Software Product Lines structuring based upon Market Demands

Montse Ereño
Computer Science Department
University of Mondragon
Loramendi 4, 20500,
Mondragon, Spain
(34) 943 79 47 00
mereno@eps.mondragon.edu

Uxue Landa
Computer Science department
University of Mondragon
Loramendi 4, 20500,
Mondragon, Spain
(34) 943 79 47 00
uxuelanda@gmail.com

Dra. Rebeca Cortazar
Computer Science Department
University of Deusto
Avda. de las Universidades, 24.
48007 Bilbao, Spain
(34) 944 139 000
cortazar@eside.deusto.es

ABSTRACT
Nowadays the increasing demand for customized products and services in traditional areas such as Automotion Manufacturing or Aeronautical Component Engineering is being satisfied with a new approach called "Product Platform". This successful approach is also being considered in the design of software-based components in these areas, which are recognized as complex and critical.

In this paper, we present the research that is being carried out at Mondragon University. This effort focuses on the analysis of existing Product Platform Development methods and the transference of this know-how to Software Product Development.

As a result, a Software Product Line (SPL) development method will be defined and applied in a real case. This method will be based upon market demands, so it should be flexible enough to respond to customer's requests and market pressure. In this paper we will explain in detail one step of the process. This step is concerned with how QFD technique can be used to the specification of components in a SPL.

Keywords
Software Product Lines, Mass Customization, Market Perspectives, Requirements analysis and specification, Viewpoint-Oriented Requirements, Components.

1. INTRODUCTION
Today's highly competitive, global marketplace is redefining the way companies do business. If we focus on the Software Development area, we can see that during the last few years the Software producer's interests have been in contradiction with the customer's interests. Indeed, producers want to maximize their benefits and, thus, to minimize their production's costs and time to market. In opposition, customers ask for better quality and for Software tailored to their individual needs. Furthermore, complexity and size of software products are rapidly increasing due to the market's evolution.

This situation is not new, as the scenario is well known in other sectors such as Automotion Manufacturing or Aeronautical Component Engineering. In these sectors, the development of a family of products—a group of related products derived from a common Product Platform—has provided an efficient and effective mean to implement a sufficient product variety to satisfy a range of customer demands.

Software community has tried to adopt this new approach of product development. They called it "Software Product Lines " (SPL). A SPL is a set of intensive systems of software that share a set of common characteristics, that satisfy the specific needs of a segment of a particular market and that are developed from a set of common assets in a pre-established way [1].SPL represent an innovative and growing concept in Software Engineering. It can also efficiently satisfy the current demand for mass customization of software.

Despite the fact that SPL have recently gained research interest, there are only few empirical studies on them. We would expect a rush by the Software industry to exploit the competitive advantages offered by SPL. However, most of the software industry is either unaware of the emerging field of SPL, or if an organization is aware, they don't understand how SPL might be applied in their situation. There is a *need of process* and *quantitative models* to help enterprises in this new way of product development.

The purpose of our research in Mondragon University is the analysis of existing Product Platform Development methods and the transference of this knowledge to Software Product Development. There is an especial interest in product specification and modularization. The Governing Body of Guipuzkoa (Spain) takes part in this research.

The rest of the extended abstract is organized as follows. In section 2, we present the previous work in this field. In Section 3, we illustrate our advances. In this section, the core part of our contribution is given. Finally, in section 4, the paper ends with a conclusion and some words on the future work.

2. PREVIOUS WORK
From a mechanical perspective or taking into consideration products of tangible structure, we can affirm that this issue has been studied widely. During the decade of the 90's, multiple studies and investigations appeared that were trying to tackle

these difficulties: how to obtain a great variety of products with a unique design that combines the greatest number of possible similarities [2]. The solution to gain this mass customization passes through the concept of product platform [3]. That is, a unique design for a platform can be personalized in such a way that can extend the variety of different products. We can mention the automation sector [4], aeronautical sector [5], aerospace sector [6] or companies like Hewlett-Packard or Black&Decker[7]]. In all, the challenge is in how to define this platform.

From a Computer Science perspective or intangible structure Product Development, considerable efforts have been made, too. Its origins can go back to the 60's. In a conference titled "Mass Produced Software Components" held in 1968, [8] introduced the *Reusability* concept as the key for the efficient design of new Software Products. The efforts made in the definition of methodologies for the development of Product Lines are considerable: PuLSE [9], KobrA [10], COPA[11], FAST [12], FORM [13], SPLIT [14], etc. From an application point of view, the SPL approach is being adopted by organizations of different sectors [15] [16][17].

