products and product architectures. So how do we achieve synergies, and how do we reap the benefits from them?

4 Synergy across products and architectures

In many organizations, synergy across products and architectures is used to speed up development and lower costs. When do such synergies emerge, and when are such synergies deliberately driven? Does it always make sense to pursue synergies? When are synergies applicable, and when not? The forum discussed these questions.

4.1 Emerging synergies

Synergies typically emerge when existing platforms are identified to be applicable to a different product. Then opportunities for reuse through synergy are the driving factor. Reuse in new or different products is the easy part of synergy. The merging back of innovations and adaptations in the original platform or products is the hard part. Synergy typically imposes additional cost to the original product or product line by the need for retesting of merged innovations. Often synergy can also cause configuration variability to suit slightly different requirements or contexts for the various products or product lines.

Emerging synergy thus needs to be managed. It is important to have visibility into which configurations synergy does add value. The difficulty is to identify and monitor the right metrics for the business case: when does a lower cost variant e.g. justify an increase in configurations ?

4.2 Business driven synergies

Markets and competitive pressures also may drive synergy: to improve economy of scale, or to achieve cost-down and efficiency improvement. In that case a business champion in the organization typically infuses the urgency, targets and funding. Ideally, this is championed at the management level just above the architecture level. Some of the typical drivers for top-down synergy are the following:

- Value proposition
- Shorter time to market
- Cost down through economy of scale and re-use of components
- Organizational efficiency/reuse of personnel and their competencies
- Development efficiency/reuse of technology
- Integration of organizations after mergers and acquisitions
- Regulations & standards (e.g. USB, Bluetooth)

The justification / motivation for synergy and common solutions characteristically centers around business concerns such as life-cycle management and integral cost considerations. A prototypical misconception is the intention to re-use a technology or component 'as-is'. Most often this soon transforms to 'use component A, but add feature B, and then C as well'.

Managing synergy can be done at the start of a project by identifying options for re-use or adaptation versus new design. In a running project, incorporating asset harvesting from the start can bring a focus on synergy, i.e.to include reuse investigations as part of a running project. At architecture level, reference architectures [Muller and Hole, 2007] support a wider re-use at corporate level across system projects.

Tools and checklists can provide support for synergy management. At the company of one SAF member, a checklist was in place with 30 questions on re-use, such as "have you exported it" (such that other divisions/projects can access/read in the artefacts). Configuration management tools provide visibility to feature variability and feature support, creating valuable input, for road mapping. Integral cost management tools can provide the data and visualization to sell synergy and re-use to decision makers. The re-use/synergy proposition must make assumptions explicit, with an argumentation what is needed to validate those assumptions.

4.3 When should you avoid synergies?

Synergies do not always make business sense. When you implement disrupting technologies, these will first need to be experimented with. Then they have to be consolidated until proven before attempting re-use (see section 3.2). When individual platforms are targeted for merging, each of them already have viable scale. At this point, striving for re-use may

cause products to lose cost effectiveness or market flexibility at the opposite ends of the market spectrum.

Targeted synergies may also have negative impact on product differentiation across product lines for e.g. a low-end range versus a high-end range. This may cause the low-end to become expensive, or may limit high-end range functionality. Tesla for example includes all HW needed for full-self driving operation in all its vehicles (and vehicle cost consequently) but requires vehicle owners to purchase SW licenses to unlock the "high-end" functionality. This can be both an advantage as disadvantage depending on competitive pricing of the base model versus license take-up rate.

4.4 What can you do as an architect?

Architects can facilitate discussions on synergy, but only if they understand the business, its customers, and the politics, internally and externally. System architects should avoid being dogmatic, but rather persuade, influence, and be transparent on the various possible architectures, and explain their respective pros and cons. In large organizations with several platforms it is wise to also have architects who look broader than single product lines and platforms. Such architects may have to operate at the corporate level to oversee the complete business

5 Scenarios for multi-product development with common technology

To illustrate how synergy may affect development and to which the business and technical concerns it may give rise, a number of scenarios were elaborated for development of two systems which potential for synergy and common technology. These scenarios are the following:

- **Common:** develop common solution first for both products, then sequentially integrate into the two products.
- **Phased:** develop and integrate per product independently.
- **Merge back:** develop as in phased scenario, but merge-back solution of product 2 into product 1 as common solution.
- **Architected:** develop products as in the merge-back scenario. At the same time, explicitly prepare the product architectures upfront to reduce later the re-integration time & effort at the end.

Figure 6 shows the respective development and sales timelines for these four scenarios. In each of the scenarios the top, gray line shows the non-recurring engineering (NRE) costs, the green lines show when which product variants can generate profit in the market.

