



Systematic Review on Software Product Line Engineering (SPLE)

Manjinder Kaur
(M.Tech. Scholar)

Department of Computer Science and Engg
Lovely Professional University, India

Parveen Kumar
(Asst. Professor)

Department of Computer Science and Engg
Lovely Professional University, India

Abstract: - *Software Engineering is a very broader term as it is not individually taken but involved with other area like Cloud Computing, Distributed Databases, Wireless Networking, Data Mining etc. Besides this, Software Engineering involves further sub areas or minor areas like Component-based engineering, Software Product Lines (SPLs), Software quality assurance, Software Reliability etc. This paper describes the concept of Software Product Lines. Software Product Lines are used in many areas, combining to form new technologies and products. A product line is a group of products that share a common development platform and vary by the composition and implementation method for the functionalities. Comparing with that per-product development approach, SPLs has the benefits of higher productivity and shorter time-to-market, more effective component reuse and reduced product development costs. Software product line development (SPLD) promotes component reuse and provides a clean separation of concerns between different layers in system architecture.*

Keywords: - *Software Product Lines (SPLs), Product Line Engineering (PLE), Core Assets, Software Product Line Development.*

I. Introduction

In today's scenario, every organization is following the concept of shifting from single software product development to product lines which is a new trend in the future times, as it is providing reduction in cost, quality enhancement and decreasing product development time. Software Product is very much expensive and complex to develop as it requires a broad range of user requirements. Therefore, with the product line strategy, thousands of features involving the most common ones including special features are taken or concerned for a particular customer needs. In product development, initial stages product lines take more time as compared to development of similar products using the singular system as it requires some efforts for planning, architecture and realization of the infrastructure. But with the infrastructure development, new product can be easily developed just by reusing that infrastructure due to which it will take less time in developing new products. A product derived from a software product line consists of various components selected from existing component libraries; these components communicate with a common platform to perform specific functionalities. Skilled software engineers use technologies and practices from a variety of fields to improve their productivity in creating software and to improve the quality of the delivered product.

II. Software Product Lines

A Software Product Lines are defined as a family of different products which shares same set of core assets or it can be said, a product line consists of multiple systems, which have same architecture and share common core assets with variability among systems. A core asset includes shared components, framework or infrastructure, tools, process, documentation, test cases as these are reused. Basically, SPL is a family of products designed to take advantage of their common aspects and predicted variability in order to improve quality, delivery time and reduction in cost. Product line engineering (PLE) helps to design, develop, deliver, and evolve a portfolio of common products, with feature variations and functions, through which each stage of the systems and the software development lifecycle from requirements to design, development and testing.

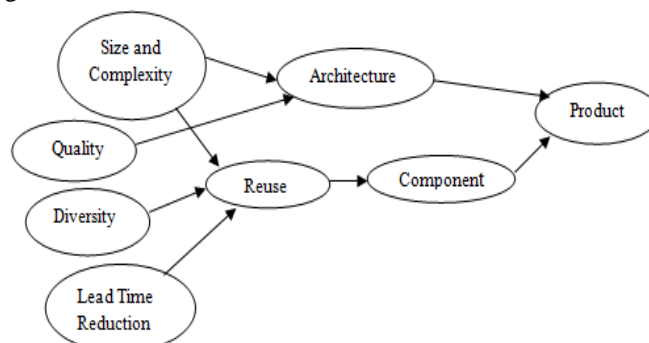


Figure 1: Basic Arguments for Software Product Lines

PLE is basically delivering efficiencies in design, development and deployment in products. To produce a product from a product line, and it should be instantiated through the following two steps:

- a) **Selection:** unneeded functionality (i.e., assets) is stripped, assets which are needed are selected, variation is also solved
- b) **Extension:** additional assets are added for the remaining variation points (possibly created from scratch)

Each core asset is designed to create systems to match the quality criteria of product line. Core asset development is its designing and management in the product line. A product line is a collection of products addressing a particular objective. It builds different models of a product using common assets. Using this technique to build a product is economic and efficient. Basically, the idea is integration instead of creation. From time to time, organization will add new member in product line based on the products that has been developed. Product evolution comes from 2 sources:

- a) **External Source:** New element from produce/ manufacturer to be included in the product line and new product will be produce from it.
- b) **Internal Source:** For product function in the scope of product line will going use the function which already exists. If it is not present, then a new function will be generated, and it will be analyzed as whether it is to be added in the product line or not.

