Product line introduction in a multi-business line context.

An experience report

B.J.Pronk
Philips Medical Systems
E-mail: ben.pronk@philips.com
Telephone 00-31-40-2764123

1 Abstract

This paper describes the experiences with the introduction of a product line architecture in a multi business line environment within Philips Medical Systems. In this product line the reuse percentage is very high. The platform that implements the generic functionality covers a very substantial part (> 75%) of the functionality of all derived systems. For this project an architecture was created that allows reuse of a software platform through binary extension. Technically this architecture relies heavily on standard technology like Windows-NT and Microsoft's component object model (COM). Initially the product line introduction was organised according to the AFE/ASE/CSE models as described by Griss et al [4]. Furthermore a broad introduction was envisaged with the first release of the product line covering multiple products. Major organisational difficulties encountered with this model included: problematic decision structures in the multi-businessline context and subcritical ASE projects. The project was therefore finally reorganised to a reduced scope with less business lines involved. Our main conclusion is that for platform development in which such substantial parts are reused a single business line organisation seems the most adequate solution. Furthermore the Griss ASE/CSE organisational model is probably not very suited for this situation.

2 Introduction

2.1 The medical imaging market

Philips Medical Systems, a subsidiary of Philips Electronics, is one of the worlds main vendors of medical equipment. Its portfolio includes all types of medical imaging equipment like magnetic resonance imaging (MRI), computed tomography (CT) and conventional X-ray. The product range further includes patient monitoring equipment, medical IT-services and various other medical products like resurrection equipment. The medical imaging market itself is a mature market with increasing pressure on product development costs. The ever increasing software content of products, the extensive maintenance phase and the relatively low production numbers are the main drivers behind the introduction of product line architectures in this domain. This paper describes a recent introduction of a product line architecture spanning multiple business lines. The paper will start with a short overview of the products involved, the chosen architecture and the applied technology. The main focus however, will be on the managerial aspects of the introduction. What organisational structure and processes have been chosen and what are the experiences with these choices? How does this solution differ from the existing business line structure and how is the transform managed?

2.2 The products

Medical imaging products apply a variety of techniques to obtain images of the human body. The classical examples of imaging devices are X-ray, magnetic resonance imaging (MRI), and ultra sound equipment. Although these products are very different in hardware and physical techniques, they all share the same global structure which is depicted in the following figure:

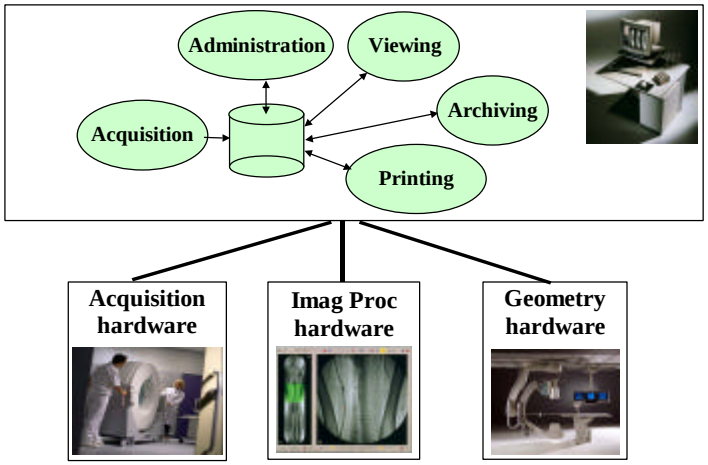

Figure 2-1: Medical Imaging Architecture

As depicted imaging equipment is usually constructed out of 4 main subsystems;

- A host processor that runs all application and user interface software and controls the peripheral devices that are needed to generate, process and view images.
- Acquisition hardware like the transducer of an ultra sound system that is used for the generation of medical images.
- Geometry, the mechanical hardware to support and position both the patient and the equipment.
- Image processor: Hardware and software that is used to process the images for optimal image quality and diagnostic capabilities.