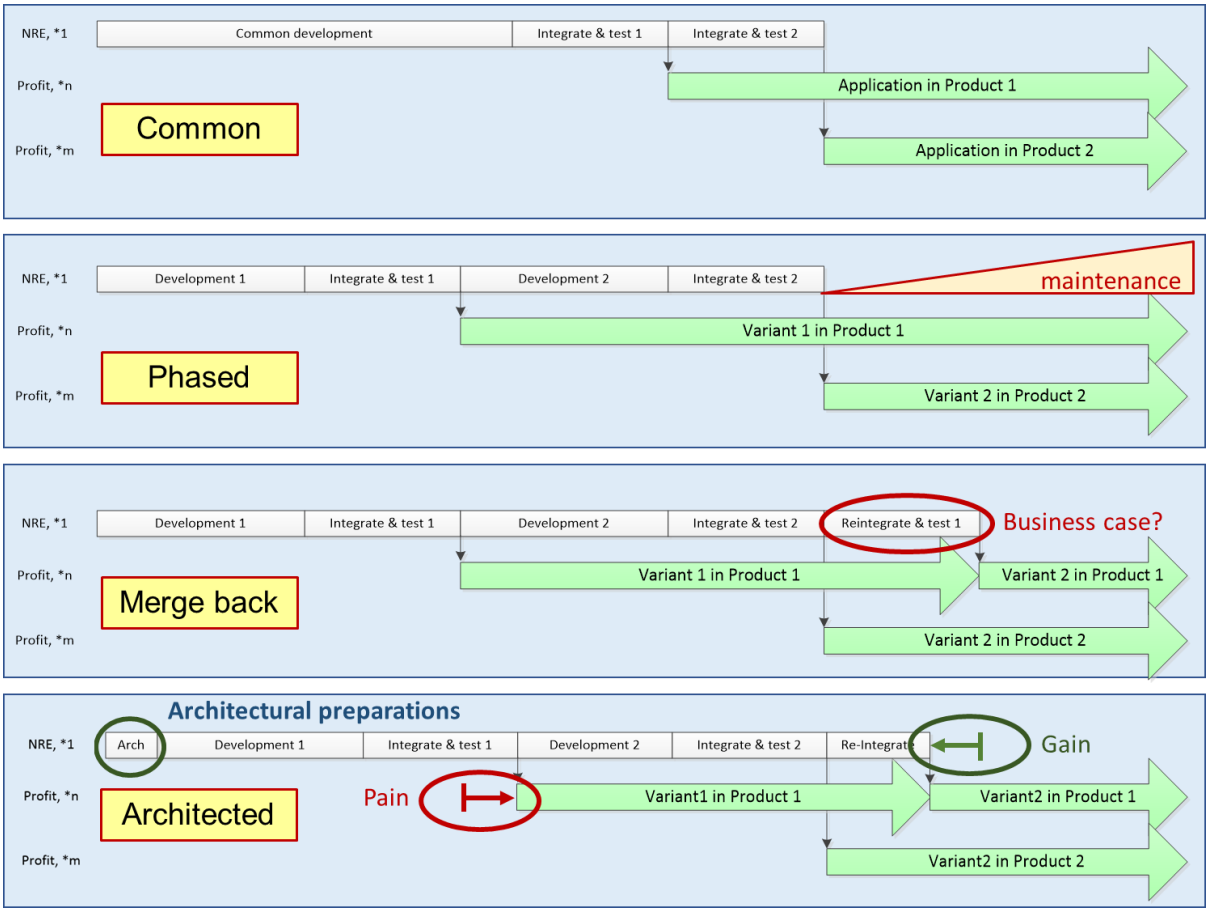

Figure 6: Four scenarios for multiple products development with common technology

Note that the scenarios in Figure 6 have a number of assumptions built-in, for instance, the common development in the first scenario for both products together has the same lead time as the sum of the separate development times of product 1 and product 2 in the other scenarios. Such assumptions should be clarified/validated. Notwithstanding, these scenarios provide a good basis for comparing business aspects.

The first scenario “Common” illustrates a common technical development then integration & test in various products. The key advantage is “develop once”; the evident disadvantage is a significantly larger time to market until the first product starts generating revenue. In competitive markets it is not only a larger lead time to revenue, but also potential loss of market share when competitors could release their products earlier.

The second scenario “Phased” shows an independent, phased, development of product 1 and product 2. The big advantage with respect to the “Common” scenario is the much earlier time to market for the first product. Product 2 follows later. The extra or improved technology developed for product 2 is never re-integrated into product 1 for a common platform/solution. The lingering extra cost in form of additional maintenance of two product variants occurs after release of both products (and their success in the market).