Product Line Development: The Product Line Development three phases to work with the products:

- a) **Core Asset Development:** Core assets or artifacts are created and maintained in product line in order to create systems or products for matching quality criteria.
- b) **Product Development:** It is mainly development or the creation of products or systems out of the core assets from product line. If an asset which is required by the system is not in the core assets, then that core asset need to be created in order to satisfy the needs.
- c) **Management:** It is involved in order to make sure that the two parties are interacting correctly. Assets are also identified to know from which part of the product line it belongs and which are part of development of the individual product system. Also the management involves management of individual projects in the product line, and the universal product line managers.



Figure 3: Product Line Development by Clements et al. (2002)

III. SPLE Framework

Software product line has two major development parts domain engineering and application engineering which are also called two stages for the development of the product under SPLs [5].

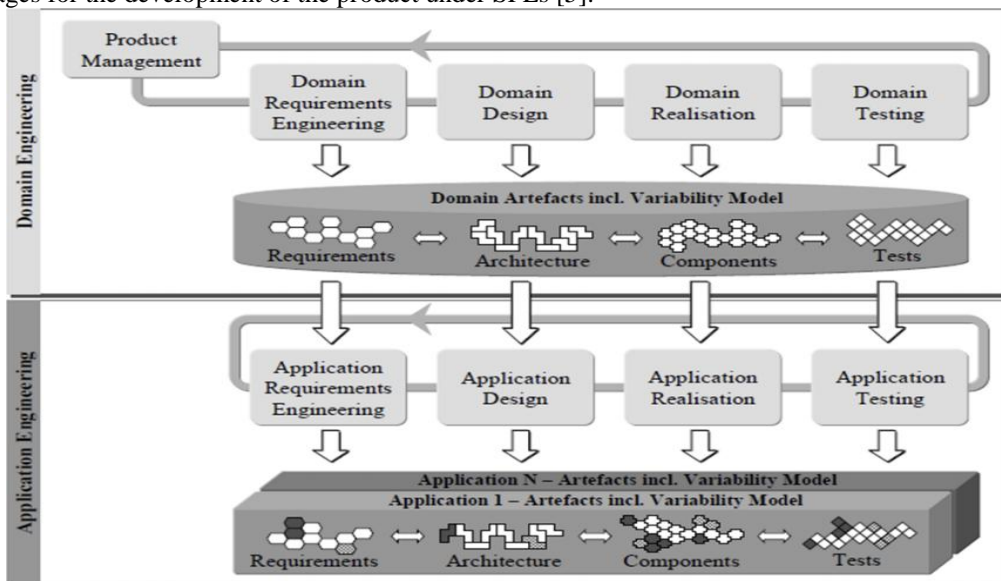


Figure 2: SPLE Framework proposed by Pohl et al. (2005)

- a) **Domain engineering** is associated with the development of core assets which are common in a family of products. The development of core assets is based on the commonality (common features) and variability (varying features) in the products.
- b) **Application engineering** is the development of individual products by reusing the core assets and adding functionality which is specific to each product.

Software Product Line is different from Individual Systems: - As in Individual Systems or single systems, different products have to be developed whether to involve new products or not. A product is started from scratch and new core assets are added to them. In Software Product Line, a single product will be developed just adding the new assets a developer want to add besides developing new product one by one having same assets.

