Experience Report on Software Product Line Evolution due to Market Reposition

Samuel A. Ajila, Antonio J. Bailetti, Razvan T. Dumitrescu
Department of Systems & Computer Engineering,
Carleton University, 1125 Colonel By Drive, Ottawa, Ontario, K1S 5B6, Canada
ajila@sce.carleton.ca bailetti@sce.carleton.ca razvandumitrescu@sympatico.ca

ABSTRACT

This paper presents the result of a study on the changes that occurred in the product line of a telecommunication equipments supplier because of the top management decision to change the product line's target market. The study examines six years of data and identifies potential relationships between changes in the product line and changes in the company's customer, inner context, and product layers. Some of the key findings are: (i) Sales are negatively related to product line growth and positively related to design turnover and the number of designers assigned to the product line. (ii) There is no relationship between the size of the code added to the product line and the number of designers required to develop and test it. (iii) There is a positive relationship between designer turnover and impact of change. (iv) The market downturn has an explicit impact on software development activities.

Categories and Subject Descriptors

D.2.9 [Software/Management]: Life cycle, Productivity, Programming teams, SQA, Time Estimation; K.6.4 [System Management]: Management audit and Quality assurance.

General Terms

Management, Measurement, Design, and Experimentation

Keywords

Product line evolution, metrics, change, designer, customer, and software product

1. INTRODUCTION

A software product line is a group of software systems sharing a common (not identical), managed set of features that satisfy a well defined set of needs (market, special mission or otherwise) and that are developed from a set of common core assets for a specific application domain [4], [5]. It is an "intermediate" level of reuse where components are used for subsequent product versions and for a family of software products.

From managerial perspective, product line is a set of products sharing common set of components and addressing related market applications [15]. A product platform is a set of components and interfaces between them [16]. A platform is an actual implementation of product line architecture [14], [15], [16], [20]. Product line architecture is an iterative process that involves many stakeholders.

The goal of the process is to separate variabilities (assets that provide distinctiveness to individual products) from commonalities (assets which belong to the product line) [4], [21], [22]. The intent is for individual products to achieve the distinctiveness required by their own functional specifications, while the overall level of commonality remains high.

There are three sources of evolution for product line architecture: (i) Moving assets from the platform to specific products. (ii) Moving assets that were part of a product to the platform. (iii) Adding new functionality or modifying existing functionality that applies to the entire product line.

The objective of this research is to study the changes that occurred in the product line of a supplier of telecommunications equipment as a result of the top management team's decision to change the product line's target market. This paper examines six years of data and identifies potential relationships between changes in the product line and changes in the company's customer, inner context, and product layers and how these relationships fit the management and engineering literature streams.

This research work is based on longitudinal study [19] of change processes. It links changes in the product line architecture of a large telecommunications equipment supplier with the company's customer base, inner context, and eight line card products over six-year period. Accordingly, the source of change in product line architecture is conceptualized to be multifaceted. It is neither linear nor singular. The output of a contextualize research has a horizontal component which describe the change in time and a vertical component which describes links between contexts.

In this paper, customers refer to the companies that buy telecommunications switches from the supplier together with the economic, social, and political circumstances in which they operate. Inner context refers to the characteristics of the product development unit's structure, designers, and inter-group relationships. Products refer to the four line card products that were repositioned so they could also be sold to the new target market and the four line card products that were developed for the new target market. The period of interest is marked by when top management decided to change the target market for its telecommunications switches to when the required product line

Workshop QUTE-SWAP@ACM/SIGSOFT-FSE12, November 5, 2004, Newport Beach, CA, USA.

repositioning was achieved. This period includes years of growth and downturn.

The goal therefore is to identify metrics [8] that can be used to records changes in product line, customers, inner context, and product layers, and to identify potential links between the layers.

2 BACKGROUND AND RESEARCH METHOD

To start with, we make three important assumptions, all consistent with the "contextualize" method [18], [19]:

1. Change in product line architecture needs to be studied in the context of other levels of analysis. For the purpose of this research, we assume that the source of change in product line architecture is the asymmetries between three interconnected levels: customers, inner context, and products. We expect that activities at some level may be more visible and rapid than at other levels.
2. Identifying the temporal interconnectedness of changes in the product line architecture, customers, inner context and products layers is important. For example, change in product line architecture is both constrained by and helps shape the other three levels of analysis.
3. History needs to be understood as events, chronology, structures, and underlying logics not just a series of events.

The study period selected for this research is six-year period starting from June 1, 1997 to June 30, 2003. During 1996, the company in question undertook a massive restructuring of the hardware support and base operating system for the product line. A new electronic repository system for design documents was inaugurated in June 1997. Development of the first product covered by this research started in May 1997. The first line of code was submitted to the revision control server (RCS) in June 1997. The first formal effort to address the needs of the service provider market occurred in November 1997 when a marketing group was established with a mandate to target the service provider market. The first release of the last product examined in this study was introduced in April 2003. During the study period, the strategic target market of the company changed from the enterprise market to the service provider market. This study allows enough time after the repositioning decision was made to observe the impact of such decision on the product line.

