Economics of Software Reuse and Market Positioning for Customized Software Solutions

M. Rokunuzzaman+ and Kiriti Prasad Choudhury+*
*School of Engineering & Computer Science, Independent University, Bangladesh (IUB), Dhaka, Bangladesh and
+Beximco Pharmaceuticals Ltd, Dhaka, Bangladesh.
zaman.rokon@yahoo.com, kpmoni@yahoo.com

Abstract—**Most of the software companies can neither be typical software product based company like Microsoft nor afford to develop each customized application for individual customer from clean slate without taking into consideration of reuse. Software companies are under increasing competitive pressure for improving delivery parameters such as cost, quality, and time. Systematic reuse is an opportunity of continued cost reduction, quality improvement and lead time reduction in software delivery. Systematic reuse largely depends on the scope of delivering customized software applications in the same market segment repeatedly to multiple customers. Thorough market analysis provides basic inputs for defining generic product concept for delivering mass customized solutions. The problem of establishing a successful new business around a generic software product concept is not challenging because of shortage of ideas, but rather problems exist in proper analysis of the market and adoption of reuse capability for continued price reduction and quality improvement to deal with evolving market forces for delivering mass customized solutions. This paper, therefore, suggests the application of market positioning strategy to benefit from software reuse for delivering customized software solutions. In this paper, economics of software reuse has been integrated with market positioning for delivering customized software solution. It is believed that such integration will improve the decision making ability of software professionals for strengthening the capability of software companies in delivering customized solutions to target market segment by taking the advantage of software reuse economics.**

Index Terms-- **Software reuse economics, Software market segmentation, Market positioning, Product-Line approach, Domain-specific engineering, Proactive investment and Customized software applications.**

I. Introduction

Software development and delivery take place in an economical context [1]. The economic analysis should be the key of taking software engineering decisions for developing business of software development and delivery. The challenge of developing software companies is no longer limited to mastering technological capability only. The tasks of crafting and executing delivery strategies are the heart and soul of managing a software business and winning in the market place. Strategy of delivering customized software solutions at competitive price appears to be a key competence need for developing software companies. Most of the start-up initiatives have the temptation to start the journey of developing software companies without taking adequate preparation of market understanding and developing the strategy of continuously improving the ability to compete. A company's strategy is the game plan which management applies to stakeout a market position in target segment, deliver its solutions, and achieve organizational objectives [2]. The central thrust of the strategy is undertaking moves to build and strengthen the company's long term competitive position and financial performance for gaining a competitive advantage over rivals that then becomes a company's strength to achieve above average profitability.

Systematic reuse is an important issue in software engineering for increasing productivity and improving quality significantly [3]-[5]. Still today, it is a research topic but already accepted as a source of huge benefits, when systematic and disciplined approach is introduced in the software engineering process.

Software product lines [6] are rapidly emerging as a viable and important software development paradigm allowing companies to realize order-of-magnitude improvements in time to market, cost, productivity, quality, and other business drivers. Software product line engineering can also enable rapid market entry and flexible response, and provide a capability for mass customization. Successful adoption of software product line practice is a careful blending of technological, process, organizational, and business improvements. Rather than building each customized solution from the scratch, as is often done today, significant gains are achievable by reusing large portions of previously built systems in the domain to construct new ones.

II. Software Resue- A Simple Idea But Complex In Practice

Since the beginning of computer programming software reuse has been practiced. Reuse of libraries, domain engineering methods and tools, reuse of design and design patters, domain specific software architecture, and component based software engineering are examples of active areas of software reuse research. An established definition of software reuse is the process of creating software systems from predefined software components. It is based on a simple and well-known idea. The most common type of reuse is the reuse of source code, but other artifacts or intermediate work products produced

doi:10.4304/jsw.6.1.31-37

during the software development process can also be reused. Reusable software items are called reusable assets. These assets may be software designs, requirements, test cases, architectures, models, design patterns, use cases, code-components etc.

Potential benefits of software reuse include reduced effort to build software systems, less time-to-market, increased quality such as robustness and decreased effort required for maintaining software [1]. Potential benefit of reuse cannot be achieved at the first project. It requires long-term investment but it increases company's assets. In addition to development and operational benefits, reuse may support strategic opportunity to lead the market, or the flexibility to respond to competitive forces and changing market conditions.