The host computer typically runs a desktop operating system. The peripherals are locally controlled by embedded real time processors. Note that these are usually very large systems including a lot of dedicated hardware and millions of lines of code.

3 The product line architecture

3.1 Introduction, scope

A few years ago it was decided that a completely new product line was to be developed, which would replace a number of existing products. The choice for a product line architecture was driven by a number of factors:

- It was appreciated that the applications of these systems had become more and more similar over the years.
- Software development had become by far the dominant discipline in the development of these systems, software reuse was therefore key for future economic success.
- Technological advances made it possible to support the entire line with one sort of technology although in different hardware configurations.

The scope of the product line was chosen in such a way that reuse would be maximised. For this purpose a platform was developed containing all common hardware and software, which would be reused as basis for all the derived products. All components used in more than one product were included in the platform. As a result about 75% of the software was common over the various products. The variation was located in about 25% of the software and in numerous supported hardware configurations. The platform itself was reused as a binary component from which specific product line

members were derived by adding specific components to the platform. See for a more detailed description of this approach reference [1]. In the subsequent paragraphs an overview is given of the product line architecture and the technology used in its implementation.

3.2 Architecture

The product line architecture of the systems under discussion has been described extensively in various papers (see references [1]- [3]) and will therefore not be discussed in depth in this paper. It is a layered architecture with the following structure:

- A technical layer that abstracts from the specific medical devices that are used in the application. This technical layer serves as a virtual machine to build medical imaging applications upon.
- The application layer that implements the actual end-user functions using the technical services layer. The application layer interacts with the user interface through a model view controller pattern.
- The User Interface Layer, the presentation layer of the system that determines the presentation of data and controls to the user.
- Finally there is an infrastructure layer, that shields the operating system calls and offers basic support classes for licensing, logging and other infrastructure to be used by all software.

For a more detailed description of the architecture see references [1] and [3].

3.3 Technology

The system is built around a standard PC that runs Windows-NT and hosts all the application software (about $2*10^6$ lines of code, that is written in C++) and the user interface software. A passive PCI backplane with many slots forms the system control bus. The PC controls the standard PC-peripherals and a number of dedicated peripherals, of which the control is integrated in the main cabinet. These devices are controlled with an I960 Intel RISC processor running under the VxWorks operating system. Other peripherals are connected through CAN busses and Ethernet. Other important aspects of this set-up include.

- Use of standard middleware solutions (DCOM) for all interfacing.
- Use of standard software packages (database, license management, network) where possible.

For a more detailed description of the applied technology see reference [1].

4 Management aspects

4.1 Set-up

4.1.1 Introduction, organisational context

This paragraph describes the organisational context of the project where the following paragraphs will deal with the managerial aspects of the development of the product line architecture. This last part of the paper has been organised as a sort of historical report and does not only describe the final situation. It also describes the modifications made in the processes and the organisation during the development of the product line architecture and lists the main issues and problems encountered.
The products as described here were developed before in completely independent business lines. Independent in the medical context means that they have their own marketing and development departments, their own factory and above all their own financial responsibility. Note that these business lines are also geographically distributed over multiple sites in different countries.
Furthermore it was decided that a number of products were to be introduced simultaneously based on the first release of the product line architecture. It should also be noted that there is no tradition or history of platform development within these business lines. Up till now all products were developed

as really separate entities with their own architecture and implementation. The only reused practised on a larger scale was that of hardware components, that were developed in a "component" business line.

4.1.2 Processes

The processes followed to develop the product line under discussion were originally based upon a approach as described by Griss et al. [4]. In this approach three main activities are distinguished:

- An Application Family Engineering activity (AFE) that defines the requirements, the architecture, and the variation points for the platform. The architects that executed this activity had full responsibility for the (technical) set-up of the platform, definition of the scope of the platform (so what is generic what is specific in the range of products that have to be covered) and the technology choices.
- A CSE (component Systems Engineering) process that is responsible for the implementation and test of the platform hardware and software. The CSE group also integrates peripherals and software from many subcontractors into the platform. The CSE group delivers a tested and integrated software platform and a number of hardware components that can be ordered. This deliverables are not supplied directly to the market but to ASE projects.
- Finally there is an Application Systems Engineering activity (ASE) in which, the system group derives a product from the platform and specific hardware and software components are developed. This is also the activity that brings products to the market.