There are three important time related constructs in this study: the time it takes to develop a new product line release; the frequency in which a metric is collected; and the frequency at which financial results and metrics related to the customer layer are collected and made available. Data collection has been organized by product release. The time it takes to develop a product line release ranges between six to ten months. Metrics were collected quarterly. Financial results are reported every 6 months, using a financial year as a time reference. There is a 3-quarter offset between the financial year and the calendar year.

The company selected for this research provides hardware and software for communication and information to companies around the world. It has a proven record of innovation and technological breakthroughs. The company has more than 25,000 employees and has offices in countries around the world, including the United Kingdom, Continental Europe, North America, Africa, Asia, and Australia.

Four layers were selected for this study: product line architecture, customers, inner context, and products. Changes in product line architecture are interdependent with changes in the customers, inner context, and product layers.

2.1 Product Line Layer (PLL)

For the purpose of this research, a product line refers to the code comprising the set of core assets and the development effort required to change it. The common set of core assets refers to the lines of code that all products in the product line shared to satisfy the needs of enterprise and service provider customers [4], [11]. The set of core assets has two attributes: size and structure. Size can be measured in terms of the number of LOC while the structure can be measured in terms of the number of independent modules. Product line growth measures the incremental change in the size of the product line over a reporting period. For each quarter, product line growth was calculated as the number of LOC added to the product line during the quarter.

Code churn is used to measure development effort [3], [7], [9]. Code churn refers to the sum of numbers of lines of code added, deleted and modified during a development project that may require a change in the set of core assets. We think that code churn is a better measure of development effort in this study because three projects included in the study were started and stopped at various times due to internal organization issues. Therefore, using development time in this case will be misleading. For each quarter, the impact of change is measured as the ratio between the code churn and code growth. A value close to one reflects changes with little modifications to the existing code base, while larger values reflect important modifications.

2.1.1 Source of Change in PLL

Some projects may require changes only to the core assets, while others may target the core assets and a particular product [1], [6]. Within each project, development effort shall be measured separately for product line (core assets) and for each product.

There are three possible sources of change in the product line architecture [11], [17]: (i) Changes driven by individual product, i.e., changes made at the product line level to accommodate a new product, or support changes in an existing product, but without any intention to generalize the changes to other products. (ii) Changes targeted to the entire product line, i.e., when there is an explicit request to implement a certain behavior on several products within the product line. (iii) Repositioning of an architectural component (asset) from an individual product to the product line.

For each quarter in the study period, we shall report a normalized weighted score for each or the three possible sources of change described above. The score for each source of change shall be reported as:

$$Score\ (class\ \_\ of\ \_\ change\ ) = \sum \frac{size\ \_\ of\ \_\ change\ \ \_\ for\ \_\ this\ \_\ class}{total\ \ \_\ change\ \ \_\ per\ \_\ quarter}$$

An adjusted product line growth shall be calculated each quarter by excluding changes due to asset repositioning from the quarterly volume of change. While asset repositioning represents a valid source of change for the product line, there is relatively less effort involved in such change (the code exists already) than for the code generated through the other two sources of change. Table 1 below shows the metrics used to measure change in the product line layer. The data interval is quarterly.

Table 1 – Metrics for changes in Product line layer

Metric	What was measured
Size of code in the product line	Number of LOC of the product line at the end of each quarter
Code churn	Sum of lines of product line code added, deleted, and modified during a given quarter
Source of change	Relative size of change for each source of change: product line, individual product, and asset reposition.
Product line growth	Increase or decrease in the size of product line code from the previous quarter, measured in LOC.
Impact of change	Ratio between the code churn and product line growth

2.2 Customer Layer

The customer layer refers to the companies that buy telecommunications switches from the supplier and the economic, social, and political circumstances in which they operate. Customers are classified based on the type of network they operate as enterprise and service providers. Enterprise customers operate networks that carry traffic that originates or terminates within the organization that owns the network. An enterprise network is a private network. Service provider customers multiplex traffic from various sources across its network. It is neither the originator, nor the terminator of the traffic (except some control traffic). A service provider sells "services" and contracts to its customers. There is a contractual relationship between the service provider and its clients specified by Service Level Agreements (SLAs). A service provider network is a shared network.

Customers influence products directly trough explicit requirements and indirectly through increases in the buying power of their customers or potential customers. The features of the product line may influence the pool of potential customers. The addition of a feature that does not satisfy customer requirements may reduce ability to sell the products in the product line.

Table 2 shows the metrics used to measure change in the customer layer.

2.3 Inner context layer

The inner context layer includes constructs that are internal to the company but are not part of the products or customer layers. Three constructs of the inner context are examined in this research: structure, designers, and relationships between functional groups.

– **Structure**: The product development unit (PDU) refers to the R&D group. This group is responsible for marketing functions, engineering functions, customer support functions, and manufacturing support. A product line unit (PLU) is a subdivision within the product development unit. A PLU is established as an organizational tool to identify clearly the development resources allocated to each product line. The company conceptualized a PLU as an informal group comprised of several functional groups working together on the same product line.

