Evolution in Software Product Line

College of Science
Illinois Institute of Technology Chicago, Illinois-60616

Hariprasad Ravi Kumar (hravikum@hawk.iit.edu)

Kavana Somanahalli Umesh (ksomanahalliumesh@hawk.iit.edu)

Abhinav Pelapudi (apelapud@hawk.iit.edu)

Tao Guo (tguo9@hawk.iit.edu)

Jiamin Pan (jpan24@hawk.iit.edu)

Abstract— *Software product lines is defined as a set of software-intensive systems that share a common maintained set of characteristics that specify the need of a particular market and that are developed from a archived set of core assets in a prescribed way.*

Keywords – *Software Product Lines (SPL)*

I. INTRODUCTION

A SPL is nothing but a set of related software products that are generated from reusable assets. 'Products are related' refers to the common functionalities they share. When this kind of reuse is targeted at a specific set of products, it can bring significant productivity and time to market improvements. As SPL is emerging as an important development paradigm in today's world, it allows the companies to realize the order-of-magnitude improvements in quality, cost and other business drivers. SPL engineering can also enable rapid market entry and flexible response, and provide a capability for mass customization.

Driving factors for the organizations to use SPL:

- Cost effectiveness because of reusability of components across products that offer overlapping functionality
- Increase in software quality

Product line assets change over time. In some cases, they change in response to a specific stimulus such as the need to meet a new standard or to address an emerging market niche. In other cases, assets are changed for reasons specific to the asset such as removing defects or achieving consistency with other assets.

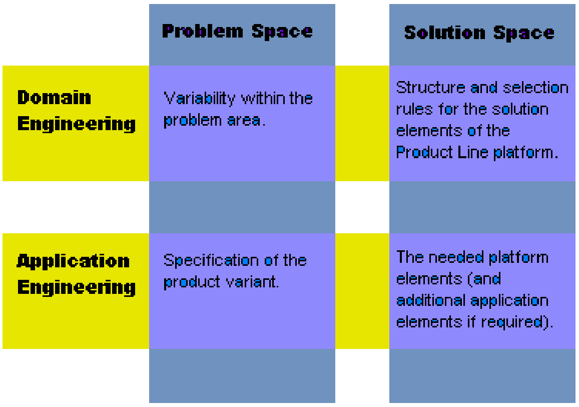
Product line assets change over time. In some cases, they change in response to a specific stimulus such as the need to meet a new standard or to address an emerging market niche. In other cases, assets are changed for reasons specific to the asset such as removing defects or achieving consistency with other assets.

II. SOFTWARE PRODUCT LINE

The main difference between the product line and just the product is the logical separation between the development of core, reusable software assets and the actual application. During the development of the application, platform software is selected and it is configured to meet the specific needs of that application.

The Product Line's commonalities and variability's are defined in the Problem Space. This reflects the range of applications in the Product Line and also their inter-dependencies. Hence, when evolving a product variant, the application developer makes use of the problem space definition to describe the required combination of problem variability's to develop the product variant.

An associated Solution Space describes the constituent assets of the Product Line and its relation to the problem space, i.e. rules like how elements of the product line are selected when certain values in the problem space are selected as part of a product variant.



	Problem Space	Solution Space
Domain Engineering	Variability within the problem area.	Structure and selection rules for the solution elements of the Product Line platform.
Application Engineering	Specification of the product variant.	The needed platform elements (and additional application elements if required).

Figure 1. The four-part division resulting from the combination of the problem space and solution space with domain and application engineering

III. SOFTWARE PRODUCT LINE EVOLUTION

Over the period of time, the software product has evolved drastically. All software evolves, not just product lines assets. A SPL aims to support the development of a family of connected software products from a common set of shared assets. SPL have a considerable life span and shows long-term investment. Without the product line organization to control and direct the evolution, it is more likely that the evolution will be unanticipated and chaotic, and that the quality and integrity of the product line will erode.