Software reuse has quite significant benefits but this simple idea is quite complex in practice [6]. It is not enough to gather interesting pieces of components into a library and offer them to people to reuse. Components have to be carefully designed and developed taking into consideration of commonality of future projects or products so that they are of high quality and work well together to meet future requirements. They have to be documented well so that re-users can understand them properly. Components have to be carefully chosen so that this extra investment will be repaid by significant reuse. This strategy works best when reuse takes place between members of a product line or product family [6].

Although significant progress has been made over the decades, many important problems remain to be addressed to increase benefit from reuse. It has been reported that a key element in the success of reuse and domain engineering is the ability to predict needed variations in future assets [7]. The necessity of involvement of a method for clearly stating reuse contexts and assumptions has been mentioned in research reporting [7]. The critical hurdle of software reuse appears to be the justification of capital investment at the beginning of projects. It is about the risk associated with the reuse of assets which are planned to be developed for future use. Such risk could be significantly reduced through market research for establishing commonality as well as variations of products to be launched customized to requirements of certain market segments. The market analysis is an opportunity assessment, which involves collecting information about market segments and forecasting demands of products having certain attributes corresponding to different segments. Market positioning is a process in which a company seeks to establish a definition of its product offerings that is consistent with customers' needs and preferences. The purpose of this work is to address risk associated with the reuse of software assets by integrating market positioning with software reuse program to reduce associated uncertainties for maximizing as well as justifying return on investment required for developing and using reusable assets.

III. Economics Of Software Reuse

Several studies have revealed that systematic software reuse increases productivity and software quality which leads to economic benefits [8]. The fact that reuse can be justified from an economic perspective which has led to the development of an array of economic model of software reuse. These economic models primarily focused on estimating benefits if reuse takes place. Moreover, development of reusable assets requires additional costs that are difficult to justify for a standalone product or under a single-project management view. Since reuse benefits accrue over multiple projects or a product line, economic modeling for management decision making should shift to long term view over a product line or multiple projects. In particular, given scare resources management needs to be able to assess whether software reuse will have kind of return that justify up-front investment in reasonability [9]. In this section, basic economic model of software reuse is stated, which basically has been adapted from previous studies. The following section integrates this model within the context of market segmentation and positioning for making upfront investment and realizing economic benefit from reuse through delivery of customized solutions to each sub-segment of the target market segment.

Let us consider that a software application is composed of *m* number of components. The reusability takes place at the component level. That means developers take the decision whether a component should be reused from existing library or to be developed from scratch. The net development cost savings *S* from the reuse of component library is estimated by the following equation [9],

$$S = \left[\sum_{i=1}^{n} (C_{N_i} - C_{CR_i}) \right] - \left[C_{PR} + A \right] \qquad (1)$$

where

n = number of products sharing reusable components.

C_{Ni} = cost to develop product *i* without reuse.

C_{CRi} = cost of creating product *i* with reuse.

C_{Ni}-C_{CRi} = expected producer's cost saved for product *i*.

C_{PR} = expected cost that the producer incurs in producing the reusable components.

A = reuse-specific overhead and setup costs incurred by the family of products.

The incorporation of the uncertainty as to whether the component will indeed be reused requires that the probability of each reuse instance should be estimated to compute the expected consumer savings. The model then becomes,

$$S = \left[\sum_{i=1}^{n} (C_{N_i} - C_{CR_i}) \right] p_i - \left[C_{PR} + A \right] \qquad (2)$$

where, p_i = probability that reuse instance *i* will be actualized.

The objective of this research is to increase this probability of reuse and the number of products which will be reusing them for maximizing cost savings. The expected benefits may have the impact on other aspects of uncertainty such as development cost saving, which depends on whether the developer takes advantage of black box reuse opportunities or modifies the component. In such situation, the probability developer will opt for black box or for white box reuse would need to be assessed in order to estimate the expected development cost savings.