Table 2 - Metrics for the customer layer

Metric	Data Interval	What was measured
Sales	Semi-annually	Sales in millions of British Pounds (GBP)
Size of projects undertaken to satisfy customers' change requests	Collected Quarterly and reported semi-annually	Project size was measures as the number of LOC made to an individual product or the product line because of a specific customer requests.
Number of LOC of the product line changed	Collected quarterly and reported semi-annually	Number of LOC changed by changes brought about by customers (service providers, enterprise etc.)

The PDU was responsible for the product line and the individual products during the entire study period. A functional group represents the group of designers with a common expertise and skills. A functional manager led each functional group. There were four distinct sites where the functional groups were located, three in North America and one in Europe. These sites will be referred to as HQ, S1, S2, and S3. HQ denotes headquarters where most of the development teams and the top management were located. S1, S2, and S3 refer to remote locations where the designers worked. At all times during the study period, the development was anchored around the functional groups. Each PLU is comprised of several functional groups. In addition, PLU have been created and dissolved opportunistically based on changes in the business strategy. Table 3 shows the functional groups that were part of the R&D organization.

Table 3 – Functional groups and location

Functional group	Location
Product line marketing (PLM)	HQ
Hardware design (HWD)	HQ
Base and OS software (BOS)	HQ
Interworking software (IWS)	S1
Signalling and protocol development(SGL)	HQ
Portcard software - I (PS1)	HQ
Portcard software – II (PS2)	S1
Network management(NMN)	S3
Sustaining engineering (SEG)	HQ
Quality assurance (QAT)	HQ, S1
Diagnostics and Manufacturing support (DAG)	HQ, S1

– **Designers**: These are individuals who architect, design, implement and test the products and the product line. Designers are a particular class of stakeholders in the product definition process. Their input to requirements definition is sought and taken into account. However, individuals belonging to the marketing function make the final decisions about product or product line requirements. For each line card product, changes in the number of designers allocated to it at each stage of the product's life were recorded. At each stage of the development process, new designers may be assigned to the project, while existing designers may get new assignments.

– **Relationships among functional groups**: Two metrics were used to measure the relationships among functional groups – (i) proportion of individuals assigned to each product and product line that belongs to the functional groups and (ii) code submissions reviewed by members of other functional groups.

Table 4 presents the metrics used to measure change in the customer layer. The data interval is quarterly.

Table 4 – Metrics for changes in inner context layer

Metric	What was measured
Number of designers	Number of designers from each functional group assigned to individual product and product line
Number of engineers in the business unit	Number of designers in each functional group that is part of the business unit
Number of code reviews undertaken by an engineer in group other than the code writer's group.	If the code reviewer belonged to a different group other than the code author's group, a score of 1 was allocated to the reviewer's functional group.

2.4 Product Layer (PL)

The products layer is comprised of eight products. Each product is a line card. Four line cards were repositioned from the old target market, enterprises, to the new target market, service providers. Four line cards were developed for the new target market, service providers.

Each product development project is used to organize data collection. Moreover, each product development project acts as an in-depth case study within the overall longitudinal study of change.

From the architectural perspective, a product uses a subset of the product line core assets, together with a set of architectural components that belong only to that product (i.e., product assets). We measure change for individual products by using the same approach as for the product line architecture. We measure the change for an individual product in terms of changes in size and code churn.

For each product, we shall also calculate the efficiency of the product line with regard to a specific product as the ratio between the size of the product and size of the changes required to the product line in order to support the newly introduced product. This metric is meaningful only at the first release of the product, and it measures an attribute of the platform rather than an attribute of the product.

The development of a product line is very much an ongoing process. In contrast, evolution on the individual product assets may be separated in development phases (when work is performed to satisfy a specific requirement change) and maintenance phases, when the only work performed on the product is related to bug fixes.

2.4.1 Releases and rate of Change in PL

Releases are auxiliary constructs, which apply to both the product layer and product line layer. From the customer perspective, a release is the result of a development process. A release occurs after the completion of a full regression cycle. For each of the products described in this paper, the first release of a product is comprised of hardware and software release, any subsequent release is only a software release .

During the study period, the product line introduced 17 releases . Each new product line release represents a new release for all individual products that belong to the product line at the release time. There was no independent product release outside a product line release. Four out of the 17 releases where special releases, which addressed the needs of a particular strategic customer. A special release does not go through the full regression testing cycle. However, all the features introduced in such a release went through the normal testing cycle in the next regular release. Special releases have not been taken into account in calculating the rate of change in this study.

Each release of a product line represents a change in the core assets and product assets from the previous release. As a result of a development process, which occurred between releases, the size and structure attributes of those assets changes from one release to the next. The evolution of volume of change across multiple releases will be measured using the rate of change developed by Lehman and Ramil [12]. The rate of change approximates the experience a company has with changes to a product or the product line. The rate of change is calculated as follows:

Let m_i be the value of an attribute for release i (i.e., number of modules). We define incremental change for release *i* as:

$$dm_i = abs(m_i – m_{i-1});$$

The rate of change (rc_i) for attribute *a* at release *i* is defined as the mean of incremental changes for all previous releases, not including release *i*.