SPL process can be seen from the process perspective and organizational perspective. In order to realize a return-on-investment, companies dealing with SPL often plan their product portfolios and software engineering activities strategically over a certain period of time ahead. Compared to single system engineering, SPL evolution exhibits higher complexity due to the variability and the interdependencies between products. The Software product has evolved based on the changes in the requirement which are incurred on the products. These new and changed requirements originate from a number of sources such as the customers using the products, introduction of new products into the product line and future needs predicted by the company.

Evolution in a SPL is complicated by the fact that evolution of a single asset can affect multiple products and many other assets. Many relationships exist among assets in the asset base of a SPL, such as the relationship between the goals in the business case and the structure of the production plan. Creating a product involves the use of many assets, some of which might be derivations of other assets such as the instantiation of the production plan template. One asset might constrain the design or structure of another such as the constraints on the architecture that originate in the business case. Changes made to one asset are propagated to other assets such as when changes to the business case result in changes to the architecture and production plan.

The following table shows the characteristics of the evolutionary and revolutionary approaches of existing set of products and new product lines.

	Evolutionary	Revolutionary
Existing set of products	Develop vision for product line architecture based on the architectures of family members. Develop one product line component at a time (possibly for a subset of product line members) by evolving existing components.	Product line architecture and components are developed based on super-set of product line member requirements and predicted future requirements.
New product line	Product line architecture and components evolve with the requirements posed by new product line members.	Product line architecture and components developed to match requirements of all expected product line members.

There are number of concepts which are related to the evolution in a SPL.

A. Specifying the direction of evolution.

We can define evolution as a process of change in certain direction. This direction can be towards any specific goal. The main challenge is to move these goals in the same direction when each has its own constraints and dependencies.

We there is a plan for evolution, there will be specific objective which will be set and a plan is formed for how to achieve it. For example, let's say that compliance with a new standard is set as an objective. The evolution's scope and cost are evaluated through the change impact analysis. This analysis determines which changes are needed to achieve the objective and begins with an architecture evaluation. Specifying the direction of evolution is integral to developing the evolution plan. The evolution plan specifies how each asset will be moved from its current configuration to the objective configuration. When evolution is due to a change to the architecture, the direction of evolution is to move from one architectural structure to another, or it can be a directed change in the value of certain quality attributes. The direction is mentioned in units that make sense for the type of evolution.

B. Influences on Evolution

The organization and structure of its SPL affects the evolution its assets. There can be two kinds of evolution. Anticipated evolution and unanticipated evolution.

There are three SPL practice areas of primary influences on the mitigation of anticipated evolution.

- Enhances the possibility that any evolution will be anticipated and proactively searches for changes related to advances in the technologies that are used to implement the products in the product line.
- Provides an understanding of which new products are likely to be successful and

which existing products will become obsolete.

The above practice areas will impact few assets. They are the business case and the product lines.

Several events external to the product line organization lead to unanticipated evolution including

- Technical managers and organizational managers sometimes make decisions that are justified politically rather than objectively, leading to unpredictable changes and eventually to unanticipated evolution.
- Although business cycles can be predicted, the effects of their changes cannot always be. Resource reductions "across the board" in a company can trigger unanticipated evolution.
- Technology cycles do not always run their course. In some cases, disruptive technologies gain rapid, widespread acceptance forcing unanticipated changes in products.

C. Evolution Propagation

Figure 2 shows an overview of the evolution process is presented. Each business unit is responsible for a small set of highly related products or one product. Also each business unit initiates the changes in product-specific requirements. These requirements are divided into the requirements specific for the product that the business unit is responsible for and also the requirements which is common for all or most of the products in the SPL. The requirements on the products are basically implemented on product specific code, but common requirements are supported by the component set and by the SPL architecture. The software product-line architecture and its component set is subsequently used to construct new releases or versions of the products in the family. These new versions of the products lead to new requirements, which are fed back into the business unit requirements, thus completing the cycle.