IV. Economics Of Reuse In Delivering Customized Solutions

Although technologies for software productivity improvement by taking the advantage of reuse through domain engineering started to appear in early 1980s, application of these technologies in individual settings and stories of success have only been reported in recent past[7]. It has been observed that market demands increasing product variation and shorter introduction time of software intensive solutions [10]. These two trends clash, unless we are able to reuse software. Proactive product line approach increases systematic software reuse. These product lines are setup to create a family of products, and they rely on a prior analysis of commonalities and differences of the members of the family. The commonalities are embodied in an overall architecture, while the differences result in variation points. Offering of customized solution or individual product to each sub-segment of the target market results in filling variation points. Upon careful market analysis, product line architecture could be conceived to create software components with explicit context dependencies and variation points, which then can be combined in different ways into products customized to the requirement of each sub-segment of the market. The blending of market segmentation with component based software engineering as a strategy for market positioning is an opportunity to maximize profit from software reuse.

It is conceived that a software company has a library of components having the potential of being reused. The development cost saving C_s of delivering a customized solution by making use of this reusable library is proposed to be estimated by the following equation,

$$C_s = \sum_{i=1}^{m} c_{n_i} - \left[\sum_{i=1}^{o} \left(\frac{c_{r_i}}{n_i} + ci_{r_i} \right) \right] \qquad (3)$$

where

m = size of the application in number of components.
c_{ni} = cost of new development of the component *i*.
o = number of components are reused.
c_{ri} = cost of development of reusable components.
n_i = number of customized delivery will be using a reusable components.
ci_{ri} = cost of integration of a reusable component *i*.

If the value of n_i is increased, the cost of a reusable component *i* decreases or the benefit from this component increases. Therefore, the cost of delivery of a customized software solution according to Eq. (3) depends on both individual reused component specific n_i and total number of reused components, o. On the other hand, overall cost savings *S* from reuse according to Eq. (1) also depends on number of deliveries which are making use of reusable library. Therefore, it could be stated that we need to figure out means to increase values of n_i, n, and o for minimizing the cost of customized application delivery by taking the advantage from reuse. Our proposed approach of market segmentation and positioning in the target market for repeated delivery of customized software solutions by making use of a component library has the capacity to increase values of these three important variables: n_i, n, and o.

V. Market Segmentation And Positioning Of Customized Software Application Delivery

The market consists of many types of customers, products, and needs, and the marketer has to define which segments offer the best opportunity for achieving company objective [11]. The process of dividing a market into distinct groups of buyers with different needs, characteristics, or behavior who might require separate products is called market segmentation. After a company has defined market segments, it can enter one or many segments of a given market. Market targeting involves evaluating each market segment's attractiveness and selecting one or more segments to enter. Market positioning is arranging a product to enter and expand in the target market segment.

Product differentiation is the process of distinguishing a product to make it more attractive to the target market segment. Differentiation is a source of competitive advantage. In economics, successful product differentiation or customization leads to monopolistic competition. Cost of differentiation is usually high for developing each differentiated product from clean slate.

Software market segmentation is the division of a market into different groups of customers with distinctly similar, not the same, needs and software solution requirements. It is about the division of a mass market into identifiable and distinct groups or segments, each of which have common characteristics and needs and display similar responses to certain software solutions. The purpose of market segmentation is to leverage scare resources. Positioning is about the design and attributes of a product and its offering in the target market segment. In order to continuously increase the competitive strength in the target market segment, a software company needs to continuously find means of improving quality, reducing cost and lowering time of delivery. It appears that mass customization in delivery of software solution to a similar group of customers of the target market segment such as hospitals and clinics appears to be a required strategy to strengthen the market position.

With a product line concept, a firm fragments its target market segments into sub-segments that it serves sequentially [12]. To begin with it has been assumed that a software company has entered the target market segment with a basic solution. Let us develop a simple model by considering only two types of customers in that target market segment. Fraction *A* of customers have high valuation and the fraction *1-A* have low valuations of the same basic solution. High and low valuation customers value an undifferentiated version of the product v_h and v_l $<v_h$ per period. The ratio of v_h/v_l will be interpreted as a measure of the price premium that the high segment is willing to pay relative to the low one. Both the firm and the consumers live forever and have a discount factor $\beta(t)$. Customers' willingness to pay *p(t)* for the undifferentiated product keeps decaying over time as shown in the following relationship [12],