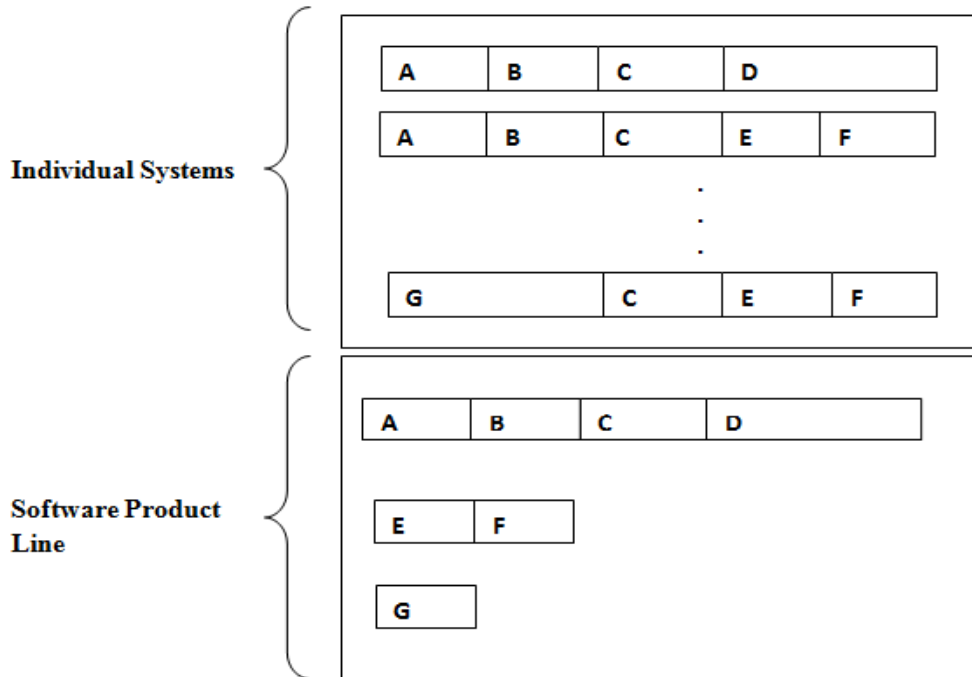


Figure 4: Individual Systems vs. Software Product Lines

Various Software Product Line Development methods are:

- a) FAST (Family-Oriented Abstraction, Specification and Translation)
- b) FODA (Feature-Oriented Domain Analysis)
- c) FORM (Feature-Oriented Reuse Method)
- d) RSEB (Reuse-Driven Software Engineering Business)
- e) FeatuRSEB
- f) ConIPF (Configuration of Industrial Product Families)
- g) PuLSE (Product Line Software Engineering)
- h) KobrA(KomponentbasierteAnwendungsentwicklung or Component-based Application Development)

IV. Product Line Engineering

Product Line Engineering:- System and Software Product Line Engineering, abbreviated as PLE, can be summed as engineering the similar products in a very efficient manner in order to satisfy the customer needs and also the organizational workflow. “Engineering” in product lines means the activities are taken into account involved in planning, producing, delivering, and deploying, and retiring products etc. As per the first generation roots, Second Generation PLE (2GPLE) is a first-class engineering practice centred on a strong factory pattern (Big Lever Software, Inc.). This is an innovative approach in experiencing the mainstream adopting across various industry sectors, in organizations ranging from small to the largest in the world. Second Generation PLE (2GPLE) is centred on a factory pattern.

As shown in the figure, engineering artifacts can include various requirements, the models and the designs, different source code, test cases, and much more, all of which need to be managed and produced in variants that match the product. These artifacts that support a product are called assets. A product configuration takes a product specification as input and configures the assets appropriately for that product. SPLE is the discipline of creating and managing software product lines [9].

A. Big Lever Software Gears: Big Lever Software provides the industry’s foremost PLE solution, offering the organization various infrastructure, tools, utmost practices and methods which are needed to create and advanced and efficient means of production for your product line. The Gears Product Line Engineering Tool and also Lifecycle Framework enable to create a programmed and efficient means of production of product for product line portfolio. Gears, whole product line portfolio can be managed as individual and automated production systems.