Rate of change is useful to position a new release into a "historical" context. Using this concept, Lehman provides a theory for evaluating the risks of a new release [12], [13]. Let s_i be the standard deviation of the incremental growth over the last *N* releases, not including release *i*. *N* is a well defined number.

A release is:

- safe if $dm_i < rc_i$;
- risky if $rc_i < dm_i < rc_i + 2s_i$;
- very risky if $dm_i > rc_i + 2s_i$.

According to this theory, each release increases or decreases the experience the organization has with changes on a particular product. After several releases of small changes (e.g., if the product was only in maintenance mode), the experience with a particular product decreases, and a new change intensive release is risky. Table 5 below shows the metric used to measure changes in the customer layer.

Table 5 – Metrics for product layer

Metric	Data Interval	What was measured
Size of product code	Quarterly	The number of LOC of the product code
Code churn	Quarterly	The sum of added, deleted and modified lines of code for individual product
Chang es on product line	Collected once on first release	The size of the change (in LOC) made to the product line code required to accommodate the new product
Code churn on	Collected once on first	The sum of added, deleted and modified lines of code belonging to the product line code that have

product line	release	been changed in order to accommodate the new product

Table 6 – Product line releases dates

Release	Release date	Products introduced
7	16 March 1998	C-1
8	4 August 1998	
9	22 October 1998	F-1
10	19 February 1999	
11	29 November 1999	I-1
12	15 February 2000	
13	23 May 2000	
14	22 July 2000	
15	7 December 2000	
16	15 March 2001	
17	4 June 2001	E-10
18	9 September 2001	
19	21 March 2002	
20	17 July 2002	F-3, C-2, I-1 repositioned,
21	12 Jan 2003	I-3
22	4 April 2003	E-1, F-1 repositioned
23	Expected November 2003	C-1 repositioned, E-1 repositioned

3. EVOLUTION DUE TO MARKET REPOSITION

In September 1998, the Business Unit's top management team announced a strategy for service provider market penetration. The strategy was comprised of the following major points - create a new product line dedicated exclusively for the service provider market; and reposition the enterprise based product line as access devices for new service provider product line. As a result, a new engineering PLU was created in October 1998. The first two products of the service provider product line were released in November 1999. A total of four products were released exclusively for the service provider product line until the first quarter of 2002. The development of two additional products for the service provider product line was cancelled early 2002.

Table 7 – Product line size, growth, and code churn

	Product line size	Product line growth	Modules	Code churn
Q2-97	887162	0	49	0
Q3-97	908712	21550	49	29080
Q4-97	932875	24163	57	53776
Q1-98	1070762	137887	65	308881
Q2-98	1087612	16850	65	44144
Q3-98	1298743	211131	87	651321
Q4-98	1309871	11128	91	34090
Q1-99	1387612	77741	97	144970
Q2-99	1476123	88511	102	132599
Q3-99	1501621	25498	107	65933
Q4-99	1513136	11515	112	49603
Q1-00	1534013	20877	114	28981
Q2-00	1550856	16843	117	63680
Q3-00	1592422	41566	124	60083
Q4-00	1674509	82087	136	89744
Q1-01	1782930	108421	139	132296
Q2-01	1812559	29629	139	50767
Q3-01	1986804	174245	139	208572
Q4-01	2040215	53411	155	199050
Q1-02	2055490	15275	156	23394
Q2-02	2108962	53472	161	97991
Q3-02	2279105	170143	163	296774
Q4-02	2298721	19616	165	24413
Q1-03	2315529	16808	171	19718
Q2-03	2384722	69193	171	85755

3.1 Product line Reposition and Releases

In March 2001, a decision was made to "merge" the two product lines. De facto, this decision repositioned the products belonging to the enterprise product line to the service provider market.

In January 2002, the service provider PLU was formally merged with the enterprise PLU creating a common enterprise and service provider PLU. During the study period, the declared objective of the company was to introduce a new product line release every 6 months. Between 1994 and March 1998, there were six releases of the product line. During the study period, there have been 16 releases. Table 6 provides the dates of the 16 product line releases that fall within the study period.

3.2 Changes in Product line

Table 7 shows the product line size, product line growth, number of modules in the product line, and code churn for the study period. The size of the product line increases four times over the six-year study period. Three phases of growth are identified – a fast growth phase (66%, Q2-97 to Q2-99), a stale growth phase (7%, Q3-99 to Q3-00), and a revival phase (49%, Q4-00 to Q2-03).

3.2.1 Efficiency of Product Line

The efficiency of the product line measures the easiness to which a product line supports a particular new product. This is a cross layer metric; it characterizes the product line, at a certain point in time, with respect to a particular product. A higher efficiency value indicates an easier integration of the new product in the product line (Table 8). For the products, which have been repositioned, we have measured the product line efficiency twice: once with regards to the product itself (when the product has been first introduced) and, for appropriate projects, with regards to the repositioning activity (which has been a standalone, measurable project). Product line efficiency does not decrease significantly over time, (in fact, the four products introduced in 2002-2003 do have higher efficiency rates than products introduced in 1998 – 1999). Such evolution signifies a product line, which did not enter a period of decay [12], [13].

