November 7, 2013

A Glossary of Product Line Engineering

Report #20130610001

Contact Information:

info@biglever.com
www.biglever.com
512-426-2227

Systems and software product line engineering (or "product line engineering" for short) is a way to engineer a portfolio of related products in an efficient manner, taking full advantage of the products' similarities while respecting and managing their differences. Like all technical disciplines, product line engineering (PLE) has its own terminology that facilitates more precise communication.

This report defines a vocabulary for PLE. As PLE becomes more and more widely used, and as more technology providers begin to offer PLE solutions, there will be the inevitable reinvention of terms and concepts, making it difficult to compare approaches and judge the efficacy of technology offerings. A standard glossary can provide a common frame of reference against which competing approaches can be compared.

1. Introduction

Systems and software product line engineering (or "product line engineering" for short) is a way to engineer a portfolio of related products in an efficient manner, taking full advantage of the products' similarities while respecting and managing their differences. By "engineer," we mean all of the activities involved in planning, producing, delivering, deploying, sustaining, and retiring products. Considering a portfolio as a single entity to be managed, as opposed to a multitude of separate products to be managed, brings enormous efficiencies in production and maintenance; these efficiencies are delivering order of magnitude improvements in engineering cost, time to market, staff productivity, product line scalability, and quality.

Like all technical disciplines, product line engineering (PLE) has its own technical vocabulary that enables efficient communication of concepts and ideas.

Like all technical disciplines, product line engineering (PLE) has its own technical vocabulary that enables efficient communication of concepts and ideas. A standard glossary can provide a common frame of reference against which competing approaches can be compared and, we hope, ward off some of the unnecessary re-invention of terms.

As PLE continues to grow in acceptance, some who are new to the concepts may be tempted to invent their own terminology for already-established ideas. Indeed, we already see starting to that happen on the part of some technology vendors or academic researchers whose offerings provide partial PLE solutions. The result will be lack of clarify, ambiguity, and confusion on the part of PLE consumers who wish to adopt the most well-proven and comprehensive solution approaches. A standard glossary can provide a common frame of reference against which competing approaches can be compared and, we hope, ward off some of the unnecessary re-invention of terms.

This report defines a glossary for product line engineering. It is centered around the factory-based approach pioneered by BigLever and widely adopted throughout the industry[1]

2. The Factory Paradigm

The concept of a product factory, as illustrated in Figure 1, plays a central role in understanding PLE and can serve as a starting point for introducing the glossary. The key elements of the factory are:

[1] http://www.biglever.com/learn/reports.html

- *Shared PLE Assets* (on the left), such as requirements, design models, source code and test cases. Shared PLE assets contain feature-based variation points that can be configured in different ways to reflect the feature selections for a product.
- *Feature Profiles* (at the top). A feature profile is a description of a product in terms of the features that the product exhibits. A feature profile is an instantiation of a feature model, in which the feature options have been selected for a particular product.
- *Gears Product Configurator* (in the center). The product configurator automatically configures all of the shared PLE assets for a product instance, based on the feature selections in a feature profile.

On the right half of the diagram is the full set of product instances that can be produced by the product configurator in the framework. There is a one-to-one correspondence to the feature profiles and the product instances – each feature profile is used by the product configurator to automatically configure the assets for the corresponding product instance. All the assets from the full engineering lifecycle are produced for each of the products in the product line.