The third scenario “Merge back” aims to avoid the additional maintenance issue of the “Phased” scenario. After release of the 2nd product, the extra technology developed for that product is merged-back into the 1st product, in such a way that both products are on the same technology line again. This requires extra integration and test effort, for no additional functionality. This then occurs at a time when the first exposure of products to the market undoubtedly elicits various requests for new features and product improvements as well. Question is thus how to sell a business case for mere re-integration to product marketing.

The fourth scenario “Architected” then aims to perform architectural preparation for the commonality upfront such as to reduce or minimize the re-integration & test cost of common technology back into product 1. The “pain” of this approach is a little later market introduction of product 1, the “gain” is reduced or minimized re-integration time and effort after product 2 release.

The comparison shows that each scenario has pro’s and con’s. Common development has a lower and delayed Return on Investment. Phased development introduces branched solutions. The merge-back scenario incurs increased development cost and risks. Some of the costs may be not be obvious upfront, or downplayed, e.g. the additional maintenance cost of branched solutions in the phased development scenario. Each scenario may be a right choice, depending e.g. on market conditions, or (lack of) technology stability. Question is what weighs in more heavily. This brings the discussion what needs to be considered when deciding on which scenario to follow for a situation /state of a product portfolio, i.e. how to construct a business case for the architectural impact, and which measures to consider?

6 Business cases for architecture improvement in project centric organizations

6.1 What measures can go into a business case for architectural alternatives

Proposals for Architecture improvements should clarify their contribution to business objectives. In project centric organizations, architects unquestionably need a keen understanding of the strategic objectives and related (portfolio) KPIs for the business at the portfolio level (see also section 2). Business cases for architectural improvements should relate to these.

To build up a business case for architectural alternatives, the forum distinguished four categories of measures as follows:

- **Financial:** Revenue, ROI, product and NRE cost, lifetime cost, maintenance cost
- **Business:** Growth, capacity, innovation, complexity, workforce competence
- **Quality:** Product conformity, defect/returns, complaints, scraps, recalls
- **Agility:** Time-to-market, time-to-profitability, product release/update rate

The financial measures, such as ROI (Return On Investment) and NRE cost (Non-Recurring Engineering cost), are most straightforward to make tangible and quantified. However, to sustain an organization's competitiveness, the business, quality, and agility measure are equally important. From these measures, detailed KPIs should be derived followed the organization's general business strategy. Suppose e.g. an organization's overall business strategy would change to product-as-a-service. Then indeed the related objectives and measures could shift to weigh in more heavily the agility and quality categories (prioritizing e.g. active rejuvenation, preventive maintenance, easier configuration management).

A business case for architectural alternatives should indicate the impact on the four categories of measures of aspects. The impact of alternatives for e.g. system decomposition, product structure, feature mapping, platform (re)use, technology adoption (over time), release cycles, integration complexity should be clarified, and how these align with the business objectives. Value and ROI considerations should include the complete lifecycle and associated cost. Architects should be transparent about the assumptions made in the business case, and risks involved. Further considerations may include the fitness for future (with projection of future costs to extend products). This requires an organization to have solid roadmaps for both products and technologies.

Across product lines, business cases for architectural alternatives may advocate where to introduce new (integrating) technologies first, and how to weigh the benefits of cross-company standardization versus individual product cost price increase. Alternative scenarios considered were one with lower economy of scale, higher NRE & maintenance, versus higher cost price for the low-end product range.

For large decisions with major business impact, typically purchasing and cost planning departments are involved. Part of the business case may also be considerations to offset pains of a low-end product line with adjusted cost targets for unit pricing, compared to a company-wide benefit for increased economy of scale and lower variability across the board.

6.2 Building a business case for architectural improvement

Multiple perspectives are needed to build a business case for architectural improvement. This requires support from various stakeholders and subject matter experts throughout the organization.

To underpin a business case, historical data on existing architectures is needed. This includes life cycle costs, re-use of components across the portfolio, and identification of enablers or tools that are used across the portfolio. This should be contrasted with external trends and experiences in similar domains.

The business case should aim to identify the root causes of previous successes/failures on architectural improvement initiatives. For this, the input is needed from relevant stakeholders such as subject matter experts, module architects, product owners, stakeholders for requirements, project leads for estimations of development time, purchasing for providing bottom up product cost. Also, sales, product management, manufacturing, sourcing, aftermarket and service may provide input on the integral lifecycle cost and consideration, as further success and failure determinants.

The four scenarios on developing multiple products (see section 5) may form a basis to show the impact of architectural alternatives. The impact of architectural synergies and alternatives on development lead times, and integral lifecycle cost should be clarified. More generally, the impact of the four categories of measures can be illustrated by comparing two or more of the scenarios. Consideration regarding technologies and components across the product portfolio should be categorized with respect to the “base”, “core”, or “key” role (see section 3.2). These should be complemented and strengthened by scenario

analysis. It is especially important to balance sales and product management concerns for business agility and flexibility versus financial and business concerns.