$$p(t) = \frac{v_c}{1+\beta(t)} \quad c = l, h \tag{4}$$

Let us take an example of the entry of a software firm in the machine vision based textile defect detection software [13] market segment of Bangladesh. It is to be noted that Bangladesh has a large textile industry with export of more than US$ 12 billion. In this industry, almost 3 to 5% fabrics is wasted due to the presence of undetected defects. It is estimated that almost US$200 million could be saved per year by improving textile defect detection accuracy by introducing machine vision based defect detection software solution. The company has developed basic version of the software which can deal with limited variations of relevant parameters such as lighting, color, texture, speed, etc. Only a small fraction of companies of the target market segment will be target buyers of this basic software of limited capability. And highly likely it will not be feasible for this company to earn expected profit by delivering this software to this small fraction of the target market. Due to the entry of competitors, the willingness to pay for this basic software will be likely decreasing over time as stated in Eq.(4). Moreover, due to the variation of fabrics which different textile mills are producing the software should be customized to meet their requirements. Such heterogeneity of requirements, limitation of willingness to pay, and competitive pressure for differentiated software solution to meet varying requirements have rendered isolated design and production of individual software solutions essentially obsolete. Across industries, standard practice involves lines of product variants that reduce cost via economies of scale and scope for reaching multiple market segments. In software industry it is about the reuse of software components. Software components which are developed the basic version could be reused for developing differentiated solution customized to the requirements of each sub-segment for benefiting from economies of reuse as stated in Eq.(1)-Eq(3). The development strategy for enjoying economies of reuse for offering customized solution to each sub-segment of the target market segment focuses on the tradeoff between increased commonality among these customized solutions and the resulting decrease in the ability to meet distinct performance targets for each solution variant.

Let us now investigate how reuse strategy help this software company to develop leading position in this market segment. Basic software sub-system of this software are (1) Image capturing, (2) Image processing, (3) Feature extraction, (4) Classification, (5) Making decision about defects. The objective of each of these modules or subsystem is to find a feasible design that exhibits product attributes matching the targets set during market positioning as closely as possible. Here the vector of product attributes $\boldsymbol{Z}_j$ for product *j* represents a set of objective, measurable aspects of the product, observable by the customer, resulting from engineering design decisions. In each engineering design subsystem *j*, search is conducted with respect to a vector of design variables $\boldsymbol{x}_j$, which represents decisions made by the designer that are not directly observable by consumers but that effect the reusability of components of that system for deriving subsequent differentiated solutions meeting customized requirements of other market sub-segments. Therefore, in order to benefit from software reuse economics design decisions to be made for the design of the current solution to be linked with the market segmentation and positioning strategy. The market positioning objective is to maximize profit Π with respect to the price which is chargeable for each differentiated or customized solution sales which could be made in each sub-segment, and the cost of developing each customized solution. Although firms can specify arbitrarily sophisticated profit functions based on their experience, internal accounting and historical demand, we propose to use a simple profit formulation based on revenue minus cost of each customized solution in the form of following equation:

$$\Pi = \sum_{i=1}^{N}\sum_{j=1}^{M} p_{ij} - \sum_{i=1}^{N} C_i \tag{5}$$

where

N = number of sub-segments of the target market segment.

M = number of target customers or potential sales in each sub-segment.

P_{ij} = price which could be charged to a customer of each market segment.

C_i = the cost of developing each customized solution targeting the *ith* market segment.

It could be easily stated that there is a strong relationship between Eq.(1), Eq.(2), and Eq.(3). Design decisions which are taken in customizing solution for a particular market segment should be taken for maximizing the reuse of newly developed components in developing solutions customized to other market sub-segment as well. Such decision will have significant impact on improvement of total savings from reusability as stated in Eq.(1), reduction of cost of producing each sub-segment specific customized solution as stated by Eq.(3) and maximization of the profit by delivering customized software solution to each sub-segment of the target market as stated by Eq.(5). From such observations, it could be concluded that there is a strong relationship between market positioning strategy and benefiting from software reuse. Therefore, software design decisions should be strongly linked with the market segmentation and positioning strategy.