As much from the theoretical point of view as form the application one, Product Lines are an interesting and promising approach. Therefore, we can appreciate how very different industries converge towards a new way of design when undertaking a product family development: SPL. Nevertheless, the software development sector is not as advanced as the mechanical industry. Most of the real cases of SPL creation are based upon theoretical methods and a great amount of intuition. There is a need of sound processes and quantitative models to help enterprises in their new way towards Product Development. This is the main motivation for this research.

3. SPL development with HOQ

This research analyzes the ability to apply a design methodology for mechanical products in the design of SPL. The base methodology was defined in a project realized in 2002 by the Design Group of Mondragon University with the cooperation of the Governing Body of Guipuzkoa (Spain)

During our research, we are focusing our effort in two main activities. On the one hand, we have been studying the process itself and the suitability to apply it in Software Manufacturing. On the other hand, we have been analyzing the importance of the Voice of the Customers for the analysis and specification of the SPL. During the rest of the paper, we will explain in depth the second subject.

3.1 Requirements Engineering for SPL: Getting the Voice of the Customer

In the context of SPL, requirements analysis and specification defines the system to be developed and forms the basis for a contract between a system provider and a customer. There are a number of inherent difficulties in this process. As [18], [19] [20] said, the hardest single part of building a software system is deciding precisely what to build. No other part of the conceptual work is as difficult as establish the detailed technical requirements. No other part of the work so cripples the resulting system if done wrong. No other part is as difficult to rectify later.

It's known by all the Software community that the most critical dimension in Product Development nowadays -and even more in the future- is to develop the products your customers want. During the product development, the Voice of the Customer (VOC) must be accurately defined. Without an accurate and complete definition of what a customer desires, the rest of the process is irrelevant.

In Product Development context, requirements specification is used to formalize and communicate the needs of a real or hypothetical customer to product developers. This is the forward flow of information from concept development to design and implementation. In our research we propose the use of QFD (Quality Functional Deployment) to deploy the Voice of the Customer throughout the product's design. Specifically, we apply the first part of QFD: the House of Quality (HOQ).

3.1.1 QFD in SPL environment

QFD is a structured approach for defining customer needs or requirements and translating them into specific plans to produce products that meet those needs. QFD can be considered as a tool for requirements analysis and specification. QFD answer the following questions:

- What are the features the customers/market desires?
- What functions must the product serve, and what functions must we use to provide the product or service?
- Based on our available resources, how can we best provide what our customer wants?

QFD is intended to be used by multidisciplinary or cross-functional teams (Marketing, Design Engineering, Finance, Quality Assurance, Manufacturing, Test Engineering, Product Support, etc). This is, of course, to add variety, varying perspectives, and hopefully provide more insight into issues that a single function team would eliminate or not consider relevant.

All these reasons, allow us to believe that QFD will support the different perspectives of the multiple customers, so we will use it to support the analysis and specification of the Product Line Requirements and to obtain the suitable Product Lines Structure based upon stakeholders demands.

3.2 Building the HOQ for SPL

The initial step of QFD is determining the voice of the customer (VOC). There is no one monolithic VOC. Customer voices are diverse. All these voices must be considered to develop a successful product. This is accomplished through extensive market research, or other more direct communication methods such as surveys, product complaint history, direct customer feedback, etc. The goal is to find the exact desires of the intended target group and design to provide these aspects, but still remain cost effective for the manufacturer as well as the customer. This

understanding of the customer needs is then summarized in a Product Planning Matrix, or House of Quality (HOQ). We use these matrices to translate higher level "what´s" or needs into lower levels "hows" product requirements or technical characteristics to satisfy these needs.

In software development different roles are found (customers, developers, designers, etc.), where each one provides different needs, with different abstractions levels and with their own perspectives. We have defined four actors. Each actor has his own perspective of the product.

- **Customer or Market**: this group represents all the possible clients around the world.
- **Enterprise**: this group is conformed for all the roles into an enterprise, who are involved on the development and commercialization of the product (Marketing, Design Engineering, Quality Assurance, Manufacturing, Test Engineering, Finance, Product Support, etc.)
- **Designer**: this is a role whose work is concern with the implementation aspects.
- **Stakeholder**: this role groups the previous roles.

When trying to unify all this information and use it in the parts identification (product structuring), we realize that we need to pass twice the first step of HOQ. In each step, different actors play their roles. In each step, we cross different views. In the first round, we cross customer expectations (customer views) against product requirements (enterprise view). During this step, a functional language is used. The result of the first step is a complete identification of the wished functional requirements for the LPS. Nevertheless, we must consider that there are characteristic of a product that affects many others. We call them "Crosscutting concerns". Concretely the term "crosscutting concerns" talks about the quality factors of the Software System (security, real time constraints, usability, persistence, etc.). In the second passage of our method, we establish the correlation between the requirements obtained in step 1 and the "crosscutting concerns". From this step we obtain a quantitative knowledge of the quality characteristics wished by the users, as well as the requirements that are affected and in what degree.