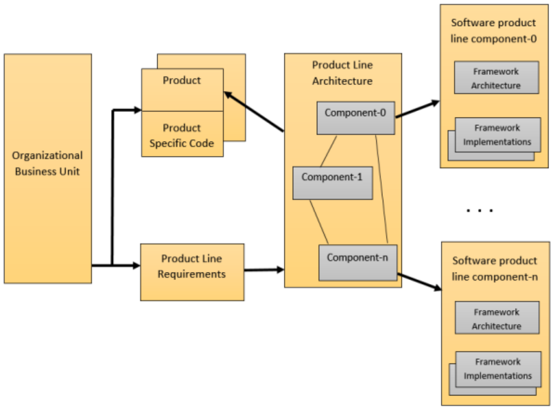

Organizational Business Unit
Product
Product Specific Code
Product Line Requirements
Product Line Architecture
Component-0
Component-1
Component-n
Software product line component-0
Framework Architecture
Framework Implementations
. . .
Software product line component-n
Framework Architecture
Framework Implementations

Figure 2: Evolution of Product Line Architecture

D. Risks of Evolution.

For the success of product lines that result from the evolution has some risks.

- As changes accumulate, related assets might no longer be compatible and might be changed in different directions. Planning for evolution by designing defensively and specifying its direction should mitigate this risk.
- Changing an asset might introduce a defect into it. Specifying the direction of evolution to include a design for the change can mitigate this risk.
- Given changes to a large number of assets, an association could potentially be lost or rerouted during evolution, resulting in an asset being omitted from a configuration and blocked from further changes. Periodically inspecting the output of a

changed process and comparing it to the input can mitigate this risk.

IV. SOFTWARE PRODUCT LINE ACTIVITIES

Product line development involves three essential activities. They are core asset development, product development, and management as shown in figure 3.

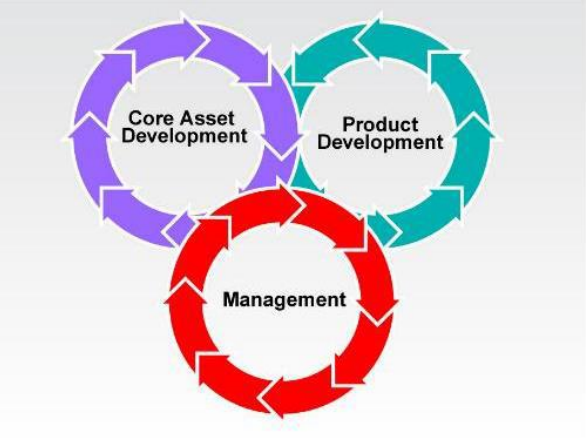


Figure 3. Product line activities

A. Core asset development

The goal of the core asset development activity is to establish a production capability for products. The following figure illustrates the core asset development activity along with its outputs and influential contextual factors.

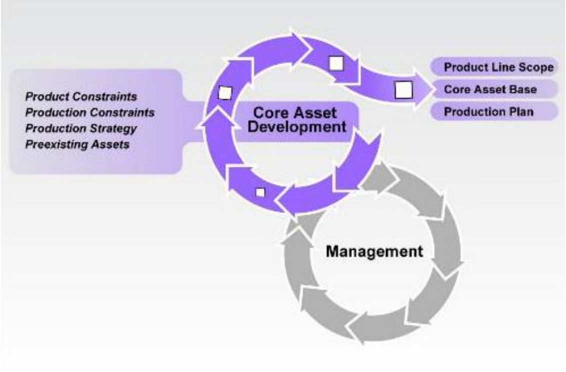


Figure 4. Core Asset Development

Core asset developers apply evolutionary force on the product developers by providing additional variants and new versions of assets. The product developers have to put on effort to understand the new procedures, processes, and interfaces. Continuous releases with trivial changes will consume many development resources with some gain. Waiting too long to release a new version of an asset can allow product teams outside the developing organization to "clone and own" the best-fitting asset and adapt it to their needs.