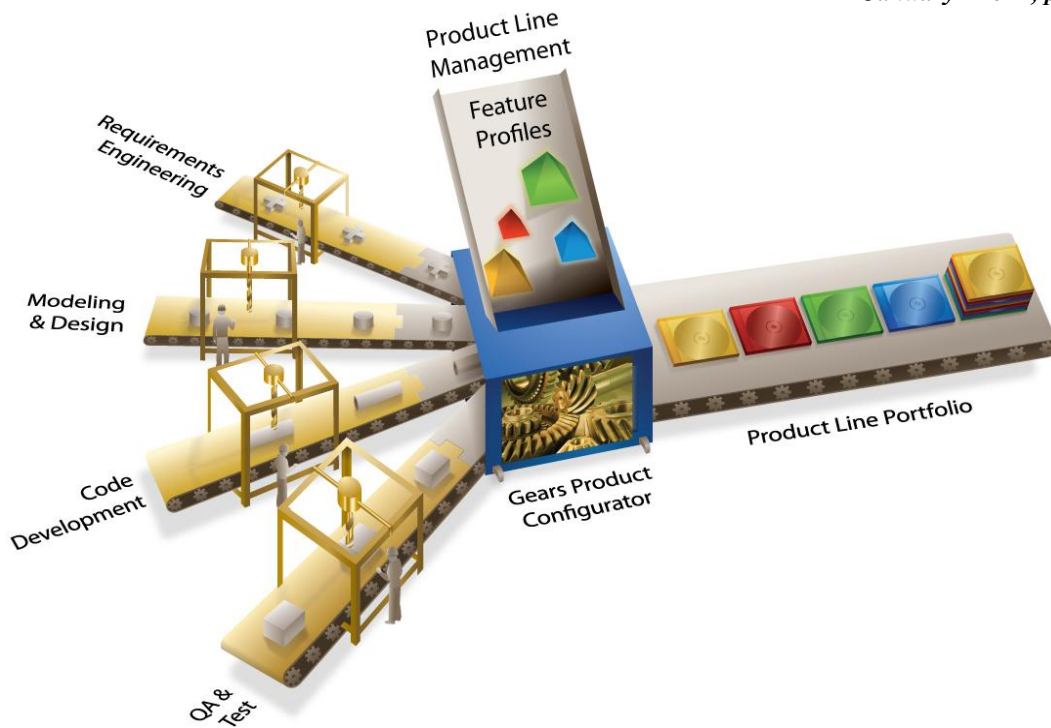


Figure 5: Product Line Management using Gears Product Configuration (Big Lever Software, Inc.)

B. Three Dimensions of Product Line Engineering [9]:

- a) **Multi-Product:** It provides the feature-based variation management and automated production line needed in delivering multiple products in a product line as multiple products will be developed or generated.
- b) **Multi-Phase:** This involves the tools needed for supporting the multiple phases of a product line. The tools may be different from the different aspects.
- c) **Multi-Baseline:** Configuration management and change for a product line is done on multiline baselines to the PLE assets instead of a single product base line. There may be number of the baselines as per the production of the various products under SPL.

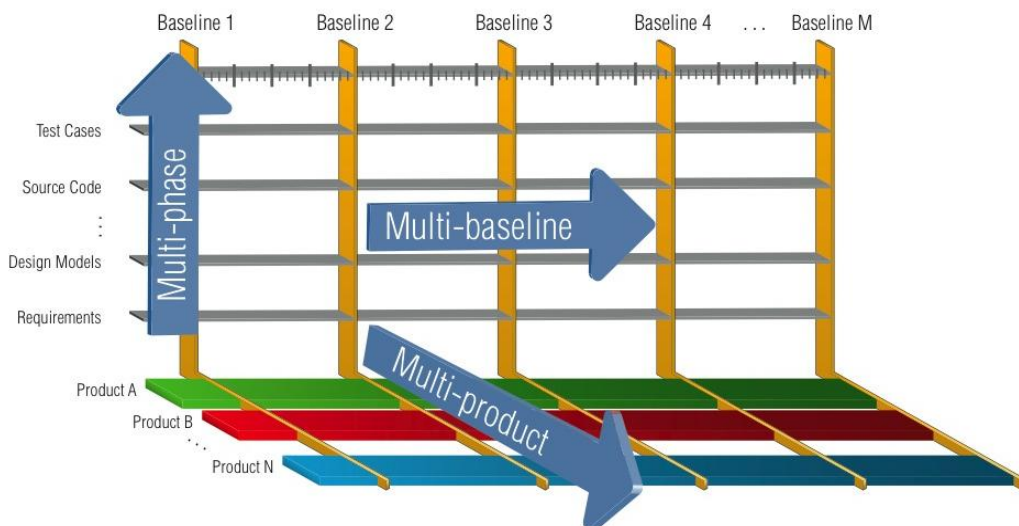


Figure 6: Three Dimensions of Product Line Engineering (Big Lever Software, Inc.)