It should be noted that the ultimate purpose of domain engineering, software product line and systematic software reuse is to improve the quality of the products and services that a company provides and, thereby, maximizes profit [7]. Despite the reality that software reuse will only succeed if it takes good business sense, very often software engineering decisions loses sight of this goal when considering the technical challenges of software reuse. The cost for evolving reusable software assets and retrofitting customized or differentiated solutions with new assets can be high. There is a need of making upfront investment for building reusable components. There are many options for the management for allocating capital by an organization to maximize return to shareholders. It has been reported that software reuse will only be chosen for investment if a good case can be made that it is the best alternative choice for use of capital[7]. In the corporate environment, the implementation of a reuse program requires a decision about when and where a capital investment is to be made. Development of reusable assets requires a capital investment and there should be a strategic decision as to whether investment will be made proactively or reactively. And this investment should be made at the beginning of developing a software business proposition targeting particular market segment in a proactive manner. Proactive investment strategy for software reuse analyzes the target market segment, conceives product line, and defines architectures; then reusable assets are designed and implemented by taking foreseeable product variations into account. Such approach tends to require a large upfront investment, and returns on investment can only be seen along with the progression of positioning in the target market segments. This approach is suitable for organizations that can predict their product line requirements well into the future. It is obvious that there is an investment risk with this approach due to uncertainty associated with the variations of product line requirements. This risk could be minimized by linking software reuse strategy with the market understanding, suitable market segment selection, and developing positioning strategy. It is believed that the integration of software reuse economics with the market segmentation and position strategy established in this paper has the potential to enable software engineers to develop convincing software reuse based strategic market position at the very beginning of software development. As a result, capital will be allocated for aligning design decisions along market positioning strategy for delivering customized solution to each sub-segment for enjoying cost and quality benefits from software reuse for maximizing profit of the company.

VI. Product Line Management For Customized Delivery Of Software Solutions

This section generalizes concepts as presented in previous sections of benefit maximization from software reuse through integration of software engineering design decisions with the market understanding, segmentation and positioning issues.

A successful product line design for maximizing benefit from software reuse should have a very close coordination between marketing and engineering teams in balancing the inevitable tradeoffs. Product lines that evolves as "optimal from marketers perspective may not be "optimal" from an engineering viewpoint, and vice versa [14]. In the design of a product line, marketing and engineering considerations are highly interdependent with a strong influence to reusability of software assets. Heterogeneous product requirements by different sub-segments of the market should be taken into consideration for leveraging the potential cost synergy from reuse across products or customized solutions in the product line. These highly interconnected relationships between the two domains determine that any required action in one domain can potentially influence the outcomes in the other domain. Therefore, for the design of an optimal or near optimal product line, the marketing and engineering requirements cannot be pursued separately or even sequentially.

Despite the compelling need for a unified framework that integrates the software design considerations from both disciplines concurrently for increasing economic benefits from software reuse there has been very limited knowledge about it in the software engineering literature. And for this reason, we propose that marketing issues should be integrated in software design for taking design decisions that not only deliver desired customized product for particular sub-segment, but also takes into consideration of the revenue and cost interactions across products in the product line for maximizing economic benefit from software reuse. Basic concept of this unified framework is depicted in Fig. 1 for a product to meet 5 product requirements by choosing design options from a set of 6. Selection of a design option has an implication on both product attributes as well as reusability of software components. A close mapping should be done between product attributes and design option in order to maximize profit serving all sub-segments of the market.

Designer options

Product requirements	y_1	y_2	y_3	y_4	y_5	y_6
x_1	√					
x_2		√	√			
x_3			√			
x_4			√	√		
x_5					√	√

Fig. 1: Mapping from design options to product requirements

Within the context of customized software development by taking advantage from reusability, we expect that designer will choose a design option from a set of available ones to maximize reusability. Remaining in line with Eq.(1), the cost reduction due to software reuse by taking advantage of commonality among products of the product line can be generalized by taking into consideration of discounted factor λ_r by the following relationship [15]:

$$\lambda_r = \begin{cases} 0 & \textit{No commonality among products} \\ \dfrac{l_{shr}}{L} \ell_r & \textit{Otherwise} \end{cases} \tag{6}$$