Core asset developers also exert evolutionary force on management to provide technology forecasts. These forecasts help core asset developers plan which assets to retire, which to invest additional work in, and which to schedule for development. By waiting too long to decide on new technologies can force delays in products or require the product development teams to custom create components that must be redesigned later for product line use as shown in figure 4.

B. Product Asset development

The product development activity depends on the three outputs of the previous activity i.e. the product line scope, the core assets, and the production plan plus the product description for each individual product. The following figure 5 illustrates these relationships.

The rotating arrows indicate iteration and intricate relationships. After the product is developed the product builders may have to give feedback on the problems encountered.

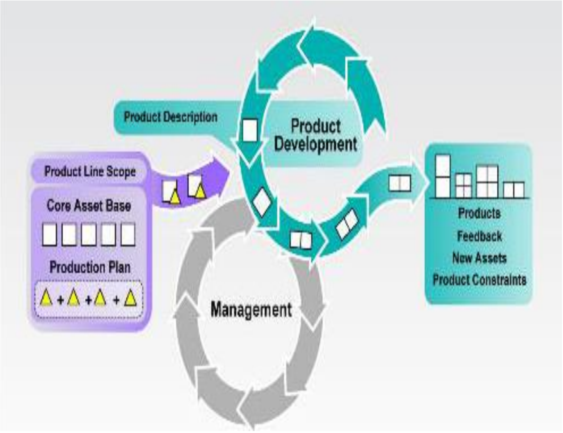


Figure 5. Product development

C. Management

Management exerts evolutionary force on the asset developers by periodically updating technology forecasts and adjusting the business plan for the product line. Asset developers respond to these forces by updating existing assets or creating new ones. Core asset developers also revise the product line scope to accommodate the new products built from the new assets.

Management exerts evolutionary force on the product builders by modifying the business case and the product line scope. Doing so may change the interval between products or at least reprioritize them. Management may also revise the risk analysis causing the product builders to revise the production plan.

V. CASE STUDY: NOKIA

A. About the Company

Nokia is About Connecting People Nokia Mobile Phones produces a wide range of mobile phones. Currently 32 different phones are manufactured covering six different protocol standards, a wide variety of functional features and capabilities, different user interface designs, and many platforms and environments. It is the world's largest manufacturer of mobile phones. Offers much more: mobile devices and solutions for imaging, games, multimedia, mobile network operators and business.

B. Impetus for a Product Line

Nokia’s market share as 23% in the second quarter 2011. It has over 132,000 employees in 120 countries, sales in more than 150 countries. It also has around 220 different phones (till 2011).

Starting since 1994, Nokia created global platform, GSM 900 platforms(2110, 8110 and 6110), GSM 900 and 1800, primarily in Europe and Asia and1900 MHz and TDMA 800, primarily in US and South America, PDC(Personal Digital Cellular), primarily in Japan.

Market maker provides software in Europe for stock market. The company decide to plan the Internet versions as a SPL. Due to its systematic product line approach, the company was able to set up products in a few days. The time to market is 2-4 days and after five products, there is a reduction of 60% in maintenance costs.

Nokia has served product lines in development. An important aspect is that product line architectures tend to be very large. This means that the architecture must be represented in multiple views to be comprehensible by its stakeholders. Each of these views describes the system from a different angle, focusing on certain characteristics of the system. A common architectural view describes the decomposition of the system into parts. The whole system is divided into sub-systems which can be further divided into other modules. An important aspect of the architecture is that it has to be understood by many stakeholders. Each stakeholder may need his own view.

The following are some of the relevant questions with regard to decomposition:

- Are the sub-systems only used on the highest level?
- What does each sub-system contain?
- What are the modules?
- Does a module contains actual code or binaries or some conceptual elements?
- How is everything eventually mapped to concrete assets like classes, files and file structures?

The initial software architecture for this product line addressed variations in hardware, communication standards, and user interfaces; the product line was selected "The Product of the Year" by Business Week and Connect magazines. The current architecture is component based in the client-server style. It allows separate service providers to be plugged in or taken out without restarting the system. This architecture supports both local and remote message passing and component management, task scheduling and event control. Nokia Mobile Phones is the world's largest mobile phone manufacturer, and they believe that SPL engineering has helped it to reach that position.