Table 8 – Product line efficiency at product release

Product	Product size	Product line change	Product line efficiency
C1	8787	1221	7.20
F1	154985	11872	13.05
I1	87779	18762	4.68
E10	161053	6102	26.39
C3	121138	7881	15.37

F3	237621	9971	23.83
I3	43859	1542	28.44
E1	181434	1109	163.60

3.2.2.1 Product Line Rate of Change and safeness

Table 9 describes the risk and safety thresholds for regular releases between 1999 and 2003. The safe rate of change and the risk threshold have been calculated using three previous releases for statistical mean and the product line growth between releases as base metric.

Table 9 – Rate of change and Risks thresholds

Re-lease	Risky threshold	Unsafe Threshold	Actual Change	Verdict
R10	140903.0	370019.4	77741	Safe
R11	125112.5	347275.5	125524	Risky
R13	125194.8	240106.2	37720	Safe
R14	110615.7	198534	41566	Safe
R16	100751.4	199992.8	190508	Risky
R18	111971.0	286217.3	203874	Risky
R19	122182.4	302379.1	68686	Safe
R20	116832.8	265818.4	223615	Risky
R22	126540.3	295198.7	105617	Safe

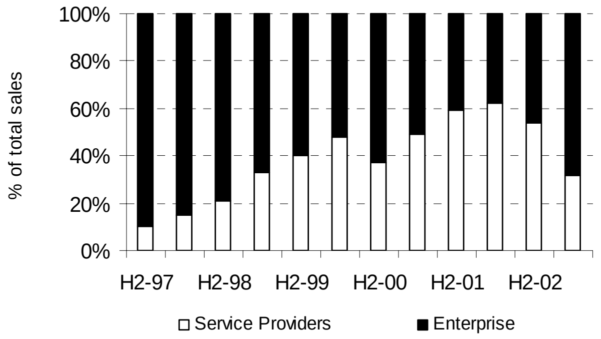

Figure 1 - Breakdown of sales by service provider and enterprise markets

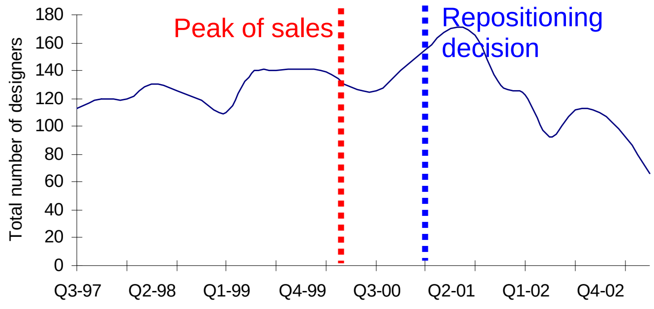

Figure 2 – Number of designers allocated to product line

The first release after the market reposition decision was taken (R18) and the first release, which contained a product, which has been repositioned, were both risky releases. None of the releases during the study period was very risky.

3.3 Changes in the Customer Layer

The size of projects driven by enterprise customers decreased over time with the decrease of their relative importance as a source of revenue. There was no project driven by an enterprise customer during the 2000-2002 market downturns. Small projects driven by enterprise customers were undertaken once again in the second half of 2002. Projects explicitly driven by service providers occurred during the second half of 2000 to first half of 2002 market decline (figure 1). As opposed to the enterprise customers, for service providers there is a certain lag between growth in the relative volumes of sales and the size of the projects they requested.

3.4 Changes in Product

In July 1997, the product line comprised four line cards. Three other line cards were at various stages of development. During the study period, development started for a total of 28 new products – out of which seven projects were cancelled before release and 11 products reached the end of life. None of the products included in this study has reached the end of life, as of 30 June 2003.

We distinguish two phases in the “development life” of a product:

– A pre-release phase, which contains the bulk of development activities. This phase terminates at the first release of the product

– A post-release phase, in which maintenance and incremental development activities do occur. The four products, which have been repositioned to the service provider market after their initial release, form the part of the post release development activities.

3.5 Changes in the Inner context

Figure 2 illustrates the number of designers allocated to the product line during the study period. The total engineering workforce assigned to the product line was between 36% and 78% of the total workforce available. Figure 3 presents the quarterly designer turnover during the market growth and the market decline phases.

4. RESULTS

4.1 Customer layer and product line layer

Figure 4 illustrates the plots of sales, the LOC added to the product line, and the LOC added to the product line minus the lines of code directly related to the March 2001 repositioning decision. For each curve, the numbers refers to percentages of the maximum number in the time series.

Three observations of interest can be made.