When there is no sharing of the component r among the products or sub-segment specific customized solutions, the discount factor is zero. The commonality significance factor ℓ_r represents the degree of cost savings by sharing the *rth* software component in the product line. L represents the total number of product or customized solutions and l_{shr} represents number of products that are sharing the *rth* component. When all the products of the product line are sharing the same component r, the discount factor λ_r equals to the corresponding commonality significance factor ℓ_r. Otherwise, the discount factor is a portion of commonality significance factor depending on the degree of sharing of the particular component in the product line. As evident when the degree of component sharing increases among the products in the product line, the cost of producing each customized product meeting requirements of the corresponding market segment decreases.

The revenue model is an important factor to consider in designing the roadmap of launching one after another product customized to each sub-segment of the market. The primary focus of our revenue model is to predict the expected revenue of each product line candidate [16]. In this model, we take into account the potential cannibalization among the products in the product line as well as the effect of competitive offerings. Like most of the product line revenue models, we start with prediction of the expected market share for each product in the product line. Given that one of the essential issues in product line design is to accurately forecast the market share of each product in the product line, we specifically account for the inherent variations in the products' utility estimates in market share prediction. The expected product line revenue can be estimated as[16]

$$E(\omega) = \sum_{l=1}^{L} [E(MS_l) \times p_l \times \Theta_l] \tag{7}$$

where

$E(\omega)$ = expected revenue from the product line.

$E(MS_l)$ = expected market share of the *lth* product.

p_l = unit price of the *lth* product.

Θ_l = size of the market sub-segment corresponding to *lth* product.

We have assumed that the target market segment has been divided into L sub-segments. During the planning stage, we propose to analyze commonality and variations of software products to be launched customized to each of these L sub-segments of the market. Based on the commonality, investment is recommended to be made in the software reusability program for minimizing cost of each product customized to corresponding market sub-segment. The expected market shares depends on the size of each sub-segment as shown in Eq.(7). Expected market share also depends on the utility and price of competitors' offerings and scope of market expansion.

Product line optimization searches for a profit-maximizing by simultaneously integrating the design options, customers' requirements, revenue, and cost into an optimization model. Mathematically, this proposed optimization problem to be expressed as follows:

$$\max_{\{y_1, y_2, \ldots, y_n\}} E(\pi) = \sum_{l=1}^{L} [E(MS_l) \times (p_l - VC_l) \times \Theta_l] - FC_L \tag{8}$$

where

y_n = a set of design options for dealing with commonality and variations of the software product to derive benefits from reuse.

$E(\pi)$ = expected profit.

VC_l = variable cost of each product line.

FL_L = the fixed cost of the product line.

As shown in the Eq.(8), the optimization problem aims to target to select designers' options having influence on reusability and product attributes of each product of the product line to maximize firm's expected profit subject to a set of constraints. Within the context of maximizing benefits from reusability such constraints will be guided by the set of commonality and variations of the set of products of the product line.

The upfront cost for the reusability program could be factored in the fixed FL_l of Eq. (8) of the product line. The cost associated with the reuse of existing components and cost for product specific to new components in developing customized version of the product for each market sub-segment could be included in VC_l of Eq.(8). The value of VC_l will largely depend on the discounted factor λ_r as defined in Eq.(6), which is largely determined by the reusability level.

It has been reported that there is a growing importance of product customization capability development among firms to compete in market segments where scope of differentiation is high [17]. It is widely believed that software component in industrial as well as consumer products has been growing in rapid space [10]. Therefore, software reuse strategy could be an important capability for not only software firms but also for software intensive product firms for reducing cost of delivering differentiated products customized to each market segment.