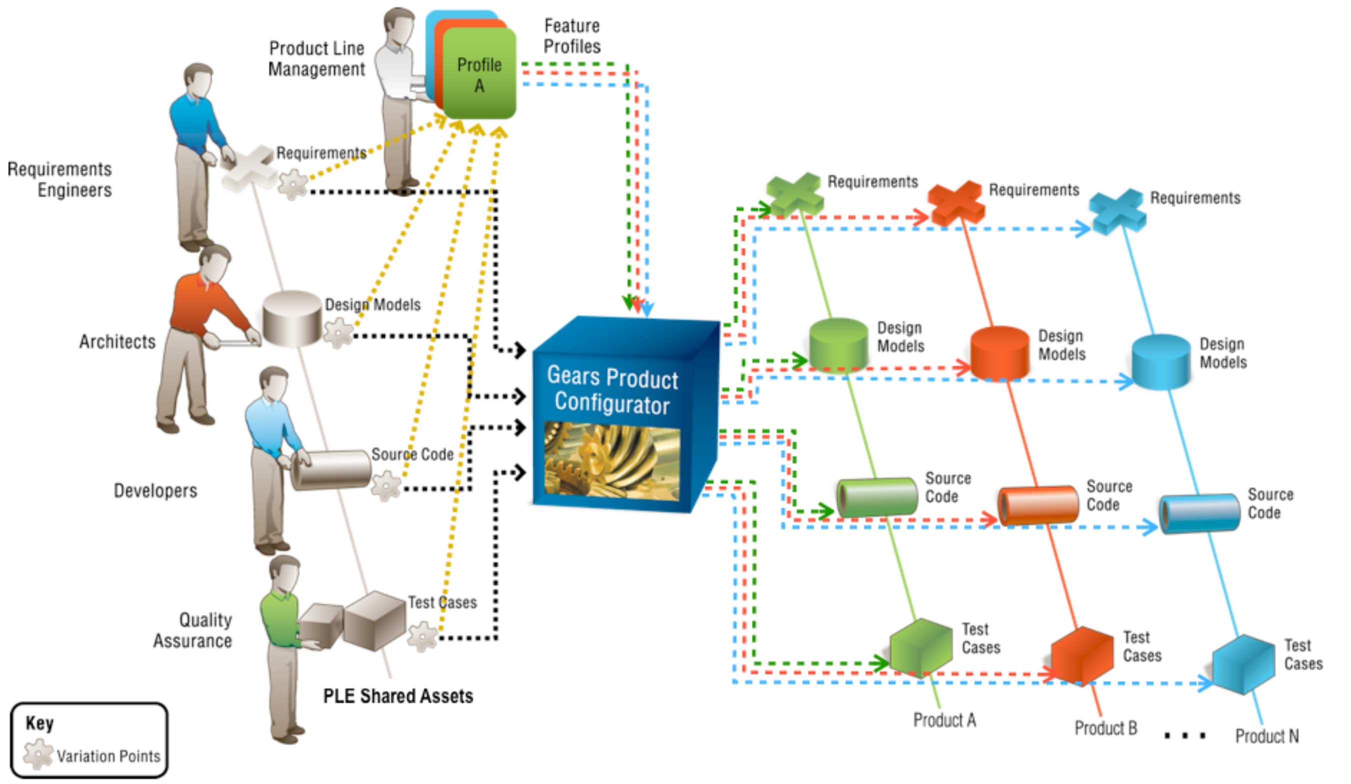

Figure 1. The PLE factory paradigm

3. Using the Glossary

An underlined term in a definition denotes another term defined in this glossary. A new reader may find it convenient to start with the term "systems and software product line engineering" and use that as the basis for exploring the glossary.

The glossary is arranged alphabetically to facilitate rapid look-up, but the terms are also color-coded based on which part of the factory they belong to. Figure 2 re-casts

Figure 1 with colored enclosures added to designate parts of the factory having to do with:

- Shared assets
- Features
- PLE overall
- Product portfolio
- Configurator

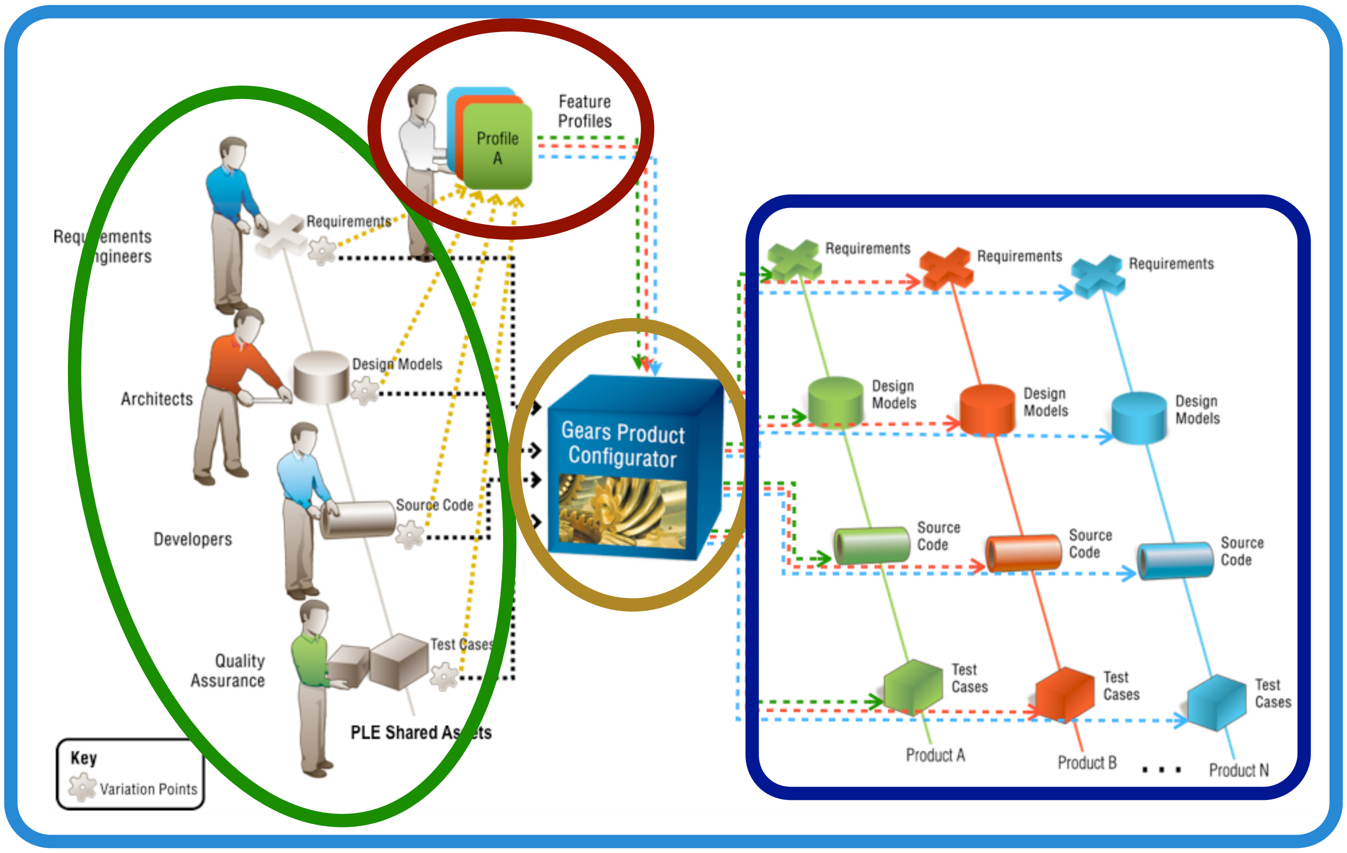

Figure 2. The factory with color-coded sectors

Using this scheme, you may wish to peruse the glossary one "sector" at a time. Again ,you can start with "systems and software product line engineering" and from there go down any path you wish.

4. Glossary

Term	Definition
Bill of Features™	Analogous to a bill of materials, but a way to define a product in terms of its features rather than its parts. A Bill of Features is a useful communication vehicle between business, product marketing, and engineering units for describing a product or family of products. It represents a selection made from a product line's feature catalog.
"clone and own"	The practice of building a new product by making a copy of the shared assets from an existing product and changing them to suit the new product. Clone-and-own is a practice considered antithetical to product line engineering because of its inability to support take advantage of commonality after the initial copy operation.
feature	A distinguishing characteristic of a product, usually visible to the customer or user of that product. An example is a function that one product can perform that others cannot. Features express the customer-visible diversity among the products in a product line. Features are captured in feature declarations.
feature assertion	A constraint or dependency among feature selections. For example, an assertion might express the constraint that a feature (or combination of features), if selected, either requires or excludes the presence of another feature (or combination of features) to be selected. Feature assertions generally capture regulatory, technical, physical, or product marketing constraints that apply to the products in a product line.
feature catalog	A synonym for a feature model, but with emphasis on the capability to define a product or suite of products based on a comprehensive set of feature selections. A feature catalog represents the full set of choices from which a Bill of Features for a product is constructed.
feature declaration	A language construct that names a feature and its possible values. Feature declarations establish the set of available choices that, when made, define a product.
feature model	A feature model consists of (a) a related and self-consistent set of feature declarations that name features and their possible values; (b) feature profiles that specify selections from among the available feature values; and (c) feature assertions that express constraints on combinations of feature selections.
feature profile	A set of selections made from the feature choices made available by a set of feature declarations. A feature profile forms a feature-based description of a product. Constructing a feature profile consists of "walking across" a set of feature declarations and making the necessary selections. The values assigned in feature profiles must satisfy the constraints and dependencies expressed by the feature assertions, if any.