C. The Payoff

Effectively covers various price segments while maintaining lower development cost. Improves efficiency to further development and manufacturing of products within each series. Changes the competition from a single market to global level. Creates a strong brand image, enables it to set prices that are based more on customer value than on cost.

It also:

- Improved management of very complex systems.
- Improved visibility and reuse of available assets.
- Improved ability to evaluate system level quality properties.
- Improved decision-making based on fact.
- Architectural trade-offs are more concrete.

VI. ADVANTAGES OF SOFTWARE PRODUCT LINE

The main aim of any organization is to produce products that are profitable to organization as well as customers. By using SPL architecture an organization can leverage maximum gain from the product. Another objective is to maintain customer satisfaction, because a product is only successful if the customers are satisfied .By following the techniques of SPL, better quality is assured which keeps the productivity and customer satisfaction high. It will also help organization to be active in the market. Another main advantage is it uses the human resources smartly and efficiently. Apart from organizational benefits there are benefits for stakeholder as well. It will lead to quick development of the product which will increase productivity rate. As the work force are used efficiently its gives opportunity to explore new markets and products. It will also help the technical Manager to predict things easily. As there will more time experimenting with new technology software developers will have job satisfaction.

VII. DISADVANTAGES OF SOFTWARE PRODUCT LINE

There are very few drawbacks of SPL as compared to its advantages. As the products depend on the previous versions, so any new change would increase the complexity of the product. It will create dependency because reusable components are used to create new products. When products are merged the scope increases but increase in scope will lead to complexity and may reduce effectiveness. It is difficult to design the architecture due to lack of guidelines, techniques and tools. It involves risk of investment as the organization cannot get immediate results. Since it is a time taking process it needs long term management. Special units have to be created if employees of an organization are increased, reorganization is necessary. Hence it's not scalable.

VIII. CONCLUSION

The SPL has the advantage of simplicity, and it is easy to communicate with others. it is possible to adopt a SPL approach without changing the existing organization, which may simplify the adoption process. It allows for effective sharing of assets and has the ability to encompass large, complex product lines and organize large numbers of engineers.

The disadvantage is obvious. It is very scalable, this may lead to highly general and flexible reusable components, but systems that do not fulfill the required quality levels. And there will be no entity or explicit incentive to focus on the shared assets. Due to the conflict, it will cause delays in the implementation of new features in the shared assets. Finally, it is difficult to achieve agile reactions to changed market requirements.

SPL evolves over the time. However, many companies are adopting SPL which will benefit the company by reusing the assets. There are many factors which leads to evolution of product like requirement from customers, upgrading the existing standard, new products in the market etc. With SPL method the company can manage similar application together. It reduces development, quality is assured, maintenance costs will be low and gets the products to market faster.

REFERENCES

[1] v.d. Linden, F. (Editor), "Development and Evolution of Software Architectures for Product Families", Proceedings of the Second International ESPRIT ARES Workshop, Las Palmas de Gran Canaria, Spain, LNCS 1429, Springer Verlag, February 1998.

[2]"A Case Study in Successful Product Line Development" By Lisa Brownsword, Paul Clements [CMU/SEI-96-TR-016] [ESC-TR-96-016], October 1996 – ―2.2 The Ship System 2000 Naval Product Line‖

[3] Maturity and Evolution in Software Product Lines: Approaches, Artifacts and Organization

[4] New Methods in Software Product Line Development, Charles W. Krueger BigLever Software, Austin, TX

[5] The Evolution of Product Line Assets, John D. McGregor TECHNICAL REPORT CMU/SEI-2003-TR-005 ESC-TR-2003-005

[6] http://www.sei.cmu.edu/productlines/

[7] Implementing Product Line Variabilities, Michalis Anastasopoulos Fraunhofer Institute Germany