1. Figure 4 suggests that sales are negatively related to the LOC added to the product line. During the market growth phase, LOC added to the product line decreases as sales increase. During the market decline phase, LOC added to the product line increases as sales decrease.
2. LOC added to the product line were at the highest levels when the two strategic decisions were made - the January 1998 decision to incorporate a Web based interface to the entire product line and the March 2001 decision to merge the product lines for the service providers and the enterprise markets.
3. After March 2001, the curve that reflects the LOC added to the product line adjusted for the code required by the

repositioning decision closely follows sales. Prior to March 2001, this curve followed the curve for the LOC added to the product line.

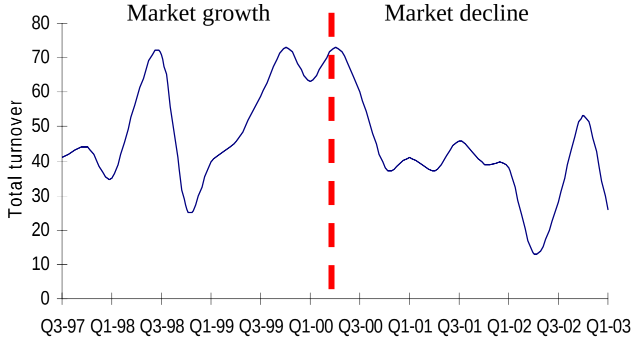

Figure 3 – Designer turnover for product line

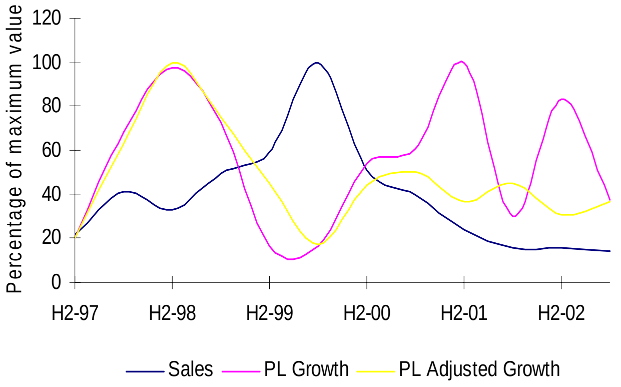

Figure 4 – Relative sales, product line growth, and adjusted growth due to repositioning

4.2 Inner context layer and product line layer

The Pearson correlation coefficient between the 24 quarterly observations of the number of LOC added to the product line and the number of software designers assigned to develop the product line is 0.009 and is not significant at $p < .10$ (two-tailed). The number of software designers included those designers assigned to the following software groups: BOS, SGL, IWS, PS1, PS2, NMN and SEG. Further examination of the data shows that the quarterly output per designer varies from 102 LOC/designer (fourth quarter 1999) to 2069 LOC/designer (third quarter 1998).

The Pearson correlation coefficient between the 24 quarterly observations of the number of LOC added to the product line and the number of test engineers assigned to the product line is 0.075 and is not significant at $p < .10$ (two-tailed).

The low correlation results suggests that the number of LOC added to the product line is not a good predictor of the number of designers required to undertake development and testing in the product line. Changes to the product line can be classified into two types: product line restructuring and product line extension. Product line restructuring includes changes to the existing code undertaken to support a new feature. Such operations entail high code churn, low number of LOC added, and very few new files added to the product line.

Product line extensions occur when new a module, class or other entity is added to the product line for the purpose of handling a specific scenario, even at the expense of duplicating code. The new entity interacts with the old 'core' via a few well defined Application Program Interfaces (APIs). Table 10 shows the impact of the change in the product line. The impact of change is calculated as the ratio between the number of LOC added to the product line and the associated code churn. This table shows that impact of change scores from products developed by remote sites are usually lower than the scores for products developed by headquarters and that groups located at the headquarters made many changes but added few lines of code while remote sites added larger amount of code but few changes. This suggests that the remote groups tend to duplicate existing code, modify it, and then add it to the product line.

4.3 Customer layer and Inner context layer

We examine two relationships in this section – the interaction between sales and power of functional groups, and the interaction between sales and designers. The study suggests that during a market decline, the power of functional groups that interact most frequently with customers increases while the power of functional groups that interacts the least with customers decreases.

The Pearson correlation coefficient between sales and designer turnover is 0.69. This coefficient is significant at $p = 0.0059$ (two-tailed). Designer turnover is driven by the actions of individual designers. At the start of a market downturn, they seek stability and quickly seek to join development projects. During the market downturn phase, increases in designer turnover are driven mainly by actions taken by the company.

Table 10 – Impact of change to product line

Product	Product line churn	Number of LOC added to the product line	Impact of change	Team	Location
E-10	23,221	6,102	3.81	SGL	HQ
F1	16,542	11,872	1.39	PS2	Remote
C1	1,401	1,221	1.15	IWS	Remote
E1	11,982	1,109	10.80	SGL	HQ
C3	20,886	7,881	2.65	PS2	Remote
F3	14,311	9,971	1.44	PS2	Remote
I3	2,871	1,542	1.86	PS2	Remote
I1	33,212	18,762	1.77	PS1/ IWS	HQ/ Remote

5 DISCUSSION AND CONCLUSION

In the first place, the decision to reposition the product line resulted in:

- **Increases in the number of designers allocated to the product line**. In December 2000, prior to the product line reposition decision, there were 126 designers assigned to the product line. This number increased to 155 in the next quarter, when the repositioning decision was taken, followed by increases to 170 and 165 in the subsequent quarters.
- **Increases in the product line growth and code churn**. Between Q4-99 and Q4-00, prior to the repositioning

decision, the product line growth was, on average, approximately 34 KLOC per quarter. In the subsequent 5 quarters (from the moment the decision was taken until the first reposition product has been released) the average product line growth was approximately 65 KLOC per quarter. In terms of code churn, it increases from an average of 58418 in the five quarters (prior to the repositioning decision) to 96628 in the five quarters after the repositioning decision.