It's interesting to rank the WHATs from the most desired ones to least desired ones, so we can give precedence to some requirements. After this step, all the Product Line requirements are discovered, and we can determinate which of them are common to all the products and which are specific.

In the next step, the abstraction level is lower than in the previous step. This time, technical aspects appear. On the vertical axis we write Product Line requirements (result of the previous step), and on the horizontal axis we write a first proposal of components. This proposal is based upon object oriented concepts (abstraction, inheritance, etc.). The correlation between the proposal of components and the product requirements allow us to detect the exceed or lack of components. The components, may be correlated among them (tiled of the HOQ) based on their behavior, so we can detect the necessity to optimize this proposal of components, adding or eliminating some of them. Figure 1 summarizes all the steps.

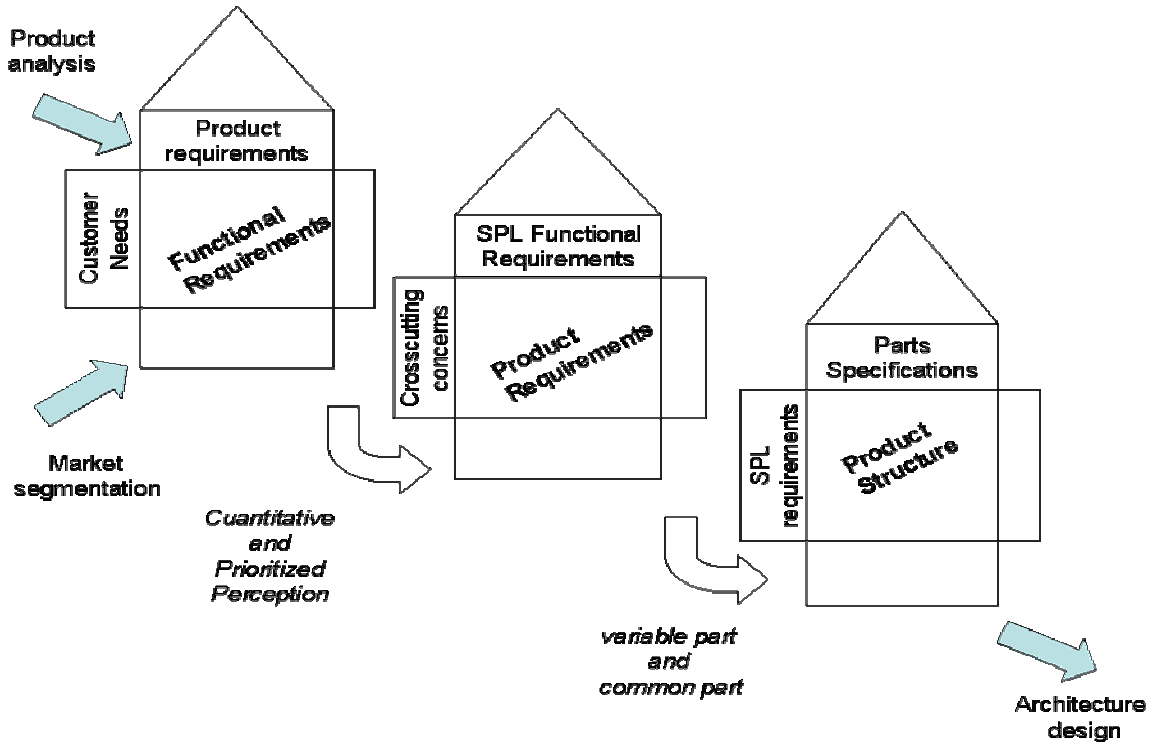

Figure 1. HOQ for SPL

4. CONCLUSIONS AND FUTURE LINES OF WORK

This paper presents a method whose aims to get the best Product Lines Structure based upon stakeholders demands. This method is integrated in a complete process for SPL development in an enterprise context.

The method applies recognized approaches from Software Engineering discipline such us Object-Oriented principles and Aspect Oriented Software Development ideas (AOSD).

The essence of the method is not a new idea. It is based on the work realized in the area of mechanical manufacture. In our research, we have just analyzed the existing product platform development methods in the mechanical area and the transference of this knowledge to Software Product Development. We must conclude that it's possible to apply mechanical methods and process to SPL as long as we complete some aspects like the adoption of SPL and the consideration of management activities. Also, it's possible to apply techniques like QFD with some adjusts.