VII. Summary & Conclusion

Since the beginning of programming software reuse has been of interest because people want to build software solutions that are bigger, more reliable, less time consuming, less expensive, and that are delivered on time. Although this is a simple idea and benefits are clearly visible once successful reuse of software assets developed in past projects is made in current projects. Despite the simplicity of the idea and visible benefits, systematic reuse as a strategic capability to deal with ever-growing complexity and diversity in products, and decrease in lead time has not been realized with mass satisfaction yet. The key bottle neck appears to be the ability to predict commonality and variations in products to be offered to justify proactive investment for reusability program. In this paper, we have presented the concept of integration of engineering design decisions with market segmentation and position strategy for increasing this prediction ability. Limitation of this work is that further development of this concept is required with the support of empirical investigation to offer scientific foundation for decision makers to make proactive investment for strategic use of software reusability to deliver software or software intensive solutions customized to each sub-segment of the target market. It is believed that such development will significantly help software and software intensive product companies to deal with ever-growing challenges of complexity, diversity, lead time, and cost through software reuse strategy. It appears that integration of reuse concept with market positioning is very much a new concept. There is an immediate need for this type of research for maximizing benefit from software reuse.

References:

[1] K. Schmid and M. Verlage, "The Economic Impact of Product Line Adoption and Evolution," *IEEE Software*, vol. 9, no. 4, pp.50-57, 2002.

[2] A. A. Thomson, J.E. Gamble, and A.J. Strickland, Strategy: *Winning in the Market Place*, McGraw Hill, 2004.

[3] M. Morisio, M. Ezran, and C. Tully, "Success and Failure Factors in Software Reuse," *IEEE Trans. Software Eng.*, vol. 28, no. 4, pp. 340-357, 2002.

[4] Mohagheghi P, Conradi R, Killi OM, and Schwarz H, "An Empirical Study of Software Reuse vs. Defect-Density and Stability," 26th International Conference on Software Engineering (ICSE'04).H. Poor, *An Introduction to Signal Detection and Estimation*. New York: Springer-Verlag, 1985.

[5] T. Menzies and J. S. Di Stefano, "More Success and Failure Factors in Software Reuse", *IEEE Transaction on Software Engineering*, vol. 29, no. 5 2003.

[6] K. Pohl, G. Bockle, and F. v.d. Linden, *Software Product Line Engineering; Foundations, Principles, and Techniques*, Springer, 2005.

[7] W. B. Fakes and Kyo Kang, "Software Reuse Research: Status and Future," *IEEE Transaction on Software Engineering*, vol. 31, no.7, July 2005, pp. 529-536.

[8] P. Mohagheghi and R. Conardi, "Quality, Productivity, and Economic Benefits of Software Reuse: a Review of Industrial Studies," *Journal of Empirical Software Engineering*, vol. 12, 2007, pp. 471-516.

[9] R. Malan, and Kevin Wentzel, Economics of Software Reuse Revisited, Hewlett-Packard Company, 1993.

[10] R. v. ommering, "Software Reuse in Product Populations," *IEEE Transaction on Software Engineering*, vol. 31, np. 7, pp.537-550, 2005.

[11] G. Armstrong and P. Kotler, *Marketing: An Introduction,* Sixth Edition, Pearson Education Services, Inc, 2003.

[12] Pascal Courty, *Some Economics of Product Line Management, Department of Economics and Business*, University Pompeu Fabra, Spain, November 20, 1998, http://bonvent.upf.es/~courty/my-public-files/wkpp.html.

[13] M. Salauddin and M. Rokonuzzaman, "Adaptive Segmentation of Knit Frabric Images for Automated Defect Detection in Semi-structured Environment," *The Proceedings of the 8th ICCIT Conference*, 2005, pp. 255-260.

[14] Jiao, Jianxin and Yiyang Zhang, "Product Portfolio Planning with Customer-Engineering Interaction", *IIE Transactions*, 2005, vol. 37, no.9, 801-814.

[15] Luo, Lan, P.K. Kannan, Babak Besharati, and Shapour Azarm, "Design of Robust New Products under Variability: Marketing Meets Design", *Journal of Product Innovation Management*, 2005, vol. 22, no. 2, 177-192.

[16] L. Luo, B. Besharati, P. K. Kannan, and S. Azarm, "Product Line Design for Consumer Durables: An Integrated Marketing and Engineering Approach," *Working Paper*, 2008, U.S. National Science Foundation through Grant Number DMI 0200029.

[17] D. Bernhardt, Q. Liu, and K. Serfes, "Product Customization," *European Economic Review*, vol. 51, pp. 1396-1422, 2007.