- **Increases in independent development asset reposition from products to the product line**. The size of independent development increased after the product line repositioning decision has been taken. There are two major reasons for such increase:
  - The reposition work itself accounts for speculative development
  - The relationship with new customer base comprised of service providers was weak; therefore, most subsequent development was speculative rather than based on explicit requests.

In absolute terms, the average independent development in the 5 quarters prior to the repositioning decision was on average 9753 LOC per quarter, while in the subsequent 5 quarters, the independent development averaged 35774 LOC per quarter. In relative terms, the independent development represented 28.20 % of the total product line growth in the 5 quarters before the repositioning decision, and increased to 54.84 % of the total product line growth in the 5 quarters subsequent to the repositioning decision.

Product line growth due to asset repositioning (from individual products to the product line) in the 5 quarters prior to the reposition decision, accounted for 21060 LOC, representing 12.18% of the total product line growth. In the 5 quarters following the market repositioning decision, asset repositioning accounted for 73192 LOC, representing 22.44 % of the product line growth.

Secondly, our study concludes that code size is not associated with designer assignment, both at the product and product line levels. A rationale for such result at the product level may be the fact that each designer assigned to a product during the pre-release development phase, developed skills in a very narrow area. While this rationale applies, to a certain extent, to the product line as well, the output per designer varies, over the study period. Such variation may be explained, in part, by over and under allocation of resources, through managerial decision, during various stages of the product line development. Such result is consistent with a study done by Elbaum and Munson [7] which have found consistently low correlations between the number of designers and the code churn for some medium size (150 KLOC) software products.

Table 11 – Product size and testing effort

Product	Size	Testing effort (person quarters)
C1	8787	10
F1	154985	13
I1	87779	32
E1	181434	20
C3	121138	21
F3	237621	26
I3	43859	8
E10	161053	28

Thirdly, we also conclude that code size is not a good predictor of testing effort (measured by allocated test engineers) at either product or product line levels. Other metrics, like code coupling and cohesion are more appropriate for such estimation [3], [8]. At the same time, contextual factors (like the target market) influence the amount of testing activities required for a particular release of the product line. In addition, testing effort does not seem to depend on the product's target market (table 11). For example, I3 which targeted both service provider and enterprise markets, required 8 person-quarters of testing, while C1, - one fifth of I3's size and targeted for one market only, did require 10 person quarters of testing.

Fourthly, during the pre-release phase, there seems to be a positive relationship between designer turnover and the impact of change. A similar relationship does not seem to hold for post-release changes.

In conclusion, we outline below three important findings from this research:

- **Code size and designer assignment** - our study conclude that code size is not associated with designer assignment, both at the product and product line level. In addition, code size is not a good predictor of testing effort. There are three particularities of a telecommunication product, which may restrict the ability of predicting the required testing effort based on the software metrics only. (i) a telecom product is a hardware-software system and in many cases an important part of the decisional complexity is moved to the hardware components; (ii) the behavior of a telecom product is usually driven by asynchronous events; (iii) different applications have different levels of tolerance to defects. For example, voice products are simpler than data products – however they require a much higher level of resiliency. In addition, contextual factors (like the target market) influence the amount of testing activities required for a particular release of the product line.
- **Effects of team collocation** - From the formal process perspective, all teams included in this study have equal privileges regarding to changes to the product line code. There is no explicit approval process required in order to change a code originated by a different team; each designer is free to change any part of the code according to his/her needs. However, this study found that project teams are more likely to change existing code originated by collocated designers, rather than changing code, which has been originated by designers from remote locations. This result confirms the findings of Herbsleb and Grinter [10] and Brooks [2] related to the role of informal contact between developers. A large amount of information related to the behavior of the code is exchanged through informal communication. Such communication is minimal between non-collocated teams. At the same time, this result qualifies the finding of Cain and McCrindle [3] related to coordination between teams. In the case of a product line development, an organization may want to use different locations for the product line development versus individual product development as an informal tool for enhancing product line stability. Less informal interaction may lead to a more formal

process in the product line changes; which may be, in particular contexts, a desired outcome.

- Market effect on product and product line development - This research finds that market downturn has an explicit impact on development activities. The first reaction to the market downturn by the studied organization was to increase the development of its core product line. Such result qualifies Lehman's fifth law of software evolution, which predicts a long term decrease of the rate of growth for a given software product [12] by restricting its applicability to stable or growing market conditions. (Lehman's law applies to the studied product line during the market growth phase). The consistency of such impact across multiple organizations should be subject to further study.