As for future lines of work, we think that the application of the method in multidisciplinary products (i.e. products with software and electronic components) would seem to be of great interest. Also, it would turn out interesting to deal with the factor time in explicit form in the method.

5. REFERENCES

[1] [Clements 2002] Paul Clements, Linda Northrop, Software Product Lines: Practices and Patterns, Addison Wesley, 2002

[2] [Miller 1998]Thomas D. Miller, Per Elgard, Defining Modules, Modularity and Modularization-Evolution of the Concept in a Historical Perspective, Design for Integration in Manufacturing, 13th Research Seminar, 1998

[3] [Gonzalez-Zugasti 2000] Javier P. Gonzalez-Zugasti, Kevin N. Otto, Modular Platform-Based Product Family Design, ASME Design Engineering Technical Conference and Computers and Information in Engineering Conference, September 2000.

[4] [Bremmer 1999] R. Bremmer, Cutting Edge Platforms, Financial Times Automotive World, pp. 30-88. September 1999.

[5] [Fujita 1998] Kikuo Fujita, Shinsuke Akagi, Tetsu Yoneda, Makbi Ishikawa, Simultaneous Optimization of Product Family Sharing System Structure and Configuratio, ASME Design Engineering Technical Conferences, September 1998

[6] [Gonzalez-Zugasti 1998] Javier P. Gonzalez-Zugasti, Kevin N. Otto, John D. Baker, A Method for Architecting Product Platforms with an Application to Interplanetary Mission Deign, ASME Design Engineering Technical Conferences, September 1998

[7] [Dahmus 2000] Jeffrey B. Dahmus, Javier P. Gonzalez-Zugasti, Kevin N. Otto, Modular Product Architecture, ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference, September 2000

[8] [McIlroy76] "Mass-Produced software components". In J.M. Buxton, P. Naur, and B.Randell, editors, Software Engineering Concepts and Techniques; 1968. NATO Conference on Software Engineering, pp. 88-98. Van Nostrand Reinhold, 1976

[9] [DeBaud 1998] Jean-Marc DeBaud, Meter Knauber, Applying PuLSE for Software Product Line Development, European Reuse Workshop 98, 1998

[10] [Atkinson 2000] Colin Atkinson, Joachim Bayer, Dirk Muthig, Component-Based Product Line Development: The KobrA Approach, "1st International Software Product Line Conference", 2000

[11] [America 2000] Pierre America, Henk Obbink, Jürgen Müller, Rob van Ommering, COPA: A Component-Oriented Platform Architecting Method for Families of Software Intensive Electronic Products, Tutorial for SPLC1, the First Software Product Line Conference, Denver, Colorado, 2000

[12] [Weiss 1999] David M. Weiss, Chi Tau Robert Lai, Software Product-Line Engineering: A family-Based Software Development Process, Addison Wesley, 1999

[13] [Kang 1998] Kyo C. Kang, Sajoong Kim, Jaejoon Lee, Kijoo Kim, FORM: A Feature-Oriented Reuse Method with Domain-Specific Reference Architectures, Annals of Software Engineering, 5:143--168, 1998

[14] [Coriat 2000] M. Coriat, J. Jourdan, F. Boisbourdin, The SPLIT Method, Building Product Lines for Software-Intensive Systems. Proceeding of the SPLC1. Massachusetts, Kluwer Academic Publishers, 147 – 166, 2000

[15] [Brownsword 1996] Lisa Brownsword, Paul Clements, A Case Study in Successful Product Line Development, Technical Report CMU/SEI-96-TR-016, 1996

[16] [Clements 2002b] Paul C. Clements, Linda M. Northrop, Salion, Inc.: A Software Product Line Case Study, Technical Report CMU/SEI-2002-TR-038, 2002

[17] [Clements 2001] Paul Clements, Sholom Cohen, Patrick Donohoe, Linda Northrop, Control Channel Toolkit: A Software Product Line Case Study, Technical Report CMU/SEI-2001-TR-030, 2001

[18] [Brooks87] Brooks, F. "No silver Bullet: Essence and Accidents of Software Engineering". Computer 20, 4 April 1987: 10-19

[19] [Davis90] Davis, A.M. Software Requirements: Analysis and Specification. Englewood Cliffs, NJ: Prentice-Hall 1990

[20] [Faulk97] Faulk, S.R. "Software Requirement: A tutorial", 128-149. Software Requirements Engineering. Los Alamitos, CA: IEEE Computer Society Press, 1997