4.1.3 Distribution over organisational entities, project organisation

Historically the products that would be replaced by the results from the new product line architecture were distributed over multiple business lines. Therefore it did not seem logical to allocate the AFE and CSE activities that had to produce the platform to one of these groups because of the possible conflicting interests in timing and priority. Furthermore this would probably jeopardize the interests (and with that the support) of the other business lines involved. Therefore the entire AFE and CSE activities were allocated to an independent business line that did not have direct market interests itself. This business line was historically occupied with the development of hardware oriented components for the various business lines. For this purpose a significant transfer of development budget from the "product" business lines to the "component" business line took place.

Because the entire product had to be developed from scratch anyhow, the AFE and CSE activities were run for the first release of the platform as a single project. This was clearly not intended to be the final situation since the scope of the AFE activities spans multiple projects. Yet for the first release it appeared to be sufficient. The architects of the AFE/CSE project defined the entire product line architecture and the division between generic (CSE/platform) and ASE (specific) components. The AFE/CSE projects defined the entire context in which the derivation projects had to operate. The ASE-projects were set-up as separate projects run by the various "product" business lines. These ASE-projects developed the system specific items and did the integration and test of the final products. They were also responsible for the marketing, introduction and production of the final products. For these projects the platform was a "supplier", that delivered a tested platform with all generic functionality included. Note that the ASE's did not have architectural responsibility themselves, they were fully bounded by the architecture as laid down by the AFE-architects. This was reflected also in e.g. the design documents of ASE projects that were all produced as "delta" documents. In these documents was described how they used (configured) the platform and which specific components they added to the defined variation points of the platform. Note also that the ASE projects were considerably smaller (<40 FTE's) than the CSE/AFE project (a few hundred FTE's). It was planned that there would be a yearly release of the platform. Each platform release of which an increasing number of specific application types (derived systems) could be derived. This approach is depicted in the following picture:

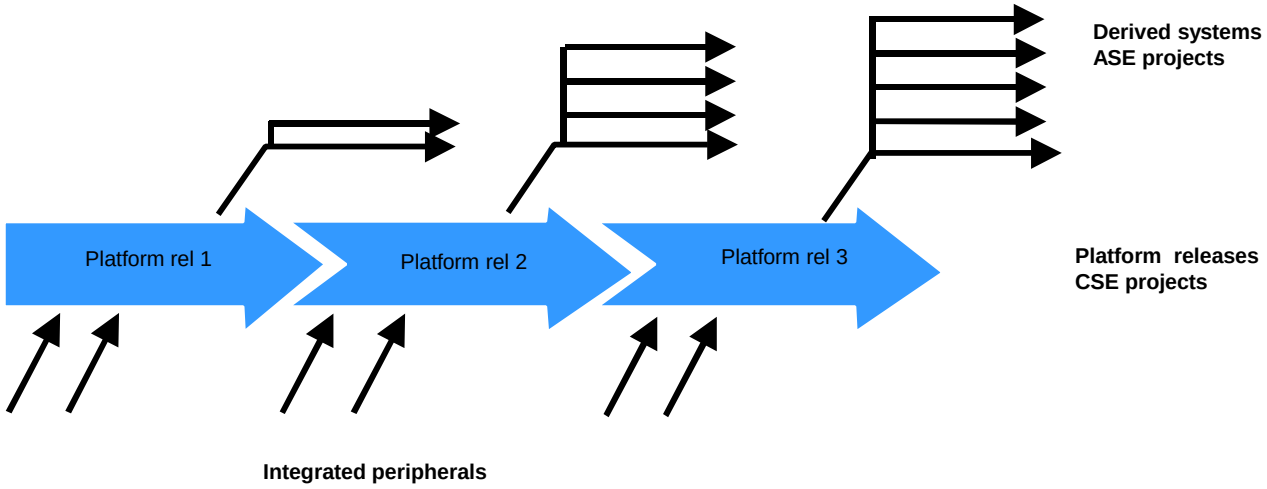

Figure 4-1: Product line organization

4.1.4 Staffing Aspects

The AFE/CSE project which developed the platform was a huge project (several hundreds of developers) that had to develop both a large set of hardware components and millions of lines of code, with the use of many new (at least for this organisation) techniques. Furthermore extensive application knowledge was needed because the core part of the systems was now to be incorporated in the platform. To cope with the increasing demand for software engineers with new technological skills, a large hiring and training program was started. Unfortunately the project coincided with a major economic upturn and a corresponding high demand for software personnel in general. It appeared impossible to hire sufficient personnel and therefore a substantial part of the project (50%) was staffed with subcontractors. This situation resulted in a very high turnaround rate of (especially software) engineers within the project. At worst the average dwell time of a software engineer in the project was as low 18 months. It is clear that this situation negatively influenced the productivity. Note that nowadays with less optimistic economic outlooks the stability of the staff has greatly improved, the amount of subcontractors is diminishing and the average dwell time is growing at a high rate. Application knowledge was brought into the project by moving a number of senior system designers and developers from the "product" business lines to the "component" line. Apart from adding application knowledge to the project this approach had another advantage. By moving the most influential and knowledgeable persons, in many aspects the technical opinion leaders to the platform group a lot of "resistance" was avoided. Yet it had one negative side effect. It made it very difficult to motivate crew to work on maintenance and development of the "old" systems anymore. To much the impression was raised that all the "best" people and the interesting work would be in the platform project.

4.2 Main issues, changes and problems

4.2.1 Organisational changes

The project(s) that developed the new product line architecture and components did not operate in a fully stable environment. As usual both management insights and market circumstances evolved during the project. Over time, this lead to two major changes that disturbed the project considerably. After about one year management decided to extend the scope of the product line towards low end products and other application areas far beyond the initial intentions. Apart from the required technical change with main issues in the cost engineering area, this also increased the organisational complexity

of the project. The number of business lines involved in the project was extended with two more. Moreover the geographical distribution of the project was extended to multiple countries. This greatly increased communication and travel overhead. Furthermore it was impossible to repeat the action of moving all "important" people to the component group. Another effect was that the "new" business lines had to join a project already under way for over a year with many of the important technological decisions already taken. This lead to considerable acceptation problems and a strong drive to reopen all discussions, with consequent delays in the project. Some time later the project was again shaken by a major change in direction driven this time by strong market pressures. Now the introduction of a system at the other end of the spectrum needed very high priority. The inevitable large delay in project completion finally lead to a redefinition of the entire program. The scope of the product line architecture was again reduced to almost its initial width. Furthermore the introduction of systems based on the platform was scheduled in a much more gradual way. In the first release of the platform only the highest priority system was introduced. The following release extended the number of products supported with only a few and so on until in two or three steps the full product range will be reached. Another major change was the removal of the difference between ASE and CSE projects. In actual project execution the overhead of project-project communication, the dependencies and the frequent discussions about responsibilities appeared to be to high. Furthermore because of the dominant size of the CSE activities, the ASE activities became subcritical in staffing. Another effect was that testing of the platform, which covered > 75% of the total system, was only possible by using an application as vehicle. The combination of these issues led to a major change of organisational approach in which the original model was abandoned. Nowadays the entire program is run as one (multi business line) project. Each release of the product line architecture project is responsible for the entire chain until release of products. For every release of the product line architecture the supported number of products is increased. This new approach now appears successful where the first releases of the platform have been completed with an increasing support for actual applications.