Term	Definition
first generation product line engineering (1GPLE)	An early form of product line engineering characterized by • A strong dichotomy between domain engineering and application engineering, or core asset development and product development. • Explicit inclusion of non-software artifacts in the collection of shared assets, but an unmistakable emphasis on software (under the umbrella of an all-encompassing software architecture) as the principal kind of shared asset. • Focus on features as the language to describe a product line's domain and a way to discriminate products from each other in that domain. • Acknowledgment of configuration management as an essential practice under PLE, but without a strong distinction between shared asset configuration management (CM) and product CM. Contrasted to the state of the art second generation product line engineering (2GPLE).
hierarchical production line	A production line that contains one or more other production lines. Hierarchical production lines allow the constituent production lines to be developed more or less independently of each other, by separate groups in an organization or ecosystem.
means of production	The mechanism that produces the artifacts for each of the products in the product line. The means of production can be manual, but for product lines of any size or frequency of change, manual production is impractical; some form of automation (such as a product configurator) is required.
portfolio	A set of related or similar products produced by an organization.
product	An organization's offering to the market, including all of the engineering artifacts necessary to build, deliver, and sustain it. In PLE, products are described by the properties they have in common with each other and the variations that set them apart. In 2GPLE the distinguishing characteristics are described by features. A product can comprise any combination of • software, • systems in which software runs, or • non-software systems that have software-representable artifacts (such as engineering models or development plans) associated with them.
product configurator	A tool (such as Gears) that automatically and configures the shared assets by exercising their variation points according to a product's feature profile. The result is a set of assets configured to support the product.
product line	A family of products that populate a portfolio and serve target market segments, and are built and maintained in a way that specifically takes advantage of the commonality shared across the family while efficiently and systematically managing the variation among subfamilies and individual products.

Term	Definition
product line engineering (PLE)	The engineering of a portfolio of related products (i.e., a product line) using a common set of shared assets and a common means of production. A particular approach to PLE can be described as first-generation PLE (1GPLE) or second generation PLE (2GPLE). Also called systems and software product line engineering. Contrast with clone-and-own.
production line	A complete system for building products in a product line. A production line consists of • one or more feature models • a collection of shared assets • a product configurator that (to produce each product) applies feature profiles to the variation points in the shared assets • a development environment in which the feature models are built and maintained, the shared assets are endowed with variation points and made available to the product configurator, and the product configurator runs.
second generation product line engineering (2GPLE)	A state-of-the-art form of product line engineering in which: • all artifacts, software and otherwise, are treated equally. As shared assets, they are endowed with variation points expressed using the same language constructs, for a consistent representation of the configurability available across all artifacts. This yields consistent and traceable variation management in artifacts across the full engineering life cycle. • products are produced through the use of high-end industrial strength automation (a product configurator) that configures the shared assets appropriately for each product, • features play a central role in variation management. The variation points in a shared asset are defined by naming the features and feature combinations under which each configuration applies. • a simplified CM policy obviates the need to manage product versions, but only shared asset versions. New versions of products are produced by using the product configurator on new versions of the shared assets. • feature models have encapsulating constructs for hierarchical product lines, facilitating cooperative feature model development across organizational boundaries.

Term	Definition
shared asset	The "soft" artifacts associated with engineering life cycle of the products, the building blocks of the products in the product line. Shared assets can be whatever artifacts are representable with software and either compose a product or support the engineering process to create a product. These can include but are by no means limited to the following: • Requirements • Design specifications • Design models • Source code • Build files • Test plans and test cases • User documentation • Repair manuals and installation guides • Project budgets, schedules, and work plans • Product calibration and configuration files • Data models and parts lists Shared assets are designed to be shared across the product line via built-in variation points. When a product is built, a statement of the product's distinguishing characteristics is applied to exercise these variation points (i.e., cause the change in the asset to occur to meet the needs of the product).
systems and software product line engineering	See product line engineering.
variation point	A place in a shared asset that differs depending on the asset's intended use. In 2GPLE, variation points are expressed in terms of the features that the asset needs to support.