The scope of this research may be extended by trying to answer two questions – Is it common behavior across organizations to respond to a decline in sales by increasing development of the code in individual product or product line targeted towards a stable or growing market? Does the power of functional groups (closest to the customers) increases during sales decline? One major limitation of this study is that the result may not be generalized to all product line evolution because it focuses on one type of product line (telecommunication).

6 REFERENCES

[1] Bosh, J & Svahnberg, M 1999. "Evolution in Software Product Lines" *Journal of Software Maintenance,* Vol. 11, No 6: 391-422

[2] Brooks, F.P. 1995, The Mythical Man-Month, After 20 Years, *IEEE Software*, Vol. 12, No 5 : 57-60

[3] Cain, James & McCrindle, Rachel 2002 "An investigation into the Effects of Code Coupling on Team Dynamics and Productivity", *Proceedings of the 26th International Computer Software And Applications Conference*, Oxford, UK : 907- 918

[4] Chastek G, Donohoe P, Kang K & Thiel S 2001 "Product Line Analysis: A Practical Introduction", *Technical Report CMU/SEI-2001-TR-001 ESC-TR-2001-001, Carnegie Mellon University*

[5] Clements, P & Northrop, L 2001 "Software Product Lines: Practices and Patterns" Addison-Wesley, New York, 2001.

[6] Cook, S, He J & Harrison, R 2001 "Dynamic and static views of software evolution" *Proceedings of IEEE International Conference on Software Maintenance (ICSM'01)*, IEEE Computer Society Press, 2001: 592-601.

[7] Elbaum, S & Munson, J, 1998. "Code Churn: A measure for estimating the impact of code change", *Proceedings of International Conference on Software Maintenance (ICSM '98)*, Bethesda, MD, November 1998: 24 - 31

[8] Fenton, N.E. & Pfleeger, S.L 1997 "Software metrics: A rigorous and practical approach", 2nd Ed, London, International Thompson Computer Press

[9] Hall, G. & Munson, J 2000 "Software Evolution: code delta and code churn" *The Journal of Systems and Software*, #54 (2000): 111-118

[10] Herbsleb J.D. & Grinter R.E, 1999 "Architectures, Coordination and Distance: Conway's Law and Beyond", *IEEE Software* Vol. 15, No 5: 63 - 70

[11] Kang, K, Donohe, P, Koh, E, Lee, J & Lee K, 2002 "Using a Marketing and Product Plan as a Key Driver for Product Line Asset Development", *Proceedings of the second software product line conference (SPLC2),* San Diego, CA, August 2002 : 336-382

[12] Lehman, M.M & Ramil J, 2001, "Rules and tools for Software Evolution Planning and Management", *Annals of Software Engineering, vol. 11 : 15-44*

[13] Lehman, M.M., 1980. "On understanding laws, evolution and conservation in the large program life cycle", *Journal of Systems and Software, 1(3): 213-221*

[14] Van den Linden, F, 2002. "Engineering Software Architectures, Processes and Platforms for Systems Families", *Proceedings of the second software product line conference (SPLC2),* San Diego, CA, August 2002 : 383-398

[15] Mayer, M, Tertzakian, P, Utterback, J, 1997, "Metrics for Managing Research and Development in Context of Product Family", *Management Science*, Vol. 43, No 1, 1997: 88 – 111

[16] Mayer, M, Utterback J, 2002, "The Product Family and the Dynamics of Core Capability", in Robarts, E (Ed), "*Innovation - Driving Product, Process and Market Change*", San Francisco, Jossey Bass: 87 – 128

[17] O'Brian L, Stoermer, C, 2001 "MAP-Mining Architectures for Product Line Evaluations", *Proceedings of the Third Working IEEE/IFIP Conference on Software Architecture,* Amsterdam, Netherlands, August 28-31, 2001: 35-45

[18] Pettigrew, A M. 1985 "Contextualist Research: A Natural Way to Link Theory and Practice," in E.E. Lawler (Ed.) *Doing Research that is Useful in Theory and Practice*, San Francisco, Jossey Bass

[19] Pettigrew, A M. 1990 "Longitudinal Field Research on Change: Theory and Practice" *Organization Science, Vol. 1, No 3, August 1990*: 267-292

[20] Robertson, D, Ulrich, K, 2002, "Planning for Product Platforms", in Robarts, E (Ed), "*Innovation - Driving Product, Process and Market Change*", San Francisco, Jossey Bass : 129 – 157

[21] Thiel, S & Perruzio, F, 2000 "Starting a Product Line Approach for an Envisioned Market", *Proceedings of the First Software Product Line Conference, (SPLC1), Denver, Colorado, August 28-31 2000.* Kluwer Academic Publishers, 2000: 495-512

[22] Thiel S & Hein A , 2002 "Systematic Integration of Variability into Product Line Architecture Design", *Proceedings of the second software product line conference (SPLC2),* San Diego, CA, August 2002: 130 – 153