4.2.2 Decision structure

From the beginning, one of the key problems in the project was its decision structure. Historically development projects interacted with their own marketing and sales group in the definition and planning of projects. Conflicts and problems could usually be solved easily since both departments shared a mutual understanding of budget and staff restrictions and the market situation. Now this situation had become much more complex, since the project had to deal with several customers. The initial solution to this was to work with teams like a marketing team that included representatives of all involved business lines. This worked fine as long as the differences in interests between the business lines were small. Whenever this was not the case the only way out was escalation to much higher levels. Where in the single business line case marketing and development usually came to an agreement without any higher management interference this appeared to be almost impossible in the multi business line case. This appears to be caused by two major points:

- Business line tried to get the most out of the platform effort it at the expense of the other lines.
- Since there was no real financial feedback, the component group had a fixed budget and operated as a cost centre, there were no restrictions in the issuing of requirements

The root cause of all these troubles appears to be the absence of a good decision model for the platform activities. The simple customer-supplier relationship breaks down because there is not really an open market. There is only one producer and a few consumers which do not have any alternative sources. Several solutions have been exercised to improve this situation. The most successful was the use of champions, appointed authorised decision makers within each team. This worked reasonably well in areas were widely (over multiple business lines) respected and knowledgeable people were available. This brought down the number of "escalation" points considerably. Yet in case of strongly conflicting interests between business lines or in the absence of such a person only strong higher management force and attention is still the only solution.

5 Conclusions

It is of course very difficult to generalise the experiences of only one case even if it is of this size (a project of more than 1000 man-year). Furthermore many of the conclusions may seem obvious. In retrospect it is hardly surprising that changing the direction of such a large project several times introduces significant delays. Furthermore managing projects of this size is difficult anyhow and subject to delays even if not concerned with product line introduction. Maybe the staffing approach may be considered interesting to others in the process of introducing product line architectures. However the project did finally manage to successfully introduce the products based on the product line architecture. So there must be few lessons in this approach for product line architecture introduction in a multi business line environment in general. Note that I assume that the conclusions are only valid for a comparable situation in which a very significant (>50%) part of the system (in this case it is even > 75%) is generic. Our conclusions fall apart in two groups

- The first trivial conclusion from our experience is that the organisational aspects are much more complex than the technological and architectural issues in a multi business line environment. This conclusion has major consequences for the scope of a product line. Organisational complexity is strongly influenced by the number of business lines involved in the effort. Much more than technical complexity this should therefore be a consideration in defining the scope of the product line architecture. The creation of a business and decision model, that defines how the various interests will be judged and how a final decision will be reached on aspects such as specification, architecture, planning needs to be defined up front. Although we have gained considerable progress with the "champion" model we did not find a really satisfactory solution for this problem. One might therefore consider to reorganise the business lines up front to another organisational form in which a simple decision model is guaranteed, e.g. by combining business lines that have to work with the same platform.
- If the product line covers very substantial parts of the products inevitably the responsibility for architecture, development and integration shifts to a central platform group. This has significant impact on the organisation and staffing of the involved business lines. Much smaller development departments are now needed. The actual responsibility will shift from basic development to application integration and testing. From the experiences in this product line effort it appears that the ASE/CSE model as proposed by Griss et al [4] is not very suitable as an organisation form for this situation. A single project approach in which each subsequent project is responsible for "keeping all existing products running" seems to work much better.

6 References

1. B.J.Pronk An Interface Based Platform Approach, in Proceedings of the First Software Product Lines Conference (SPLC1) August 28-31 2000, Denver, Colorado, Kluwer Academic Publishers
2. J.G.Wijnstra Supporting diversity with Component Frameworks as Architectural Elements, Proceedings of the International Conference on Software Engineering (ICSE2000) ACM press 2000
3. J.G. Wijnstra Component Frameworks for a Medical Imaging Product Family, Proceedings of the Third International Workshop on Software Architectures for Product Families- IWSAPF-3, Las Palmas de Gran Canaria, march 15-17 2000
4. I. Jacobson, M. Griss, P. Jonsson, Software Reuse- Architecture, Process and Organization for Business Success, Addison Wesley, New York, 1997.