C. SPL Tools available these days are:-

- a) Gears
- b) MTP (Meta Programming Text processor)
- c) XVCL
- d) Dopler
- e) Pure:: Variant
- f) Varmod

- g) FAMA-FW
- h) FeatureIDE
- i) MetaEdit+
- j) PULSE-BEAT
- k) Holmes
- l) FeatureMapper
- m) S.P.L.OT.(Software Product Line Online Tool)

D. Advantages of software product line:

- a) Productivity gains (the core assets and architecture are reused)
- b) Quality gains (those reused are verified)
- c) Decrease time-to-market
- d) Decreased product risk
- e) Increased market agility
- f) Increased customer satisfaction
- g) More efficient use of human resources
- h) Ability to maintain market presence
- i) Decreases number of Defects
- j) Decrease development cost
- k) Large scale reuse

E. Disadvantages of software product lines:

- a) Investors are reluctant to invest because product line as a software development technique is still under development phase.
- b) Only targeted to big industry (aerospace, automobile, military, mobile phone) , need some case studies from small industry.
- c) Open source community should be involved in the development of product line tools and techniques except few bigger organizations
- d) There are some management and organizational risks involved which needs to mitigate.
- e) The area need equal attention from both academics and industry
- f) Lack of availability of tools, a PhD student in Linkoping University recently presented his thesis which shows a proof of the unavailability of the tools in the software product line.
- g) The problem has also been identified in “Software Product Line Practices and Patters” written by P. Clements and L. Northrop.[5]

F. Application of SPLs

- a) Cummins, Inc., reports that as a result of building the software inside their engines used the PLE, software projects that take a year to perform now take about a week.
- b) The U.S. Army expects to save \$584 million in development costs by procuring a family of live training systems as a product line rather than a series of separate acquisitions.
- c) Hewlett Packard with a product line approach have approached higher where they were able to build products 10 times as complex, with 1/4 of the staff, in 1/3 of the time, and with 1/25 the number of bugs of earlier products.
- d) Nokia has also followed the product line approach, thus producing 25 to 30 different phone models a year.
- e) Weather Stations are also following this approach of product line.

V. Conclusion

SPLs are under research in various areas. Developing of common asset that include test cases, architecture, requirements, design, test plans, reusable software components and many other different artifacts. Individual products are developed from a set of common assets which can lead to increase the productivity, decrease the development effort, cost and time to market. Here it is concluded that Software Product line as an emerging technology being used also bears some disadvantages besides advantages. It has various tools and applications now days. Various companies are involving SPLs. The area needs more attention from the industry as there are more tools available in the academics as compared to the commercial category. The tool support is also significant as there are no standards available to manage the variability among family of products. In the future I will work on the involvement of Software Product Lines with Software Architectures that is Product Line Architectures (PLA), with component model using SPL sample as a creating feature model. In the end I, would like to thank my god, my guide, my family and friends for supporting me in each and every step while working within these areas and supporting for research work to continue.

References:

- [1] Atkinson. Colin, Bayer, Joachim and Muthig, Dirk (2000) “*Component-Based Product Line Development: The Kobra Approach*” SPLC.
- [2] Bayer Joachin et. al. (1999) “*PuLSE: A Methodology to Develop Software Product Lines*”, ACM Press.

- [3] Chaudhary Ankit, Verma, Basant K.(Member IEEE), Raheja, Jagdish L. (2010) “*Product Line Development Architectural Model*”, IEEE.
- [4] J. Bosch, (2000) “Design and Use of software Architectures, Adopting a Product-Line Approach”, Addison-Wesley Publishing
- [5] Klaus Pohl, Günter Böckle, Frank van der Linden, (2005) *Software Product Line Engineering, Foundations, Principles, and Techniques*, Springer.
- [6] Munir Qaiser, Shahid Muhammad et. al., (2010) “*Software Product Line: Survey of Tools*”, LiU Electronic Press.
- [7] P. Clements and L. Northrop, (2002) “*Software Product Lines: Practices and Patterns*”, Addison- Wesley Publishing.
- [8] <http://www.productlineengineering.com/>
- [9] http://www.biglever.com/overview/software_product_lines.html
- [10] <http://www.softwareproductlines.com>