The business case should be presented to 'the business' at the right level, i.e. project, product line, or portfolio, as appropriate for the respective organization. The benefits of synergy should be targeted to include aspects such as lower integral development cost, lower delivery time, increased quality, lower risks, increased economy of scale with improved logistics, and possibly lower amount of stock parts to keep etc. At the portfolio level synergy should at least target lower price per unit in the low end products, increased competitiveness, flexibility, and agility (with shorter time-to-market) to be prepared to address future market needs.

7 Conclusions

Project-centric organizations have a strong focus on specific customer needs. The solution is often tailor-made, following customer requirements to the letter. This leads to a frequent problem that underlying needs such as adaptations for further (international) markets, or potential for future product upgrades, are not jointly explored. Consequently, customer and supplier jump to solutions that are not explicitly anchored in the underlying needs.

Infusing architectural synergies at the portfolio level can create business benefits, but only if the underlying technology and market needs are mature and stable enough. Projects and product only profit from synergies if their applicability stays relevant over longer times, leading to lower integral development and lifecycle cost and larger economy of scale. When in contrast, markets and technology as still moving rapidly, then synergies may backfire, through reduced development agility to meet developing market demands.

A sign of good quality of architecture is stability over time. At portfolio level, stability means that all organisation's stakeholders along a product lifecycle must understand and support benefits from architectural synergies. Building a business case for synergies at the portfolio level hence often is a long process, with interaction of many stakeholders, to achieve alignment and support, precisely to create and maintain such stability.

8 Literature

Bosch, J., Achieving simplicity with the three-layer product model. Computer, 46(11), 34-39, 2013. DOI: 10.1109/MC.2013.295

Brown, J., Project Portfolio Management: Metrics that Work, Distant Production House University, 2011. https://www.dphu.org/uploads/attachements/books/books_5854_0.pdf

van Kessel, P., O'Leary, M., Polak, M., Portfolio management transformation, Ernst and Young, 2015. https://www.ey.com/Publication/vwLUAssets/Portfolio-management-transformation/$FILE/EY-PRM-Portfolio-management-transformation.pdf

Muller, G., Hole, E., Reference Architectures; Why, What and How, System Architecting Forum 2007 http://www.architectingforum.org/whitepapers/SAF_WhitePaper_2007_4.pdf

Muller, G., How to Create a Manageable Platform Architecture, Gaudi site, 2018 http://gaudisite.nl/HowtoManageablePlatformArchitecturePaper.pdf

Appendix: Metrics for project portfolio management

Portfolio management is focused on the decision-making process around which programs and projects should be executed, based on their alignment with the goals and objectives of the organization. Architects should be aware, as architecture improvements across projects and products will be judged against this decision-making process and strategic objectives.

At the portfolio level, organizations strive to align funding with strategic objectives. To do so, large organization maintain portfolio dashboards in which the relevant portfolio aspects are measured with metrics. In section 2.1, five categories of portfolio management aspects were distinguished: portfolio mix; demand and capacity; portfolio value; portfolio health; and financial management. The following five tables provide sample metrics for each of these portfolio management categories [Brown, 2011].

Table 2 Key question and metrics for the "Portfolio mix" portfolio management category

Portfolio mix	**Is the funding aligned to strategic objectives?**
Metrics	% of Portfolio spend "run the business"
	% of Portfolio in "grow the business"
	% of Portfolio in "innovate the business"
	% of Portfolio in Short/Medium/Long-term projects
	% of Portfolio in Large and Extra Large Projects

Table 3 Key question and metrics for the "Demand and Capacity" portfolio management category

Demand and capacity	**Do we have the right % growth prioritization and sequencing of projects given current capacity?**
Metrics	% growth in project intake
	% of growth in initiatives
	Resource utilization (human, material, capital)
	Recruiting pipeline
	Production capacity

Table 4 Key question and metrics for the "Value" portfolio management category

Value	**To what extent did the portfolio so far achieve the organization's strategic objectives?**
Metrics	% Portfolio projects on time
	% Portfolio projects on budget
	Portfolio and sub portfolio Internal Rate of Return (IRR)[1]
	$ saved for consolidation efforts

Table 5 Key question and metrics for the "Portfolio Health" portfolio management category

Portfolio Health	**Are current programs/projects progressing as planned?**
Metrics	Counts and amounts for programs and projects
	# of issues by severity

Table 6 Key question and metrics for the "Financial management" portfolio management category

Financial management	**How effective are program /project budgets managed?**
Metrics	% variance to plan
	% funding in-flight
	$ committed but not spent

[1] The Internal rate of return (IRR) is the interest rate at which the net present value of all the cash flows (both positive and negative) from a project or investment equal zero. Internal rate of return is typically used to evaluate the attractiveness of a project or investment.